

Ohjelmistotuotannon laadunvarmistuskäytäntöjen kehittäminen Maanmittauslaitoksen asiakasvirastoille tuotettavassa IT-palvelutuotannossa

Pekka Saarinen

Opinnäytetyö
Liiketalouden ylempi
ammattikorkeakoulututkinto
Tietojärjestelmäosaamisen
koulutusohjelma
2017



Tekijä Pekka Saarinen	
Koulutusohjelma Tietojärjestelmäosaamisen koulutusohjelma	
Opinnäytetyön nimi Ohjelmistotuotannon laadunvarmistuskäytäntöjen kehittäminen Maanmittauslaitoksen asiakasvirastoille tuotettavassa IT-palvelutuotannossa	Sivu- ja liitesivumäärä 61 + 36
<p>Tietojärjestelmät ovat muuttuneet viime vuosikymmeninä organisaatioiden tukitoiminnosta ydin-toiminnoksi. Samalla tietojärjestelmien laadunvarmistuksen merkitys on kasvanut.</p> <p>Tämä opinnäytetyö tutki laadunvarmistuksen muutostarpeita Maanmittauslaitoksen Tietotekniikan palvelukeskuksessa (Mitpa). Tutkimus rajattiin koskemaan ohjelmistotuotannon laadunvarmistuskäytäntöjä Mitpan Vallilan toimipaikassa, joka vastaa asiakasvirastoille tuotettavasta IT-palvelutuotannosta. Laadunvarmistuksen osa-alueista valittiin tutkimuskohteiksi testaus, käytettävyyden varmistaminen ja katselmoinnit. Tutkimuksessa selvitettiin laadunvarmistuskäytäntöjen kehittämistarpeita sekä laadunvarmistuskäytäntöjen heikkouksia ja vahvuuksia. Ensisijainen tavoite oli selvittää laadunvarmistuksen kehityskohteita. Tutkimus toteutettiin v. 2015-2017 aikana.</p> <p>Teoriaosa käsittelee ohjelmistotuotantoa, ohjelmiston kehittämisen vaiheita, elinkaarimalleja, ohjelmistotuotannon työkaluja sekä muutoksen johtamista. Teoriaosan lähdemateriaalina käytettiin kirjallisuutta ja painamattomia lähteitä (mm. korkeakoulujen kurssimateriaalia ja opinnäytetöitä).</p> <p>Aineistonkeruuseen sisältyi kahden henkilöstökyselyn toteuttaminen Vallilan toimipaikassa. Niistä ensimmäinen toteutettiin keväällä 2015 ja jälkimmäinen talvella 2017. Tutkimus tuotti vajaan 40 sivun laajuisen aineiston. Aineisto muodostuu kahden kyselyn vastauksista ja vastauksista työpajamuotoisesti suoritetusta analyysistä. Aineiston tuottamiseen osallistui noin neljäsosa Vallilan toimipaikan henkilöstöstä ja he edustivat yhtätoista tehtäväroolia. Kerättyä aineistoa analysoitiin kvalitatiivisesti.</p> <p>Tärkeimpiä laadunvarmistuksen kehittämistoimenpiteitä heikkouksien korjaamiseksi ovat kokonaisvaltaisen laatuajattelun ja -vastuun edistäminen sekä Vallilan toimipaikassa että asiakasvirastoissa, määrittelyjen tarkentaminen, katselmointien toteutuksen ja seurannan tarkentaminen, testausautomaation laajentaminen, laadunvarmistuksen ohjeiden parantaminen, osaamisen tehokkaampi jakaminen (esim. projektityössä syntyneistä hyvistä käytännöistä tiedottaminen) ja laadunvarmistuksen henkilöresurssien vahvistaminen. Lisäksi tulee kehittää esimiesten ja henkilöstön välistä yhteistyötä työn esteiden ym. henkilöstön havaitsemien ongelmien ratkaisemisessa.</p> <p>Vallilan toimipaikan vahvuuksia laadunvarmistuksessa ovat mm. henkilöstön substanssitetämys, testaaajien ammattitaito, testauksen merkityksen ymmärtäminen työyhteisössä, nykyaikaiset työvälineet ja testausmenetelmien kehittämiseen panostaminen.</p>	
Asiasanat ohjelmistotuotanto, laadunvarmistus, muutosjohtaminen	

Author

Pekka Saarinen

Degree programme

Master's Degree Programme in Information Systems Management

Thesis title

Development of quality assurance practices in software engineering in National Land Survey, office of Vallila

Number of pages and appendix pages

61 + 36

During the recent decades, the position of organizations' information systems has changed from support function to core function. Due to this change the importance of quality assurance has increased.

This thesis investigated the change requirements concerning software quality assurance in National Land Survey (NLS). It is an agency of the Finnish Government. The target unit of thesis was NLS' Service Center of Information Technology and its office in Vallila, Helsinki. Vallila office is responsible for delivering IT services for agencies operating in the same administrative sector. It is governed by the Ministry of Agriculture and Forestry. Thesis focused on the following areas of quality assurance: testing, usability assurance and reviews. The objective was to investigate the needs of development concerning quality assurance practices as well as weaknesses and strengths of quality assurance practices. The primary objective was to explore ways to develop practices of quality assurance. The research was carried out during 2015-2017.

Theoretical framework of thesis deals with software engineering, software development phases, life cycle models, tools in software engineering and change management. The source material for the theoretical framework was collected from literature and unprinted sources (e.g. course material of universities and theses).

The collection of data included the implementation of two surveys at Vallila office. The first was implemented in spring 2015 and the latter in winter 2017. The research produced nearly 40 pages of material. The material consists of personnel's responses to surveys and an analysis of personnel's responses. The analysis was made in a workshop. About one quarter of the personnel at Vallila office participated in the production of material. Those employees represented eleven roles in IT. The collected material was analyzed qualitatively.

The most important quality assurance measures to correct weaknesses are the promotion of comprehensive quality thinking and quality responsibility at both Vallila office and customer agencies, clarification of specifications, refinement of implementation and follow-up concerning reviews, extension of test automation, improvement of quality assurance guidelines, more efficient sharing of know-how (e.g. informing good practices of project work) and strengthening the human resources of quality assurance. In addition, co-operation between supervisors and personnel should be developed in order to solve problems identified by the personnel.

Vallila office's strengths in quality assurance are, among others, personnel's administrative sector-related knowledge, skills of the testers, work community's understanding towards the importance of testing, modern software engineering tools and the development of testing methods.

Keywords

software engineering, quality assurance, change management

Sisällys

1 Johdanto.....	1
1.1 Tutkimusaiheen valinta ja sen perustelut.....	1
1.2 Tutkimusongelmat ja tutkimuksen rajaus.....	2
1.3 Tutkimuksen keskeiset käsitteet.....	3
2 Teoria.....	4
2.1 Ohjelmiston kehittämisen vaiheet.....	4
2.1.1 Esitutkimus.....	5
2.1.2 Määrittely.....	5
2.1.3 Suunnittelu.....	6
2.1.4 Toteutus.....	7
2.1.5 Laadunvarmistus.....	8
2.1.6 Käyttöönotto.....	12
2.1.7 Ylläpito.....	12
2.2 Ohjelmiston elinkaarimalleja.....	13
2.2.1 Vesiputousmalli.....	13
2.2.2 Prototyypilähestymistapa.....	14
2.2.3 Spiraalimalli.....	15
2.2.4 RUP.....	16
2.2.5 Ketteriä kehittämismalleja.....	17
2.3 Ohjelmistotuotannossa käytettävät työkalut.....	19
2.4 Muutoksen johtaminen yleisesti.....	20
2.5 Esimerkki muutoksen johtamismallista: John Kotterin muutosprosessi.....	22
2.5.1 Luo kiireen ja välttämättömyyden tuntu.....	23
2.5.2 Perusta ohjaava tiimi.....	24
2.5.3 Laadi visio ja strategia.....	25
2.5.4 Viesti muutosvisio.....	26
2.5.5 Anna henkilöstölle laajat valtuudet.....	27
2.5.6 Lyhyen aikavälin onnistumiset.....	28
2.5.7 Vakiinnuta parannukset, tuota lisää muutoksia.....	29
2.5.8 Juurruta uusia toimintatapoja yrityskulttuuriin.....	31
3 Opinnäytetyön kohdeorganisaatio ja kehittämismenetelmä.....	33
3.1 Kohdeorganisaatio.....	33
3.2 Aineistonkeruumenetelmä ja sen perustelut.....	34
4 Opinnäytetyön tulokset.....	35
4.1 Kyselyn 1 vastaajat ja vastausaktiivisuus, kevät 2015.....	35
4.2 Kyselyn 2 vastaajat ja vastausaktiivisuus, talvi 2017.....	36
4.3 Kyselyiden vastausten analyysi.....	36
5 Johtopäätökset.....	46

5.1 Aineiston analyysi.....	46
5.2 Kehittämisehdotukset kohdeorganisaatiolle.....	47
6 Pohdinta ja jatkotutkimusehdotukset.....	54
Lähteet.....	58
Liitteet.....	62
Liite 1. Kyselylomakkeet	62
Liite 2. Kyselyn 1 vastaukset (kevät 2015).....	71
Liite 3. Kyselyn 2 vastaukset (talvi 2017).....	72

1 Johdanto

Opinnäytetyön toteuttamisen taustana on iso, kolmea virastoa koskenut organisaatiomuutos. Muutosprosessi käynnistyi v. 2013. Prosessin aikana toteutetut yt-neuvottelut johtivat yhdessä virastossa henkilöstövähennyksiin.

Prosessin lopputulos oli se, että Tiken (Maa- ja metsätalousministeriön tietopalvelukeskus) tietotekniikkapalvelut ja Geodeettinen laitos (GL) yhdistyivät Maanmittauslaitokseen (MML) 1.1.2015. Lisäksi Tiken Tilastopalvelut-yksikkö siirtyi 2015 alussa toimintansa aloittaneeseen Luonnonvarakeskukseen (Luke). Edellä kuvattujen muutosten myötä Tiken ja Geodeettisen laitoksen toiminta itsenäisinä virastoina päättyi. Yhdistämällä Tiken tietotekniikkapalvelut Maanmittauslaitoksen ICT-toimintoihin syntyi Maanmittauslaitokseen uusi toimintayksikkö, Tietotekniikan palvelukeskus (Mitpa). Se toimii maa- ja metsätalousministeriön hallinnonalan IT-palvelukeskuksena. Yksikössä työskentelee vajaat 300 työntekijää. Olen entinen Tiken työntekijä. Työskentelen Mitpassa laadunvarmistuksen asiantuntijana.

1.1 Tutkimusaiheen valinta ja sen perustelut

Hakiessani tietojärjestelmäosaamisen koulutusohjelmaan Haaga-Heliaan syksyllä 2014 pohdin osana hakuprosessia opinnäytetyöni aihetta. Tuolloin tarkoitukseni oli tutkia - virastojen yhdistymisestä johtuen - Mitpan laadunvarmistusta kokonaisuutena ja kirjoittaa tutkimustulosten pohjalta laadunvarmistuskäsikirja Mitpan laadunvarmistuskäytäntöjen yhdenmukaistamiseksi. Laadunvarmistuskäytännöillä tarkoitetaan laadunvarmistuksen osalualueita (esim. testaus ja katselmoinnit) sekä niissä käytössä olevia työmenetelmiä ja työvälineitä. Opinnäytetyöprosessin aikana toimintayksikön laadunvarmistuskäytännöt dokumentoitaisiin yhteen keskitettyyn tietolähteeseen. Yhdenmukaiset toimintatavat sujuvoittaisivat ohjelmistotuotannossa työskentelyä, yksiköiden yhteistyötä ja henkilöiden siirtymistä yksiköiden välillä sekä selkeyttäisivät viestintää. Opintojeni alkuvaiheessa, keväällä 2015 järjestetyssä kyselyssä Mitpan henkilöstö valitsi toimintayksikkömme tärkeimmäksi kehityskohteeksi toimintatapojen yhdenmukaistamisen. Lähimmät esimieheni eivät kuitenkaan pitäneet keväällä 2015 laadunvarmistuskäsikirjan toteuttamista realistisena tutkimusaiheena. Ratkaisusta johtuen jouduin muuttamaan opinnäytetyöni aihetta. Rajasin tutkimukseni kohteen toimipaikkaani Helsingin Vallilassa ja sen laadunvarmistukseen.

Aloitin työni Tikessä vuonna 2008. Edeltävänä vuonna Tikessä oli aloitettu ketterän sovel- luskehityksen sisäänajo. Menetelmäksi oli valittu Scrum. Sen jälkeen ohjelmistotuot- toamme on kehitetty panostamalla mm. yksikkötestaukseen ja ottamalla käyttöön jatkuvan integraation palvelin (Hudson/Jenkins), järjestelmätestauksen automatisointi sekä katsel-

moinnit. Tikessä osallistuin kahden viimeksi mainitun tehtävän osalta niiden suunnitteluun sekä niihin liittyvien ohjeiden laatimiseen ja tehtäviin perehdyttämiseen.

Yleisesti prosessien uudistamisen, työmenetelmien kehittämisen ja uusien työvälineiden käyttöönoton perustana on tarve tukea sujuvaa työskentelyä ja organisaation tavoitteiden saavuttamista. Opinnäytetyössäni halusin tutkia laadunvarmistuksen muutostarpeita eli millaisia kehittämistarpeita on laadunvarmistuskäytännöissä Mitpan Vallilan toimipaikassa (115 työntekijää). Tutkimukseni tuloksilla voitaisiin tukea sekä työnantajani että toimittajiemme/alihankkijoiden toiminnan kehittämistä. Tutkin laadunvarmistuksen kehitystarpeita toteuttamalla kaksi kyselyä Vallilan toimipaikassa. Tein opinnäytetyöni v. 2015-2017 aikana kyetäkseni vertaamaan kahtena eri ajanjaksona toteutetun kyselyn tuloksia. Keräämäni aineisto auttaisi myös selvittämään, toimivatko prosessit valitsemillani laadunvarmistuksen osa-alueilla sekä niihin liittyvät työmenetelmät ja työvälineet sujuvan työskentelyn tukena.

Opinnäytetyön esitutkimuksena toimi v. 2015 keväällä Soveltava tutkimus ja kehittäminen -opintojaksolla (MET2HY101) toteuttamani kehittämistehtävä. Se tutki laadunvarmistuksen kehittämistarpeita toimintayksikköni Sovelluskehitys-vastuualueella. Sain opintojaksosta kiitettävän arvosanan. Kurssipalautteessaan opettajani totesi tutkimusaiheeni hyödyllisyyden ja aiheen olevan hyvä pohja opinnäytetyön toteuttamiselle. Hän neuvoi minua opinnäytetyössäni täydentämään sisältöä muutosjohtamisen opeilla. Esitutkimuksen kyselyn tulokset (liite 2) toimivat vertailupohjana opinnäytetyön kyselyn tuloksille (liite 3).

1.2 Tutkimusongelmat ja tutkimuksen rajaus

Tutkimusongelmat kysymyksen muodossa ovat:

- Mitkä ovat ohjelmistotuotannon laadunvarmistuskäytäntöjen heikkoudet ja vahvuudet asiakasvirastoille tuotettavassa IT-palvelutuotannossa?
- Miten ohjelmistotuotannon laadunvarmistuskäytäntöjä tulisi kehittää asiakasvirastoille tuotettavassa IT-palvelutuotannossa?

Tutkimuksen kohde rajataan seuraavasti:

1. Tutkimus kohdistuu vain ohjelmistojen laadunvarmistukseen (englanniksi software quality assurance).
2. Laadunvarmistuksen osa-alueista: testaus, käytettävyyden varmistaminen ja katselmoinnit.
3. Tutkimus käsittelee vain Mitpan Vallilan toimipaikassa (asiakasvirastoille tuotettavassa IT-palvelutuotannossa) käytössä olevia laadunvarmistuksen käytäntöjä. Mitpan muissa toimipaikoissa ja Maanmittauslaitoksen muissa toimintayksiköissä käytössä olevat laadunvarmistuskäytännöt on suljettu tutkimuksen ulkopuolelle.

4. Tutkimuksen ensisijainen tarkoitus on selvittää laadunvarmistusta koskevia kehittämistarpeita.

Tutkimuskohteen rajauksesta huolimatta tutkimuksen tulokset ovat osittain yleishyödyllisiä. Tuloksia voidaan siten hyödyntää myös muissa Maanmittauslaitoksen toimipaikoissa.

1.3 Tutkimuksen keskeiset käsitteet

Tutkimuksen ydinkäsitteet ovat:

- *Ohjelmistotuotanto*. Työmenetelmiä ja työnjohdon menetelmiä, joita hyödynnetään tietokoneohjelmien sekä tietokoneohjelmistojen tuottamisessa.
- *Laadunvarmistus*. Ohjelmistotuotannon osa-alue, joka pyrkii estämään virheiden päätyminen ohjelmistoon, toisaalta löytämään ohjelmistossa olevat virheet. Ohjelmiston laatua tarkastellaan kokonaisuutena, jossa laadukas ohjelmisto syntyy laadukkaan toiminnan lopputuloksena.
- *Muutosjohtaminen*. Yhteisen ymmärryksen ja oppimisen rakentamista muutoksen tai muutosten toteuttamiseksi jossakin yhteisössä, kuten yrityksessä, virastossa tai kunnassa. Muutosjohtamisella ohjataan ihmisten toimintaa jonkin tavoitteen saavuttamiseksi. Muutosten vakiinnuttaminen edellyttää kaikkien ihmisten sitouttamista muutosten toteuttamiseen. Muutosjohtamisen oppeja voi hyödyntää myös yksilö omien tavoitteidensa suunnittelussa.

2 Teoria

Käsite ohjelmistotuotanto (*software engineering*) tarkoittaa työ- ja työnjohdon menetelmiä, joita hyödynnetään tietokoneohjelmien ja niiden muodostamien tietokoneohjelmistojen tuottamisessa. Ohjelmistotuotantoon sisältyy kaikki toiminta, jonka tavoitteena on valmistaa tietokoneohjelmia tai -ohjelmistoja. (Haikala & Märijärvi 2004, 16; Wikipedia 2015a.) Haikalan ja Märijärven (2004, 16) mukaan ohjelmistotuotantoon sisältyvät seuraavat ohjelmiston tuotantoprosessin osa-alueet

- laatujärjestelmä
- projektinhallinta
- dokumentointi
- tuotteenhallinta
- laadunvarmistus
- määrittely
- suunnittelu
- toteutus
- testaus
- käyttöönotto
- ylläpito.

2.1 Ohjelmiston kehittämisen vaiheet

Ohjelmiston kehittämistarve voi kohdistua joko olemassa olevaan tai täysin uuteen ohjelmistoon. Tarpeen voi synnyttää esimerkiksi asiakkaan liiketoiminnassa tapahtuneet muutokset tai teknologian kehittyminen. Organisaation tietojärjestelmäkehityksen pitkän aikavälin tarpeet kuvataan kokonaistutkimuksessa. Siinä mm. priorisoidaan kehittämistarpeet eli määritetään, missä järjestyksessä tietojärjestelmäprojekteja toteutetaan. (Pohjonen 2002, 26.)

Ohjelmistoa kehitetään vaiheittain. Se helpottaa eri vaiheissa toteutettavien tehtävien ja niiden aikataulun sekä riippuvuuksien hahmottamista. Vaihejako toimii myös projektinhallinnan tukena. Periaatteessa yksittäisen vaiheen käynnistäminen edellyttää sitä edeltävän vaiheen valmistumista. Käytännössä vaiheet ovat usein päällekkäisiä. Vaiheiden muodostama kokonaisuutta kutsutaan ohjelmiston elinkaareksi (*life cycle*). Se tarkoittaa ohjelmiston kehittämisen aloittamisen ja ohjelmiston käytöstä poistamisen välistä ajanjaksoa. (Haikala & Märijärvi 2004, 36; Pohjonen 2002, 26.)

2.1.1 Esitutkimus

Esitutkimuksessa (*feasibility study*) tai tarvekartoituksessa (*requirements study*) arvioidaan, onko idea tai ehdotus ohjelmiston kehittämiseksi järkevä. Tällöin tutkitaan perusteellisesti, *miksi* ohjelmistoa pitäisi kehittää. Samalla arvioidaan, ovatko kehittämisen tavoitteet realistiset, mihin sidosryhmiin (esim. ohjelmiston loppukäyttäjät) kehittäminen vaikuttaa ja mitä ratkaisumalleja on tavoitteiden saavuttamiseksi. Vaiheessa määritetään yleiset järjestelmätason vaatimukset, joista käytetään myös nimitystä asiakasvaatimukset (*customer requirements*). Niiden sisältöä kerätään sidosryhmien edustajilta esim. haastattelemalla tai työpajoissa. Asiakasvaatimukset kuvaavat asiakkaan eli ohjelmiston tilaajan tarpeita. Siten esitutkimus tuottaa tietoa ohjelmiston kehittämisen käynnistämisestä päättävälle henkilölle. (Haikala & Mikkonen 2011, 66; Haikala & Märijärvi 2004, 37-38; Pohjonen 2002, 27-29.) Esitutkimuksessa ei oteta kantaa siihen, miten ohjelmiston kehittäminen tulisi tehdä (Wikipedia 2015a).

Vaiheen synnyttämä tuotos on esitutkimusraportti (Pohjonen 2002, 27). Sen sisältö on tiivistetysti seuraava (Pohjonen 2002, 27):

1. Organisaation tietojenkäsittelyn nykytilanne.
2. Kuvataan ongelmat, joihin ohjelmiston kehittämistarpeella etsitään ratkaisua.
3. Kuvaus sidosryhmistä, joihin kehittäminen vaikuttaa.
4. Alustavat kehitettävälle ohjelmistolle asetetut tavoitteet ja rajaukset.
5. Ratkaisuvaihtoehdot ohjelmiston kehittämiseksi.
6. Alustava suunnitelma ohjelmiston kehittämishankkeelle.

Esitutkimus on ohjelmiston elinkaaren näkökulmasta erittäin kriittinen vaihe. Sen lopputulos voi olla, että kehittämishanketta ei ole järkevää käynnistää, koska hankkeella ei ole onnistumisen edellytyksiä. Mikäli hanke käynnistyy, esitutkimus tarjoaa pohjatietoja hankkeen suunnittelulle. Siksi vaiheen aikana tuotettavat asiakasvaatimukset ja huolellisuus niiden selvittämisessä vaikuttavat ratkaisevasti kehittämishankkeen onnistumiseen. (Haikala & Märijärvi 2004, 37; Pohjonen 2002, 27.)

2.1.2 Määrittely

Määrittelyvaiheessa (*requirements specification*) analysoidaan asiakasvaatimuksia. Samalla hahmotetaan käsitystä siitä, *mitä* ohjelmisto tekee. (Haikala & Mikkonen 2011, 61; Haikala & Märijärvi 2004, 38-40; Pohjonen 2002, 28; Wikipedia 2015b; Wikipedia 2015c.) Vaatimuksia tarkennetaan, jolloin syntyy seuraava ryhmittely (Haikala & Mikkonen 2011, 61; Haikala & Märijärvi 2004, 38-40; Lescelius 2010, 45; Pohjonen 2002, 28; Wikipedia 2015b; Wikipedia 2015c):

1. Toiminnalliset vaatimukset. Kuvataan, mitä eri käyttäjät tekevät ohjelmistolla. Määrittellään käyttöliittymä ja ohjelmiston kommunikointi toisten ohjelmistojen kanssa.
2. Ei-toiminnalliset vaatimukset. Kuvataan mm. suorituskykyä (esim. kuormankesto ja vastausajat), käytettävyyttä ja tietoturvaan koskevat vaatimukset.
3. Rajoitteet. Toiminnallisille vaatimuksille asetettuja rajoituksia, esim. ohjelmiston toteutuksen sitominen tiettyyn ohjelmointikieleen.

Määrittelyvaiheessa tulee huomioida kaikki mahdolliset vaatimusten lähteet. Niihin sisältyvät eri sidosryhmien edustajat ja heidän esittämänsä tarpeet, ohjelmiston toteuttajaorganisaatiossa käytössä olevat ohjeet sekä ulkoiset tekijät, kuten standardit ja lainsäädäntö. Täten varmistetaan, että määrittelyssä ohjelmiston vaatimuksista kerättävät tiedot ovat mahdollisimman kattavat. (Haikala & Mikkonen 2011, 65; Pohjonen 2002, 28-29.)

Myös vaatimusten huolellinen dokumentointi on tärkeää. Vaatimukset kuuluu dokumentoida siten, että jokainen vaatimus numeroidaan. Vaatimusten toteuttamisen aikataulutusta helpottaa, jos ne on priorisoitu. Mitä selkeämmin ja tarkemmin vaatimukset on kirjattu, sitä helpompaa on niiden toteuttaminen, testaaminen ja jäljitettävyyden hankkeen myöhemmissä vaiheissa. (Haikala & Mikkonen 2011, 64; Pohjonen 2002, 30-32.) Jäljitettävyydellä tarkoitetaan vaatimuksen seurantaan määrittelystä toteutukseen saakka ja päinvastoin. Jäljitettävyyden avulla auttaa myös hahmottamaan vaatimusten välisiä riippuvuuksia. (Haikala & Mikkonen 2011, 63.) Määrittelyvaiheen lopputuloksena syntyvä tuotos on toiminnallinen määrittely (*functional specification*) (Haikala & Märijärvi 2004, 38-39). Toiminnallinen määrittely toimii pohjana mm. ohjelmiston tekniselle määrittelylle, testauksen suunnittelulle ja käyttöohjeiden laatimiselle (Pohjonen 2002, 32).

Puutteellinen vaatimusmäärittely voi aiheuttaa merkittäviä lisäkustannuksia ohjelmiston kehittämisessä (Pohjonen 2002, 29). Usean tutkimuksen mukaan ohjelmistoprojektien epäonnistumisten syistä 60 % liittyy vaatimuksissa ja niiden käsittelyssä olleisiin puutteisiin (Haikala & Mikkonen 2011, 61).

2.1.3 Suunnittelu

Suunnitteluvaiheessa (*design*) määritetään, *miten* ohjelmisto toteutetaan (Haikala & Märijärvi 2004, 40). Vaihe jakautuu yleensä kahteen osaan: arkkitehtuurisuunnitteluun ja modulisoinnintuun (Haikala & Märijärvi 2004, 40; Pohjonen 2002, 32). Haikala & Mikkonen (2011, 180-181) esittelee ohjelmiston suunnittelua ohjaavia periaatteita, joita ovat

- yksinkertaisuus
- suoraviivaisuus
- osittaminen

- rajapinnat
- toteutuksen yhdenmukaisuus.

Arkkitehtuuri tarkoittaa kokonaisuutta, joka sisältää ohjelmiston osat ja niiden väliset suhteet. Standardointijärjestö IEEE:n määritelmän mukaan arkkitehtuuriin kuuluvat myös ohjelmiston suunnittelemista sekä jatkokehitystä ohjaavat periaatteet. (Haikala & Mikkonen 2011, 178.)

Arkkitehtuurisuunnittelussa (*architectural design*) määritellään ohjelmiston keskeinen terminologia eli käsitteet ja jaetaan ohjelmisto osiin. Niitä kutsutaan moduuleiksi. (Haikala & Mikkonen 2011, 178; Haikala & Märijärvi 2004, 40.) Ne vastaavat tietyistä ohjelmiston toiminnoista ja kommunikoivat ympäristönsä, kuten toisten moduulien, kanssa rajapinnan kautta. Moduulien tulee olla mahdollisimman itsenäisiä. Siksi arkkitehtuurin suunnittelua ohjaa tavoite, jonka mukaan moduuleilla tulee olla mahdollisimman vähän riippuvuutta toisistaan. Tämä helpottaa mm. testausta ja ylläpitoa. (Helsingin yliopisto 2009, 11; Haikala & Märijärvi 2004, 40; Pohjonen 2002, 32-33.) Erityisesti ohjelmiston osat, joihin ennustetaan tehtävän tulevaisuudessa eniten muutoksia, tulisi eristää omiin moduuleihinsa (Pohjonen 2002, 38).

Moduulisuunnittelu (*module design*) tarkoittaa moduulien sisäisen rakenteen suunnittelua (Haikala & Märijärvi 2004, 81). Suunnittelussa tavoitellaan mahdollisimman pientä moduulin kokoa (Pohjonen 2002, 32).

Suunnitteluvaiheessa luotu tuotos on tekninen määrittely (*technical specification*). Se sisältää kuvaukset mm. ohjelmiston tarkoituksesta, toteutustyön periaatteista, laitteisto- ja ohjelmistoympäristöstä, moduuleista, rajapinnoista ja tietokanta-arkkitehtuurista. (Haikala & Märijärvi 2004, 83; Pohjonen 2002, 33.) Dokumentti toimii työohjeena sovelluskehittäjille heidän suunnitellessaan ja toteuttaessaan ohjelmistoa (Pohjonen 2002, 32).

2.1.4 Toteutus

Toteutusvaiheessa (*implementation*) koodataan eli kirjoitetaan ohjelmiston osien lähdekoodia jollakin korkean tason ohjelmointikielellä, kuten Javalla tai C-kielellä. Tästä työstä vastaavat sovelluskehittäjät. Tietokoneelle asennettu kääntäjäksi (*compiler*) kutsuttu ohjelma muuttaa tekstin muotoon kirjoitetun lähdekoodin konekielelle. Se on kieli, jota tietokoneen prosessori ymmärtää. Konekieli muodostuu biteistä eli ykkösistä ja nolista. Sovelluskehittäjien kirjoittaman lähdekoodin sijaan toteutukseen voidaan joskus käyttää myös sovelluskehittäjä. (Helsingin yliopisto 2002; Murch 2002, 102; Pohjonen 2002, 34; Teknillinen korkeakoulu 2001; Wikipedia 2013a; Wikipedia 2015d; Wikipedia 2015e.)

Koodinkirjoittamisprosessin aikana sovelluskehittäjät muuntavat moduulisuunnittelun fyysiseksi sovellukseksi, jonka tietokone suorittaa. Lisäksi sovelluskehittäjät etsivät virheitä kirjoittamansa ohjelmiston osan lähdekoodista. Tätä kutsutaan yksikkötestaukseksi. (Kasurinen 2013, 51; Murch 2002, 102; Pohjonen 2002, 34.) Kun sovelluskehittäjien toteuttamia ohjelmiston osia on yhdistetty toisiinsa, tälle kokonaisuudelle suoritetaan integrointitestaus. Silloin esimerkiksi tarkastetaan, toimiiko moduulien välinen kommunikointi. (Kasurinen 2013, 54.)

Toteutuksen onnistuminen riippuu toteuttajien ammattitaidosta ja siitä, miten hyvin toteuttavan ohjelmiston ominaisuudet on pystytty määrittelemään kehittämishankkeen edellisissä vaiheissa. Osa toteuttajan ammattitaitoa on ohjelmointityyli. Siihen sisältyy mm. lähdekoodin selkeys, koodin kommentointi ja nimeämiskäytäntöjen noudattaminen. (Pohjonen 2002, 34-35.) Toteutuksen mittareita ovat (Pohjonen 2002, 34-35):

1. Ohjelmistolle asetettujen vaatimusten täytyminen.
2. Toiminnallisen ja teknisen määrittelyn noudattaminen.

2.1.5 Laadunvarmistus

Haikalan & Märijärven (2004, 48) mukaan ohjelmiston laatu tarkoittaa ”ohjelmistotuotteen kykyä täyttää käyttäjänsä kohtuulliset toiveet ja odotukset”. Laatua tulee kuitenkin tarkastella kokonaisuutena, jossa laadukas ohjelmisto syntyy laadukkaana toiminnan lopputulokseksi. Tällöin toimintaa ohjaa esimerkiksi organisaation laatujärjestelmä. (Haikala & Märijärvi 2004, 48.)

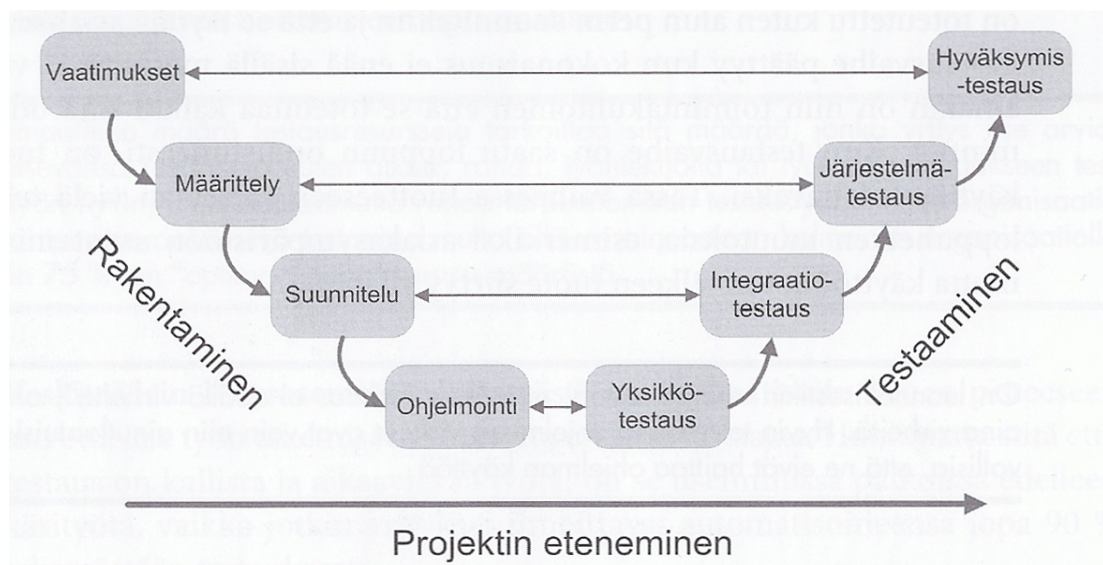
Kaikkiin ohjelmiston kehittämisen vaiheisiin sisältyy laadunvarmistustehtäviä. Niitä ovat mm. katselmoinnit, testaus ja auditoinnit. (Haikala & Märijärvi 2004, 37; Pohjonen 2002, 78; Wikipedia 2015f.) Laadunvarmistus (*quality assurance*) pyrkii - mahdollisimman varhaisessa vaiheessa - estämään virheiden päätyminen ohjelmistoon, toisaalta löytämään ohjelmistossa olevat virheet. Muita tehtäviä ovat mm. tavoitteiden, prosessien, menettelytapojen ja standardien noudattamisen arviointi sekä sen tulosten raportointi organisaation johtohenkilöille. Myös laadunvarmistuksen toimintatapoja arvioidaan ja niitä kehitetään. Laadunvarmistukseen liittyviä käsitteitä ovat myös verifiointi ja validointi. Verifiointissa tarkastetaan, onko ohjelmisto toteutettu määritysten mukaisesti. Validoinnissa tutkitaan, vastaako ohjelmisto käyttötarkoitustaan. (Haikala & Märijärvi 2004, 51; Murch 2002, 187-189; Pohjonen 2002, 78.)

Katselmointi (*review, technical review*), tarkastus (*inspection*) ja läpikäynti (*walkthrough*) ovat työmenetelmiä, joissa tutkitaan ohjelmiston elinkaaren eri vaiheissa syntyneitä tuo-

toksia, kuten määrittelydokumenteja ja lähdekoodia. Menetelmien ideana on koota yhteen joukko ihmisiä, joiden tavoitteena on etsiä ja poistaa virheitä tuotoksesta ennen kuin sitä käytetään elinkaaren seuraavaan vaiheeseen, esim. toteutuksen, tietolähteenä. Menetelmät toimivat myös oppimisprosessina. (Haikala & Mikkonen 2011, 197-200; Haikala & Märijärvi 2004, 51; Jyväskylän yliopisto 2004; Wikipedia 2015g.) Menetelmien sisältö tiivistetysti (Haikala & Mikkonen 2011, 197-200; Haikala & Märijärvi 2004, 51; Jyväskylän yliopisto 2004; Wikipedia 2015g):

1. Katselmointi. Yleensä projektin jonkin etapin, kuten määrittelyvaiheen, arviointia kokonaisuutena. Tällöin selvitetään, onko kaikki dokumentit ym. kyseisen vaiheen tavoitteisiin sisältyneet tuotokset saatu valmiiksi.
2. Tarkastus. Muodollinen, osallistujien tarkan roolituksen sisältävä kokoustekniikka, joka painottuu testausta edeltävissä vaiheissa syntyneiden tuotosten tarkastamiseen. Yleinen arvio on, että tarkastuksilla kyetään löytämään n. 50-80 % kaikista virheistä.
3. Läpikäynti. Epämuodollinen versio tarkastuksesta. Käytetään erityisesti lähdekoodin esittelyyn, arviointiin sekä kommentointiin.

Ohjelmistokehityksessä testaus (*testing*) on perinteisesti määritelty suunnitelmalliseksi virheiden sekä puutteiden etsimiseksi. Testauksessa löydetty virhe (*bug, error*) tarkoittaa, että ohjelmisto toimii spesifikaatioiden, kuten toiminnallisen ja teknisen määrittelyn, vastaisesti. Virheiden etsiminen ei voi kuitenkaan perustua ainoastaan spesifikaatioihin, koska niissä on aina puutteita. (Haikala & Mikkonen 2011, 205-206; Kasurinen 2013, 50.)



Kuva 1. Testauksen V-malli (Kasurinen 2013, 14)

Testauksen vaiheistusta on tapana visualisoida V-mallilla (kuva 1). Se on muunnos vesiputousmallista, joka on yksi ohjelmiston elinkaarimalleista (katso luku 2.2). Vesiputousmalliin verrattuna V-mallissa testaus ei ole vain yksi, erillinen vaiheensa, vaan jokaista vaihetta kohden on oma testaustehtävänsä: yksikkötestaus linkittyy toteutusvaiheen tuotoksiin, integrointitestaus suunnitteluun, järjestelmätestaus määrittelyyn ja hyväksymistestaus asiakkaan vaatimuksiin. (Kasurinen 2013, 14; Lehtimäki 2006, 151.)

V-mallissa testauksen suhdetta tuotoksiin kuvataan testaustasoilla (Kasurinen 2013, 14). Niiden tarkempi kuvaus on seuraava (Auer ym. 2013, 93; Haikala & Mikkonen 2011, 206-209; Kasurinen 2013, 51-57; Lescelius 2010, 47):

1. Yksikkötestaus (*module testing, unit testing*). Suoritetaan toteutustyön yhteydessä. Tällöin verrataan ohjelmiston yksittäistä osaa, kuten moduulia, luokkaa, funktiota tai oliota moduulisuunnittelua koskeviin kuvauksiin teknisessä määrittelyssä. Sovelluskehittäjä tekee testitapauksia, joiden avulla hän tutkii, kääntyykö kirjoitettu koodi virheettömästi ja reagoiko toteutettu ohjelmiston osa oikein sille annettuihin syötteisiin. Testauksessa voidaan tarvita testipetejä (*test bed*) ja niihin sisältyviä, ohjelmisto-osan ympäristöä simuloivia testiajureita (*test driver*) ja tynkämoduuleita (*test stubs*). Niillä tutkitaan ohjelmiston osan toimivuutta. Yksikkötestauksen suorittaa sovelluskehittäjä.
2. Integrointitestaus (*integration testing*). Testien suunnittelemisen pohjana käytetään arkkitehtuurisuunnittelua. Testaus painottuu ohjelmiston osat toisiinsa yhdistävien rajapintojen toimivuuden tutkimiseen, esim. moduulien väliseen viestinvaihtoon. Yleensä sovelluskehittäjät vastaavat integrointitestauksesta ja se etenee rinnakkain yksikkötestauksen kanssa.
3. Järjestelmätestaus (*system testing*). Aloitetaan, kun ohjelmisto on yksikkötestattu ja sen jälkeen integrointitestauksen yhteydessä kasattu toimivaksi kokonaisuudeksi. Järjestelmätestauksen tehtävänä on tarkastaa, toimiiko ohjelmisto kokonaisuutena oikein ja täyttääkö ohjelmisto kaikki sille asetetut tavoitteet. Testien suunnittelu ja suorittaminen perustuu ohjelmiston toiminnalliseen määrittelyyn. Muita testien suunnittelemisen lähteitä ovat mm. käyttöohjeet. Koska määrittelyssä on tavallisesti puutteita, järjestelmätestausta täydennetään muilla testaustehtävillä, esim. käytettävyydestestauksella, suorituskykytestauksella (järjestelmän vastausaika ja kuormankesto) ja tutkivalla testauksella. Järjestelmätestauksessa tärkeä periaate on testaajien riippumattomuus. Siksi on suositeltavaa, että järjestelmätestauksen suorittavat eri henkilöt kuin sovelluskehittäjät.
4. Hyväksymistestaus (*acceptance testing*). Testauksen työvaihe, jonka päätavoitteena on tutkia, onko ohjelmisto riittävän laadukas täyttämään vaatimusmäärittelyssä asetetut vaatimukset. Hyväksymistestaus tarkoittaa ohjelmiston virallista tarkastamista, jonka suorittaa asiakas. Testaus suoritetaan ohjelmiston lopullisessa käyttö-

ympäristössä. Mikäli tässä vaiheessa löydetään virheitä, ne tulisi käsitellä siten, että vain käyttöönnoton estävät virheet korjataan. Tällöin loput virheet korjataan ohjelmiston tulevia versioita toteutettaessa.

Testaustasoihin jakamisella on merkityksensä myös ohjelmiston kehittämiskustannuksissa. Mitä myöhäisemmässä vaiheessa ohjelmiston virhe havaitaan, sitä kalliimpaa on virheen korjaaminen. Siksi testaus aloitetaan mahdollisimman aikaisin. (Haikala & Mikkonen 2011, 208.) Edellä kuvattu kustannusvaikutus koskee myös tilannetta, jossa ohjelmistoon halutaan lisätä uusia ominaisuuksia kehittämisprosessin loppuvaiheessa (Kasurinen 2013, 57).

Yksittäisen virheen korjaaminen edellyttää usein muutoksia moneen ohjelmiston osaan. Kaikkia muutoksia ei välttämättä huomata tehdä. Virhekorjaus voi siten synnyttää ohjelmiston toimintoihin uusia virheitä. Siksi tarvitaan regressio- eli uudelleentestausta. Tällöin, virheen korjaamisen jälkeen, ei testata ainoastaan korjattua toimintoa, vaan kaikkia ohjelmiston toimintoja. (Haikala & Mikkonen 2011, 208.)

Käytettävyys vaikuttaa käyttäjien työskentelyn tehokkuuteen ja tarkkuuteen sekä käyttäjätyytyväisyyteen (Tampereen teknillinen yliopisto 2007, 10). Käytettävyystestauksessa testin ohjaaja laatii testihenkilöille (yleensä 3-5 henkilöä) tehtäviä, jotka heidän kuuluu suorittaa arvioinnin kohteena olevalla ohjelmistolla. Tehtäviä suorittaessaan henkilöt kuvailevat suullisesti, mitä he ohjelmistolla tekevät. Ohjaaja seuraa henkilöiden toimintaa ja kirjaa muistiinpanoja. Tarvittaessa testaustilanne voidaan videoida. Käytettävyystestausta voidaan täydentää käytettävyysasiantuntijoiden tekemällä heuristisella arvioinnilla. Se pohjautuu heuristiikkoihin eli sääntö- tai ohjekokoelmiin. (Jyväskylän yliopisto 2003; Tampereen teknillinen yliopisto 2007, 23; Tampereen yliopisto 2000.)

Testausta tehdään manuaalisesti eli käsityönä ja koneellisesti. Testauksen automatisointi tarkoittaa, että testit suorittaa jokin testausohjelma. Tällöin testausohjelmalla esim. luodaan testiskripti eli sarja käskyjä ja ohjeita, joiden perusteella ohjelma käyttää testattavaa ohjelmistoa. Automatisointi ei valmisteluiden osalta eroa manuaalisesta testauksesta: testitapaukset pitää suunnitella ennen testien suorittamista. Automatisointi soveltuu hyvin tilanteeseen, jossa testattaviin ohjelmiston osiin tehdään vain vähän koodimuutoksia ja tiettyjä testejä suoritetaan toistuvasti. (Pohjolainen 2003, 23-25.)

Täydellinen testaus ei ole käytännössä mahdollista. Tähän vaikuttavat mm. käyttäjien kirjoittamat syötteet ja niissä esiintyvien variaatioiden suuri määrä. (Kasurinen 2013, 19-20; Pohjonen 2002, 36.) Siksi laadukas testauksen johtaminen, testien suunnittelu ja testien suorittaminen edellyttävät (Kasurinen 2013, 20)

- syvällistä ohjelmiston toiminnan ymmärtämistä
- testauskohteiden selkeää priorisointia.

Auditoinnin (*auditing, audit, assessment*) avulla arvioidaan organisaation toiminnan laatua peilaamalla sitä organisaation laatujärjestelmään tai standardiin (Haikala & Mikkonen 2011, 144-145). Jotkin organisaatiot hakevat laatujärjestelmälleen sertifikaattia eli haluavat osoittaa järjestelmänsä noudattavan jotakin kansainvälistä standardia, esim. ISO 9001 (Haikala & Märijärvi 2004, 50). Laatujärjestelmällä tarkoitetaan yleensä laatukäsikirjaa ja siihen liittyviä ohjeita. Niiden tehtävänä on varmistaa, että organisaatiossa tuotetaan tuotteita tai palveluita niille asetettujen laatukriteereiden, aikataulun ja budjetin mukaisesti. Pelkkien organisaation toiminnassa olevien puutteiden etsimisen sijaan auditoinnissa listataan myös hyviä toimintatapoja. Auditoinnit ovatkin yksi kanava levittää organisaatiossa tietämystä hyvistä käytännöistä. Auditointi toteutetaan joko sisäisesti tai ulkopuolisen tahon toimesta, esim. kohdeorganisaation työntekijöitä haastatteleamalla. (Haikala & Mikkonen 2011, 145; Pohjonen 2002, 78; Wikipedia 2015h.)

2.1.6 Käyttöönotto

Käyttöönoton (*installation*) yhteydessä asiakkaan hyväksymästä ohjelmistosta valmisteluun kokonaisuus, joka asennetaan määrätyille laitteille ohjelmiston lopulliseen käyttöympäristöön. Tätä tehtävää kutsutaan tuotantoonsiirroksi tai tuotantoonvienniksi. Sitä edeltäviin tehtäviin kuuluu mm. tarvittavan laitteiston ja tietoliikenneyhteyksien valmistelu. Lisäksi tulee huolehtia käyttäjien ja ohjelmiston ylläpitohenkilöstön kouluttamisesta. Mikäli kehittäminen on kohdistunut käytössä olevaan ohjelmistoon, tuotantoonsiirrossa on huomioitava, että se voi aiheuttaa käyttökätkön. Siksi tuotantoonsiirrosta tulee tiedottaa ajoissa ohjelmiston käyttäjiä. (Auer ym. 2013, 93; Pohjonen 2002, 37; Wikipedia 2015a.)

2.1.7 Ylläpito

Ylläpito (*maintenance*) on ohjelmiston elinkaaren pisin vaihe. Sen tehtävänä on varmistaa tuotantokäytössä olevan ohjelmiston toimintakyky. Vaiheeseen sisältyy mm. virheiden korjaamista, uusien ominaisuuksien toteuttamista sekä toteutuksen ja testauksen dokumentointia. (Pohjonen 2002, 37-38.) Ylläpito on kustannuksiltaan korkein elinkaaren vaihe: se voi muodostaa 70-80 % ohjelmiston koko elinkaaren kustannuksista (Auer ym. 2013, 94; Pohjonen 2002, 38).

Ohjelmiston elinkaari päättyy alasajoon eli käytöstä poistamiseen (Wikipedia 2015a). Syitä käytön päättämiseksi ovat esimerkiksi (Wikipedia 2015a):

1. Toinen, parempi menetelmä korvaa ohjelmiston käytön.

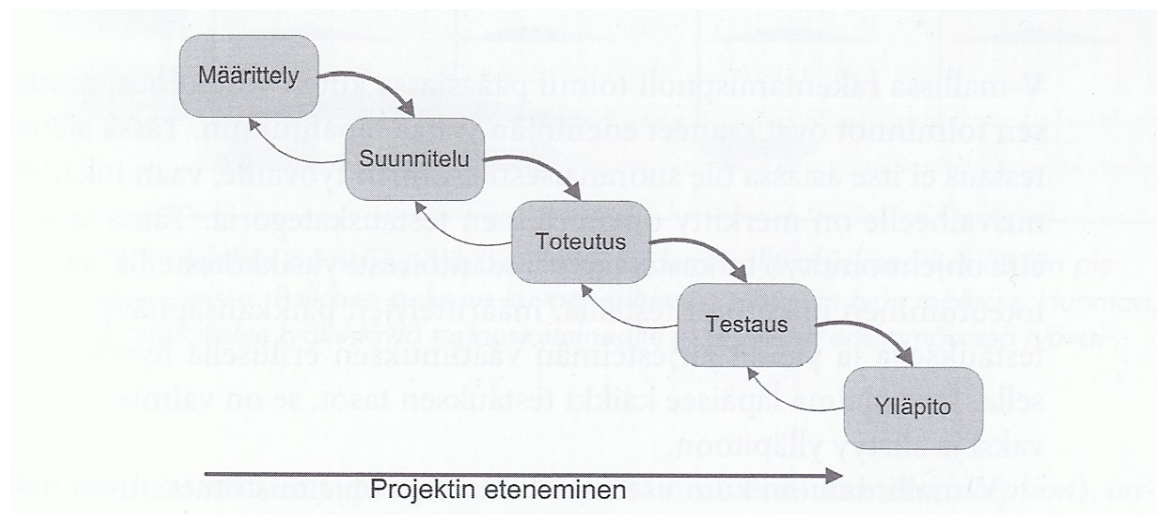
2. Tarve ohjelmiston toiminnoille päättyy.
3. Ohjelmiston toiminnot siirretään uuteen ohjelmistoon.
4. Ohjelmiston kehitystyö loppuu.

2.2 Ohjelmiston elinkaarimalleja

Ohjelmiston kehittämisen vaiheistus voidaan laatia ohjelmiston kehitystyölle tai ohjelmiston koko elinkaarelle (Haikala & Märijärvi 2004, 36). Elinkaarimalli määrittää, miten ohjelmiston kehittämisen vaihejakoa sovelletaan (Pohjonen 2002, 25). Malleissa tulee huomioida se, että ne eivät kuvaa yksittäisen organisaation toimintaa, vaan ne ovat ehdotuksia. Mallien soveltamisen yksityiskohdissa on siten vaihtelua organisaatiokohtaisesti. (Kasurinen 2013, 29.)

Karkeasti ottaen elinkaarimallit jaetaan kahteen kategoriaan: suunnitelmalähtöisiin ja ketteriin (Kasurinen 2013, 24). Useaan elinkaarimalliin sisältyy iteraatio-termi. Se tarkoittaa nk. miniprojektia, jonka vaiheistus on sama kuin vesiputousmallissa (katso luku 2.2.1): määrittely, suunnittelu, toteutus ja testaus. (Haikala & Märijärvi 2004, 45.)

2.2.1 Vesiputousmalli



Kuva 2. Vesiputousmalli (Kasurinen 2013, 13)

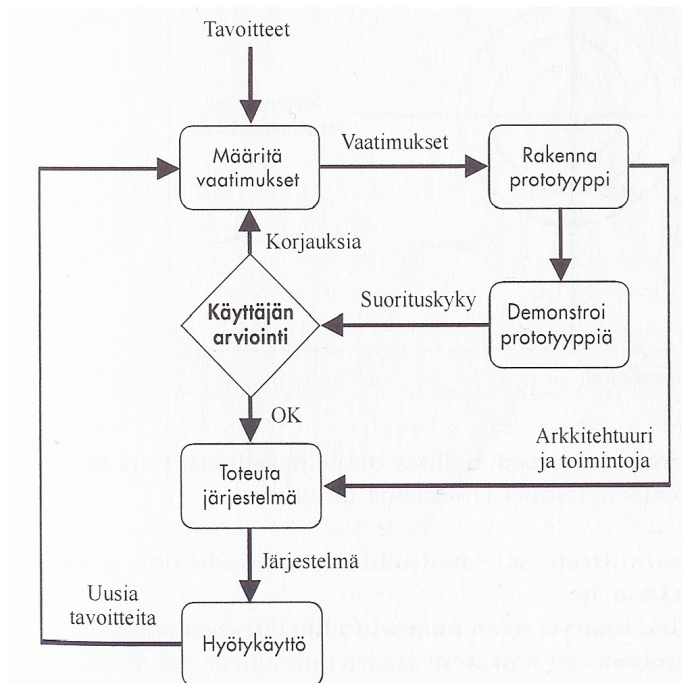
Vesiputousmalli (kuva 2) on tunnetuin elinkaarimalli (McConnell 2002, 137; Pohjonen 2002, 40). Mallissa ohjelmiston kehittäminen toteutetaan peräkkäisinä, ei päällekkäisinä vaiheina. Jokaisen vaiheen päätteeksi järjestetään katselmointi. Kehittämisprosessi ei ete-

ne seuraavaan vaiheeseen ennen kuin katselmoitava vaihe todetaan valmiiksi. (McConnell 2002, 136.)

Kuuluisin kuvaus vesiputousmallista on amerikkalaisen Winston W. Roycen vuonna 1970 julkaisemassa artikkelissa, vaikka siinä ei vesiputous-termiä mainitakaan (Haikala & Mikkonen 2011, 37; McConnell 2002, 137; Wikipedia 2015i). Royce korosti esittämässään mallissa iteratiivisuutta eli mahdollisuutta palata vaiheissa taaksepäin, kuten oheisessa kuvassa on esitetty. Hänen artikkelinsa ymmärrettiin laajasti väärin: malli tulkittiin jäykkänä, vain yhteen suuntaan etenevänä ohjelmiston kehitysprosessina. (Haikala & Mikkonen 2011, 37.)

Periaatteessa vesiputousmalli sopii kehittämisen malliksi silloin, kun kehittämisen kohde tunnetaan ja kyetään kuvaamaan (dokumentoimaan) yksityiskohtaisesti jo hankkeen alkuvaiheessa (McConnell 2002, 136-137). Käytännössä malli ei toimi kirjaimellisesti noudatettuna, koska ohjelmiston vaatimukset tarkentuvat ja niihin tehdään muutoksia hankkeen edetessä (Haikala & Märijärvi 2004, 41).

2.2.2 Prototyypilähestymistapa



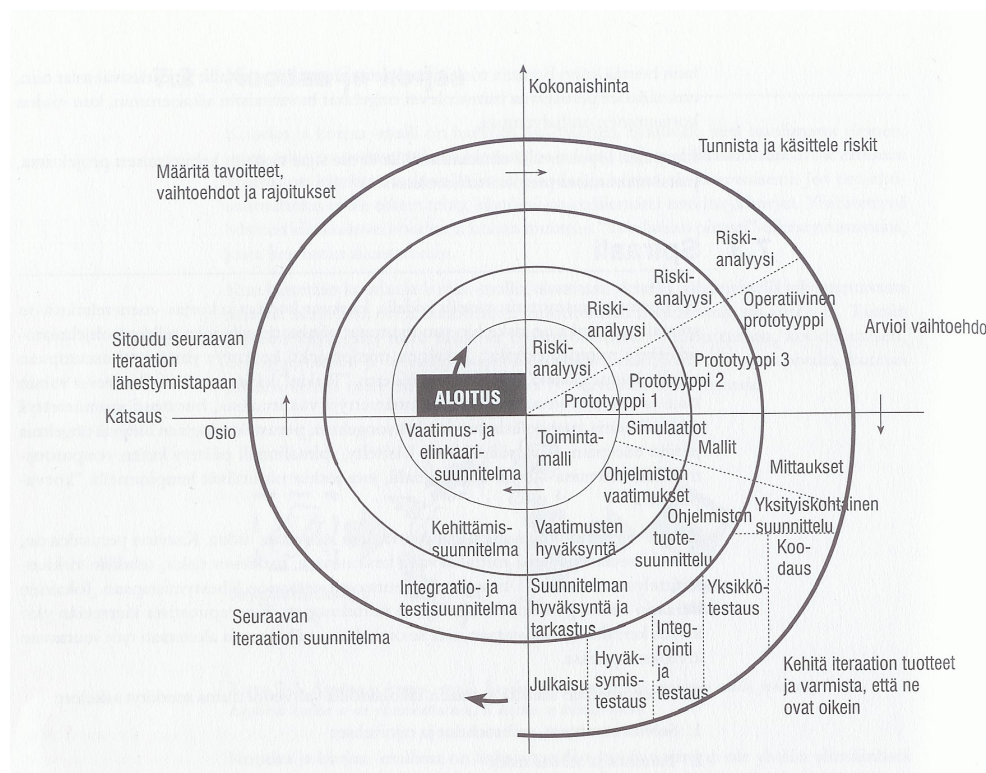
Kuva 3. Prototyypilähestymistapa (Pohjonen 2002, 41)

Prototyypilähestymistapa (kuva 3) tai protoilu tarkoittaa kehittämismallia, jossa suppeiden vaatimusten pohjalta toteutetaan ohjelmiston prototyyppi eli alkuperäinen tai ensimmä-

mäinen versio (Pohjonen 2002, 41; Wikipedia 2013b). Prototyyppi voidaan toteuttaa esim. ohjelmoimalla tai paperille piirtämällä (Haikala & Mikkonen 2011, 38-39). Asiakkaan palautteen perusteella parannetaan prototyyppiä. Sen arviointia ja kehittämistä jatketaan niin pitkään, että asiakas on toteutettuihin ratkaisuihin tyytyväinen. Tämän jälkeen prototyyppiä käytetään varsinaisen ohjelmiston toteutuksen perustana. Kun elinkaaren aikana ilmenee uusia vaatimuksia, ne läpikäyvät edellä kuvatun protoiluprosessin ennen toteuttamista. (Pohjonen 2002, 41.)

Asiakas kykenee prototyyppin avulla hahmottamaan paremmin ohjelmiston toiminnallisuuksia. Siksi protoilu sopii erityisesti käyttöliittymän määrittelyyn. (Haikala & Märijärvi 2004, 43.) Protoilussa on tärkeää muistuttaa asiakasta, että prototyyppi on vain keskeneräinen malli. Samalla sen ominaisuuksia tulee arvioida kriittisesti ennen kuin ohjelmistoa toteutetaan mallin pohjalta. Muutoin on vaarana, että ohjelmistoon valitaan toiminnallisuuksia, joita on esimerkiksi työlästä toteuttaa tai niiden suorituskyky jää heikoksi. (Haikala & Mikkonen 2011, 39; Pohjonen 2002, 41.)

2.2.3 Spiraalimalli



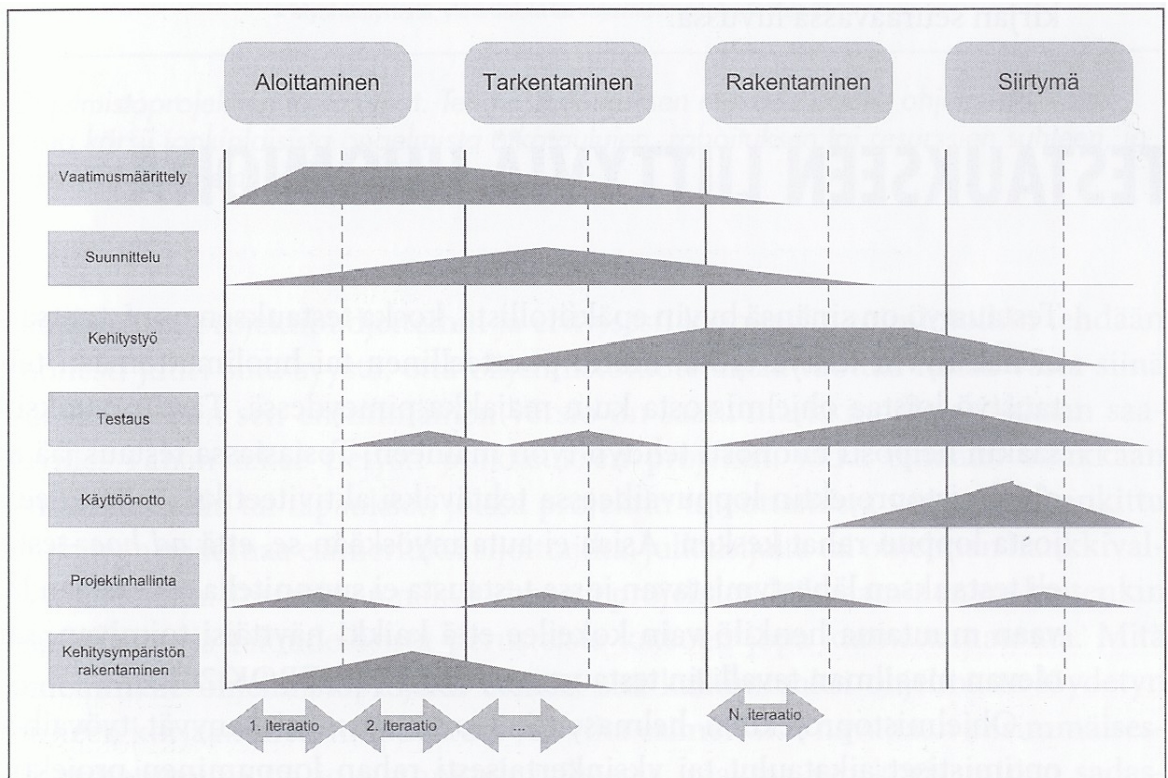
Kuva 4. Spiraalimalli (McConnell 2002, 142)

Spiraali (kuva 4) on prosessin iteratiivisuuteen ja riskien arviointiin painottunut kehittämis-
malli. Riskejä voivat olla mitkä tahansa tekijät, joihin liittyy epävarmuutta, kuten määritte-
lyn tarkkuus tai ohjelmiston suorituskyky. Iteratiivisuudella eli vaiheiden toistamisella pyri-
tään pienentämään riskejä. (McConnell 2002, 141; Pohjonen 2002, 43.)

Mallissa ohjelmistohanke on pilkottu miniprojekteiksi. Toteutus pääsee käynnistymään
vasta, kun merkittävimmät riskit on huolella analysoitu ja huomioitu. (McConnell 2002,
141.) Kehittämisprosessi voidaan keskeyttää, mikäli riskit arvioidaan liian suuriksi. Mitä pi-
temmälle spiraalissa edetään keskiöstä kohti ulommaista kehää, sitä enemmän ohjelmis-
ton yksityiskohdat tarkentuvat. Riskienhallinnan painottamisen ansiosta spiraalimallin yh-
tenä vahvuutena pidetäänkin sitä, että kehittämisen edistyessä ja siten kustannusten kas-
vaessa riskit vähenevät. (McConnell 2002, 143; Pohjonen 2002, 43.)

Jotta malli olisi toimiva, se vaatii riskianalyysin hallitsemista. Lisäksi asiakkaan tulee sitou-
tua vahvasti kehittämisprosessiin. (Pohjonen 2002, 43.)

2.2.4 RUP



Kuva 5. RUP (Kasurinen 2013, 25)

RUP-mallissa (kuva 5) ohjelmiston kehittäminen muodostuu neljästä päävaiheesta. Kukin
päävaihe sisältää yhden tai useamman iteraation. Sen pituus on n. 1-3 kuukautta. (Haika-

la & Märijärvi 2004, 45; Kasurinen 2013, 27.) Päävaiheisiin sisältyy (Haikala & Märijärvi 2004, 45; Kasurinen 2013, 25)

- aloittaminen: jäljittelee esitutkimusvaihetta
- tarkentaminen: tuotetaan ohjelmiston perusarkkitehtuuri
- rakentaminen: jokaisella iteraatiolla toteutetaan ohjelmistosta uusi versio, joka annetaan valikoitujen käyttäjien käyttöön
- siirtymä: jäljittelee käyttöönottoa.

Luonteenomaista RUP:lle on se, että ohjelmiston rakentaminen aloitetaan riskialtteimmista osista (Kasurinen 2013, 25-26). RUP:n eroja vesiputousmalliin verrattuna (Kasurinen 2013, 25-26):

1. Vaiheet ovat päällekkäisiä.
2. Osa kehittämistyön tehtävistä (esim. vaatimusten tarkentaminen sekä testaaminen) toistuu kaikissa vaiheissa.
3. Testaus aloitetaan välittömästi, kun ohjelmiston 1. prototyypiluonnos on valmis.
4. Vaatimuksiin voidaan tehdä joustavasti muutoksia kehittämisprosessin aikana.

2.2.5 Ketteriä kehittämismalleja

Auer ym. (2013, 43) mukaan ketterän ohjelmistokehityksen peruseriaate on ”tuottaa asiakkaan tarpeisiin parempi ohjelmisto nopeammin ja vähemmin resursein”. Toisen määritelmän mukaan ketterä kehittäminen tarkoittaa viestinnän, asiakasyhteistyön ja ohjelmiston toiminnallisuuksien korostamista, yksityiskohtaisen ennakkosuunnittelun tai sopimus pohjaisten määrittelyjen sijaan (Kasurinen 2013, 27).

Vuonna 2001 perustettiin Yhdysvalloissa Agile Alliance-järjestö. Sen perustajat halusivat kehittää vaihtoehtoisia menetelmiä perinteisille, raskaille ohjelmiston kehittämismalleille, kuten vesiputousmallille tai RUP:lle. Järjestö julkaisi periaatelistan nimeltään Agile Manifesto. Sen mukaan tärkeintä ohjelmiston kehittämisessä on - riippumatta esim. prosessista, työkaluista ja dokumentoinnista - tyytyväinen asiakas ja toimiva ohjelmisto. (Haikala & Mikkonen 2011, 43-45; Kasurinen 2013, 27.)

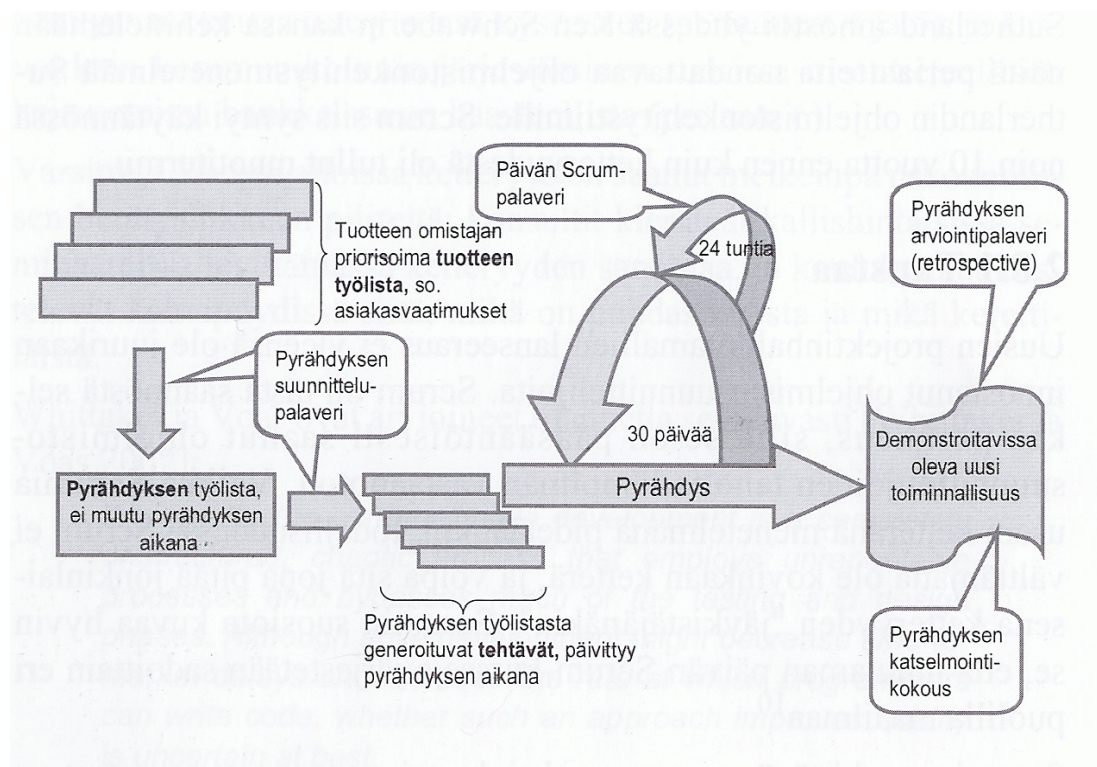
Ketterät kehittämismallit soveltuvat parhaiten pienehköihin kehittämishankkeisiin. Laajemmissa, kuten kansainvälisissä hankkeissa, ketteriä menetelmiä voi olla vaikea soveltaa. (Haikala & Mikkonen 2011, 45-46; Haikala & Märijärvi 2004, 47.)

Extreme programming (XP)

Extreme Programming eli XP on menetelmä, jonka tarkoituksena on parantaa ohjelmiston laatua ja valmiutta vastata asiakkaan muuttuviin vaatimuksiin. Menetelmä perustuu lyhyisiin kehityssykleihin ja nk. tarkistuspisteisiin, jolloin ohjelmistoon voidaan tehdä muutoksia muuttuneiden vaatimusten pohjalta. (Wikipedia 2015j.)

XP:lle ominaista on mm. jatkuva testaaminen ja nk. pariohjelmointi. Pariohjelmoinnissa toinen henkilö kirjoittaa lähdekoodia, toinen katselmoi koodia. (Haikala & Märijärvi 2004, 47; Wikipedia 2015k.) Asiakkaan halutaan olevan yksi ohjelmistoa rakentavan projektiryhmän jäsen (Haikala & Mikkonen 2011, 154).

Scrum



Kuva 6. Scrum (Haikala & Mikkonen 2011, 48)

Tiivistäen Scrum (kuva 6) tarkoittaa menetelmää, jolla hallitaan hankkeen toteutusvaiheen iteraatioita. Niiden pituus voi olla esimerkiksi kaksi viikkoa. (Haikala & Mikkonen 2011, 47-51; Kasurinen 2013, 28.) Scrumia voi tarkastella toisiaan seuraavien vesiputousten sarjana (Haikala & Mikkonen 2011, 37).

Yksittäisen iteraation, jota kutsutaan sprintiksi eli pyrähdykseksi, tavoitteena on tuottaa uusia ohjelmiston toiminnallisuuksia, joita voidaan nk. demota eli esitellä asiakkaalle ja muiden sidosryhmien edustajille. Sidosryhmien edustajat arvioivat tämän jälkeen toiminnallisuuksia, joista he antavat palautteensa. (Haikala & Mikkonen 2011, 47-51; Kasurinen 2013, 29.)

Lean ja Kanban

Lean on johtamisfilosofia, joka perustuu japanilaisen autonvalmistajan Toyotan alun perin kehittämään tuotantomenetelmään (Toyota Production System). Se on yhdistelmä johtamistaitoja ja käytäntöjä. (Wikipedia 2015l; Wikipedia 2015m.)

Ohjelmistokehityksessä Leanin oppeja sovelletaan seuraavien keskeisimpien periaatteiden pohjalta (Haikala & Mikkonen 2011, 54-55):

1. Kaikki, mikä ei tuota lisäarvoa asiakkaalle tai mitä tehdään varastoon, on hukkaa.
2. Arvosta ihmisiä. Keskity ihmisiin, jotka tuottavat lisäarvoa.
3. Panosta tehokkaaseen viestintään ja hyvään työilmapiiriin.
4. Panosta jatkuvaan kouluttamiseen ja toiminnan tehostamiseen.

Kanban (sana on japania ja tarkoittaa taulua tai kylttiä) on Lean-periaatteita noudattava työnhallintatapa. Se soveltuu hyvin esimerkiksi ohjelmiston ylläpitotehtävien hallitsemiseen. (Haikala & Mikkonen 2011, 55; Wikipedia 2015n.) Kanban-menetelmän sisältö on seuraava (Haikala & Mikkonen 2011, 55; Wikipedia 2015n):

1. Visualisoi työnkulku. Sen ilmaisemiseen käytetään tehtävätaulua. Siihen on merkitty jokaista työvaihetta kohti oma sarakkeensa, kuten ei aloitettu, aloitettu, tarkastuksessa ja valmis. Tehtävät on merkitty kortteihin. Korttia siirretään tehtävätaulussa sen mukaan, miten korttiin merkityn tehtävän suorittaminen etenee.
2. Yksittäisessä vaiheessa saa olla kerrallaan vain rajallinen määrä tehtäviä.
3. Mittaa läpimenoaika. Kehitysprosessia voidaan optimoida mittaamalla, paljonko aikaa yksittäisen tehtävän suorittamiseen kuluu kussakin vaiheessa.

2.3 Ohjelmistotuotannossa käytettävät työkalut

Ohjelmistojen kehittämisessä tarvitaan omia työkaluohjelmia. Yhteisnimitys näille työkaluille on CASE (Computer Aided/Assisted Software Engineering). (Haikala & Märijärvi 2004, 83; Wikipedia 2015a; Wikipedia 2015o.) CASE-työkaluja käytetään esimerkiksi (Haikala & Märijärvi 2004, 83; Wikipedia 2015a; Wikipedia 2015o)

- vaatimusmäärittelyssä (esim. käyttötapausten mallintaminen)

- ohjelmiston suunnittelussa ja toteutuksessa (esim. moduulien mallintaminen, lähdekoodin kehitysympäristöt ja koodin versionhallinta)
- ohjelmiston testaamisessa
- projektinhallinnassa.

2.4 Muutoksen johtaminen yleisesti

Organisaatioiden muutokset liittyvät yleensä taloudelliseen tilanteeseen, palveluihin, henkilöresursseihin sekä rakenteellisiin uudistuksiin. Muutokset voivat koskea myös organisaation toimintaa, työtapoja tai henkilöstön osaamispotentiaalia ja sen löytämistä. (Keva; Kotter 1996, 3.) Muutoksia toteutetaan, jotta organisaatio saavuttaisi paremman tilanteen. Siihen tähdätään esimerkiksi tuloksellisuuden, tuottavuuden, innovatiivisuuden tai työhyvinvoinnin parantamisella. Organisaation ylimmän johdon tehtävänä on kuvata muutoksen suunta, tavoitteet ja vaiheistus. (Kauppalehti 2014; Keva; Pirinen 2014, 62.)

Ihmisten johtamisen pitää olla yksilöllistä. Siksi esimiestyön laadulla on suuri merkitys muutosten toteuttamisessa. Muutosprosessin aikana esimiehen tulisi käyttää 70 % työajastaan ihmisten johtamiseen. Pelot, toiveet ja mahdollisuudet ovat osa muutoksia. Esimiehen on kyettävä käsittelemään näitä ajatuksia kyetäkseen johtamaan ihmisiä keskellä muutosprosessia. Osa ihmisistä kokee muutokset uhkana, osa mahdollisuutena esim. vastuun kasvattamiselle. Uhka liittyy usein tunteeseen, jonka mukaan muutos tarkoittaa jostakin luopumista. Sen vuoksi johdon ja lähiesimiesten on tarkkailtava ihmisten reaktiota yksilöllisesti. Henkilöstön sopeutumista helpottaa muutosten hyödyistä ja niiden myönteisistä vaikutuksista keskusteleminen sekä muutosten negatiivisten vaikutusten avoin käsittely. (Kauppalehti 2014; Keva; Pirinen 2014, 61-64; Ponteva 2012, 15-16.)

Muutoksen johtaminen tarkoittaa yhteisen ymmärryksen ja oppimisen rakentamista. Tällöin kaikki organisaation jäsenet saadaan osallistumaan muutosten toteuttamiseen: toteutuksessa tarvitaan ihmisten osaamista ja innovatiivisuutta. Muutosten juurtuminen on mahdollista vain sitouttamalla kaikki ihmiset muutosten toteuttamiseen. (Keva; Kotter 1996, 146.) Suunnitelmista on siirryttävä ripeästi tekoihin: tekeminen vaikuttaa ihmisten ajatteluun, mikä vuorostaan vaikuttaa ihmisten käyttäytymiseen (Kauppalehti 2014). Menestyksekkäs muutosjohtaminen vaatii (Juuti & Virtanen 2009, 146, 151-153; Kauppalehti 2014; Kotter 1996, 158-160; Mattila 2008, 29, 92; Pirinen 2014, 70-71):

1. Vahvoja esimiestaitoja (mm. avoimuus, kuunteleminen ja kehittymisen halu).
2. Aktiivista vuorovaikutteista viestintää, muutosprosessin alusta loppuun saakka.
3. Tietämystä organisaation toiminnasta ja toimintaympäristöstä.
4. Kykyä tunnistaa ilmiöitä, joilla on suurin vaikutus organisaation toimintaan.
5. Aiempien muutoshankkeiden virheistä oppimista.

Osa muutosprosessia on vanhoista käytännöistä luopuminen. Silloin pitää päättää, mistä käytännöistä luovutaan ja miten. Vanhoista käytännöistä irtaantuminen on tärkeää, jotta uusia toimintamalleja on mahdollista istuttaa organisaatioon. (Kauppalehti 2014.)

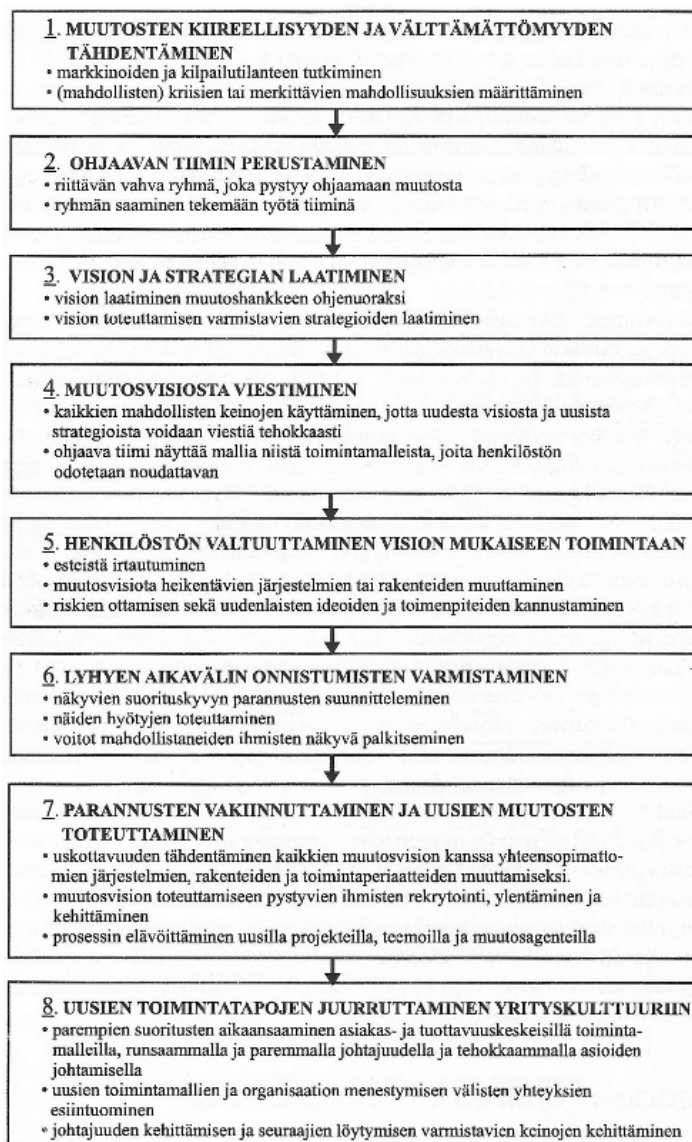
Muutosten onnistunut toteutus edellyttää tietoja ja taitoja organisaation, työyhteisön sekä yksilön tasolla. Muutoskykyisen organisaation tunnusmerkit ovat sekä oppimisen edellytyksiä että oppimisen tuloksia. (Ilmarinen, 23.) Tunnusmerkkejä ovat (Ilmarinen, 23-24; Kotter & Rathgeber 2014, 123, 128, 130; Mattila 2008, 29, 92-93; Ponteva 2012, 16):

1. Kyky pysähtyä. Muutos aiheuttaa kiirettä, joskus myös kaaosta. Kiireessä on vaikea tehdä valintoja ja suorittaa tehtäviä. Pysähtymistä tarvitaan, jotta ihminen kykenee rakentamaan ajatteluun: se auttaa ymmärtämään, valitsemaan ja toimimaan viisaammin. Muutokset synnyttävät uusia, yllättäviä tilanteita, joissa voi olla vaikea valita parasta tapaa toimia. Siksi tarvitaan yhteisiä tilaisuuksia (esim. muutosfoorumia), joissa ihmiset voivat yhdessä jakaa ajatuksia ja siten parantaa kykyään ymmärtää muutosta. Ajattelutapojen muuttaminen voi auttaa ihmisiä muuttamaan käytöstään. Se edellyttää tiedon keräämistä ja analysointia sekä tiedon esittämistä loogisesti. Tunteilla on erittäin suuri merkitys: niiden muuttumisella voi olla ihmisen käyttäytymiseen vielä suurempi vaikutus kuin ajattelutapojen muutoksilla.
2. Puheeksiottaminen ja rakentava vuorovaikutus. Muutostilanteissa tarvitaan kuuntelemisen ja kysymisen taitoa sekä ideointia, arviointia, kyseenalaistamista ja ratkaisukeskeistä keskustelua. Tavoitteena on yhteinen ymmärrys. Tällöin on tärkeää arvostaa toisten ihmisten ajatuksia ja keskittyä kuuntelemiseen.
3. Luopuminen, uuden aloittaminen ja priorisointi. Uuden vaiheen aloittaminen edellyttää ihmiseltä kykyä luopua vanhasta. Kyky aloittaa uutta edellyttää toisaalta oikeutta epäonnistua. Muutos on kuormittava tekijä: tilanteesta selviytyminen vaatii työtehtävien priorisointia. Priorisointiin on tärkeää saada apua esimieheltä.
4. Keskeneräisyys ja yksilön toimintakyvyn säilyttäminen. Muutoksen keskellä asioiden valmistuminen vie oman aikansa. Uusia töitä aloitetaan, vaikka aiemmat ovat kesken. Siksi keskeneräisyyden sietäminen on tärkeä taito. Se auttaa osaltaan ymmärtämään ja hyväksymään muutoksen tavoitteita sekä vahvistaa uskoa tulevaisuuteen.
5. Valmius oppia uutta ja osaamisen vahvistaminen. Muutosten toteutumisen rinnalla ihmisen pitää olla valmis oppimaan uutta. Se vaatii aikaa, energiaa sekä myönteistä asennoitumista uutta kohtaan. Työntekijöiden tulee pohtia osaamisensa kehittämistarpeita, samalla työnantajan on turvattava henkilöstölle mahdollisuudet oppimiselle ja osaamisen vahvistamiselle. Luottamus osaamiseensa ja sen kehittämiseen antavat työyhteisölle voimaa selviytyä muutostilanteista.

2.5 Esimerkki muutoksen johtamismallista: John Kotterin muutosprosessi

Yhdysvaltalainen tohtori ja Harvardin kauppakorkeakoulun emeritusprofessori John Kotter on yksi arvostetuimmista muutosjohtamisen asiantuntijoista. Hän on lisäksi mm. liikkeenjohdon konsultti ja kirjailija. (Kotter 1996, 163; Wikipedia 2017.)

Kotterin tunnetuimpiin teoksiin lukeutuu Muutos vaatii johtajuutta (*Leading Change*). Sitä pidetään merkkiteoksena muutosjohtamista käsittelevässä kirjallisuudessa. Kirja kuvaa muutoksen toteuttamista kahdeksanvaiheisena prosessina. (Wikipedia 2017.) Siihen sisältyy tärkeä periaate: yhtäkään prosessin vaihetta (kuva 7) ei saa jättää pois (Kotter 1996, 72). Prosessin vaiheet on kuvattu tarkemmin alaluvuissa 2.5.1-2.5.8.



Kuva 7. Kahdeksanvaiheinen muutosprosessi (Kotter 1996, 18)

2.5.1 Luo kiireen ja välttämättömyyden tuntu

Mittavien muutosten aikaansaaminen edellyttää, että suuri osa (n. 20 %) organisaation ihmisistä osallistuu ja sitoutuu niiden toteuttamiseen. Se vaatii kyseiseltä ryhmältä huomattavasti korkeampaa työsuoritusta heidän normaalitasoonsa verrattuna. (Kotter 1996, 31.)

Kiireellisyiden ja välttämättömyyden painottamista tarvitaan yhteistyön rakentamiseksi, jotta mahdollisimman suuri osa ihmisistä ymmärtää muutosten tarpeellisuuden (Kotter 1996, 31-32; Kotter & Rathgeber 2014, 125). Yleinen este muutosten toteuttamiselle on organisaatiossa vallitseva tyytyväisyys olemassa olevaan tilanteeseen. Tyypillinen virhe on se, että organisaatiossa keskitytään huonojen tulosten ilmaantuessa keskinäiseen syytelyyn. Suurin este muutosten käynnistämiseksi onkin usein se, että ongelmien olemassaoloa ei myönnetä, eikä ongelmista kyetä avoimesti keskustelemaan. (Kotter 1996, 32-34.) Organisaatiossa vellova tyytyväisyyden tila johtuu esimerkiksi seuraavista tekijöistä (Kotter 1996, 34-37):

1. Näkyvää kriisiä ei ole käynnissä, eikä (vielä) näköpiirissä.
2. Organisaatiokulttuurissa ei hyväksytä avoimuutta, huonojen uutisten esittämistä ja ristiriitojen esille tuomista. Konfliktitilanteiden välttely on tärkeämpää kuin avoin, rehellinen keskustelu.
3. Johtamiskulttuurissa korostetaan varovaisuutta ja asioiden johtamista, ihmisten johtamisen sijaan.
4. Johdossa vallitsee itsetyytyväisyyden, jopa ylimielisyyden ilmapiiri. Johto korostaa saavutuksia, jotka kuuluvat menneisyyteen. Johdolla heikko kokonaiskuva organisaation tilanteesta. Johdon mielestä syitä muutokseen ei ole, koska työntekijät eivät havaitse uhkia. Kokonaisuudesta kantaa vastuun vain yksi henkilö, kuten toimitusjohtaja: vastuu ei jakaudu ja siksi vastuun kantamista vältellään. Organisaatiorakenne ohjaa työntekijät keskittymään oman yksikkönsä tavoitteisiin, jolloin kokonaiskäsitys organisaation suorituksista ja tuloksesta hämärtyy.
5. Organisaation suoritusnormit ovat alhaiset ja suoritustasoa mitataan virheellisesti. Osasyynä tähän voi olla se, että yksiköiden tavoitteet on kuvattu epämääräisesti.
6. Suoritustasosta ei saada riittävästi palautetta ulkopuolelta.
7. Ihmisluonteeseen kuuluva kieltämisen kyky. Tämä korostuu kiire- ja stressitilanteissa.

Aiempi menestys luo organisaatioon itsetyytyväisyyden ilmapiiriä. Se vuorostaan estää ymmärtämästä muutostoimenpiteiden tärkeyttä. Siksi itsetyytyväisyyden ilmapiirin voimaa ei saa aliarvioida. (Kotter 1996, 36-37.) Itsetyytyväisyyttä murretaan rohkeilla teoilla, esimerkiksi (Kotter 1996, 37-41)

- luomalla tietoisesti kriisejä, kuten sallimalla virheiden laajat vaikutukset tai antamalla virheellisten suunnitelmien toteutua (jätetään puuttumatta)
- nostamalla suoritustasoa siten, että nykyisellä tasolla ei voida saavuttaa uusia tavoitteita
- levittämällä avoimesti tietoa henkilöstölle organisaation tilanteesta suhteessa kilpailijoihin
- vaatimalla ongelmien avointa käsittelyä
- levittämällä tietoa organisaation kyvyttömyydestä hyödyntää olemassa olevia mahdollisuuksia.

Periaatteessa muutostoimenpiteiden edistämisessä ja niiden tarpeellisuuteen herättämisessä ovat ylimmän johdon jäsenet avainasemassa. Mutta myös keskijohdolla voi olla muutosten käynnistämisessä suuri rooli. Kyseisten esimiesten kannattaa muistaa, että yleensä osa ylimmästä johdosta on muutoksia kannattavia henkilöitä ja he tarvitsevat keskijohdolta tullutta palautetta muutoshankkeiden perustelemiseksi. (Kotter 1996, 41-42.) Muutosten käynnistäminen yleensä edellyttää, että niitä kannattavat enemmistö henkilöstöstä, 75 % kaikista johtavassa asemassa olevista sekä kaikki ylimmän johdon jäsenet (Kotter 1996, 43).

2.5.2 Perusta ohjaava tiimi

Yksi ihminen ei kykene toteuttamaan muutosprosessin kaikkia vaiheita, kuten luomaan oikeaa visiota, edistää sen sisäistämistä organisaatiossa ja poistamaan esteitä muutosten tieltä sekä kyetä tekemään suuria määriä päätöksiä. Siksi tarvitaan muutoksia ohjaavaa, vahvaa tiimiä. Sen perustajan pitää aidosti ymmärtää ryhmän olemassaolon välttämättömyys. Tiimin jäsenten pitää luottaa toisiinsa ja heillä pitää olla yhteinen tavoite sekä halu tavoitteen saavuttamiseksi. Heidän välillään pitää olla keskinäistä kunnioitusta ja välittämistä. (Kotter 1996, 45-46, 54, 56.) Ryhmän koko on, organisaation koosta riippuen, 5-50 henkilöä (Kotter 1996, 52).

Tiimin jäsenissä on oltava riittävä määrä johdon edustajia (esim. yksi neljästä ylimmästä johtajasta), jotta muutosten toteuttamisen takana on tarvittava vaikutus- ja päätösvalta. Lisäksi johdon edustajien tehtävänä on sitouttaa muita tiimin jäseniä sen työskentelyyn. Ilman sitoutumista yhteinen työskentely jää puutteelliseksi. Se vaikeuttaa ongelmien ja mahdollisuuksien hahmottamista, päätöksentekoa sekä muutostoimenpiteiden toteuttamista. Tiimin kokoonpano pitää olla kattava, jotta siitä löytyy auktoriteettia, johtajuutta, erilaisuutta ja päätöksenteon pohjaksi riittävästi asiantuntemusta (esim. substanssietämystä ja työkokemusta) sekä tietoa. Ne luovat uskottavuutta tiimin toimintaan. Tiimin pitää

myös pystyä käsittelemään nopeasti suuria tietomääriä. (Kotter 1996, 47-48, 50-51, 54; Kotter & Rathgeber 2014, 125.) Iso haaste ohjaavan tiimin toiminnalle on sen kyky reagoida nopeisiin toimintaympäristön muutoksiin (Kotter 1996, 49). Onnistuakseen työssään tiimin pitää olla tasapainoinen yhdistelmä sekä ihmisten että asioiden johtamisosaamista. Asiaosaamisella ohjataan prosessia, ihmisten johtamistaidoilla johdetaan muutosta. Jälkimmäinen tarkoittaa mm. kykyä antaa valtuuksia toisille ihmisille. Ihmisten johtamistaito on kriittistä osaamista muutoshankkeen onnistumisen kannalta. (Kotter 1996, 50-52.)

Koska jäsenten keskinäinen luottamus on ryhmän toimintakyvyn perusta, heidän pitää oppia tuntemaan toisensa. Tuntemusta rakennetaan yhteisissä tapaamisissa. Niissä ihmiset kertovat avoimesti näkemyksistään organisaatiosta, sen mahdollisuuksista ja ongelmista. Näistä näkemyksistä keskustellaan. Ymmärrys ja sitä kautta luottamus kasvavat näiden keskusteluiden myötä. (Kotter 1996, 55.)

2.5.3 Laadi visio ja strategia

Osa johtamisesta pohjautuu autoritäärisyyteen eli henkilön vaikutusvaltaan sekä mikrojohtamiseen eli pikkumaisiin, yksityiskohtaisiin käskyihin sekä niiden noudattamisen tarkkaan valvontaan nojaavaan johtamismalliin. Ne ovat perua 1800-luvun yritysjohtamisesta. Näiden johtamismallien tarkoituksena on olemassa olevien rakenteiden säilyttäminen. Ne eivät sovellu muutoksen johtamiseen. (Kotter 1996, 59-60.)

Vallitsevaa tilannetta ylläpitävien voimien murtamiseksi tarvitaan visiota. Onnistunut muutosjohtaminen perustuu visioon ja sen jalkauttamiseen. Se on merkittävän johtajuuden perusta. Visiolla tarkoitetaan tulevaisuudenkuvaa, jolla pyritään - suorasti tai epäsuorasti - ohjaamaan ihmisten toimintaa jonkin tavoitteen saavuttamiseksi. (Kotter 1996, 59-60.) Muutosprosessissa vision tehtävänä on (Kotter 1996, 60-61; Kotter & Rathgeber 2014, 125):

1. Kuvata ja selkeyttää muutoksen yleistä suuntaa. Se toimii pohjana yksityiskohtaisille päätöksille.
2. Perustella ihmisille tavoitteita ja niiden saavuttamiseksi tehtäviä toimenpiteitä.
3. Kertoa uhkista, joita muutosten toteuttamatta jättämisestä seuraa.
4. Kertoa eduista ja hyödyistä, joita muutoksilla saavutetaan => toivon rakentaminen ihmisille.
5. Helpottaa ihmisten toiminnan koordinoimista kannustamalla ja ohjaamalla ihmisiä oikeansuuntaisiin tekoihin.
6. Ohjata päätöksentekoa tukemaan muutosten toteutusta. Tällöin voidaan esimerkiksi estää tai keskeyttää kehityshankkeet, jotka eivät palvele muutosten toteuttamista.

Muutokset edellyttävät ihmisiltä yleensä uhrauksia (mm. resurssien vähentämistä, uusien taitojen opiskelemista ja työtehtävien vaihtamista), jotka eivät ole lyhyellä aikavälillä heidän etujensa mukaisia. Siksi, inhimillisestä näkökulmasta, toivon luomisella on suuri merkitys. Se auttaa murtamaan muutoksia vastustavaa ajattelua. (Kotter 1996, 61.)

Visio on osa menestyksekkään muutoksen kokonaisuutta, jonka muut osat ovat strategiat, suunnitelmat ja budjetti. Kaksi ensiksi mainittua liittyvät ihmisten johtamiseen ja jälkimmäiset asioiden johtamiseen. Strategioilla toteutetaan visiota, suunnitelmia tarvitaan strategioiden toteuttamiseksi ja budjetti tarkoittaa suunnitelmien muuttamista rahassa mitattaviksi. (Kotter 1996, 62.) Tehokas visio on (Kotter 1996, 62-64, 67, 72)

- sisällöltään yksinkertainen, jopa arkipäiväinen => helppo muistaa
- sisältö pystyttävä selittämään viidessä minuutissa - kiinnostusta herättäen
- kuvaa tulevaisuutta ja tavoitteita - riittävän kunnianhimoisesti
- realistinen => tavoitteet on mahdollista saavuttaa
- kuvaa, *miten* tavoitteet on tarkoitus saavuttaa
- vetoaa sidosryhmissä (mm. henkilöstö, asiakkaat, osakkeenomistajat ja paikallisyhteisö) sekä pitkän aikavälin tavoitteisiin että etujen saavuttamiseen
- auttaa henkilöstöä erottamaan tärkeän toiminnan epäolennaisesta
- tukee päätöksentekoa
- joustava => sallii liikkumatilaa niin ajattelulle kuin teoille.

Tavallisesti vision ensimmäisen version laatii yksi henkilö. Sen jälkeen versiosta keskustellaan ja sen sisältöä muokataan ohjaavassa tiimissä. Tämä voi olla hyvin pitkä prosessi: se voi kestää kuukausista useaan vuoteen. Tuona aikana visiota kehitetään sekä järjellä että tunteella. (Kotter 1996, 69, 71.) Jotta visio olisi toteuttamiskelpoinen, se vaatii syvälistä tietämystä organisaatiosta, toimintaympäristöstä sekä kilpailutilanteesta. Myöhemmin, vision tavoitteita toteuttaessa, vaaditaan usein organisaatiolta myös nopeaa päätöksentekokykyä ja hierarkisuuden vähentämistä. (Kotter 1996, 66.)

2.5.4 Viesti muutosvisio

Visio motivoi ja ohjaa muutostoimenpiteiden toteuttamista. Sen vuoksi vision tulee luoda yhteistä näkemystä toivotusta tulevaisuudenkuvasta. Sen pitää olla helposti ymmärrettävä ja yhdistää organisaation eri sidosryhmiä. (Kotter 1996, 73-74; Kotter & Rathgeber 2014, 126.)

Yleinen syy vision välittämisen epäonnistumiseen on riittämätön viestintä. Tällöin vision sisältö hukkuu muuhun tietotulvaan. (Kotter 1996, 75.) Onnistuakseen viestinnän pitää noudattaa seuraavia periaatteita (Kotter 1996, 76, 79-85; Kotter & Rathgeber 2014, 133):

1. Käytetään selkeää ja yksinkertaista kieltä. Ammattisanasto ja teknokraattinen kieli karsitaan pois.
2. Käytetään kielikuvia (esim. sanoja, jotka symboloivat organisaation asemaa tulevaisuudessa), vertauksia ja esimerkkejä. Näiden tekijöiden pitää sopia kyseiseen organisaatioon. Niiden tulisi olla tunteisiin vetoavia, jolloin ne muistetaan helpommin. Tunteisiin vaikuttaminen on tärkeää, koska tunteet ohjaavat voimakkaasti ihmisten käytöstä.
3. Runsaasti viestintäkanavia, mm. suuria ja pieniä kokouksia, muistioita ja tiedotteita. Hyödynnettävä sekä virallista että epävirallista viestintää. Karsitaan pois muutosten kannalta epäolennaisia viestintää.
4. Paljon toistoa. Organisaation sisällä levitetään aktiivisesti vision sanomaa yhdistämällä se erilaisiin käsiteltäviin aiheisiin, kuten suunnitellun hankkeen hyötyjen tarkasteluun.
5. Muutoksen avainhenkilöiden toimittava esimerkkinä eli viestityn vision mukaisesti.
6. Ristiriitaisuuksien selvittäminen ja korjaaminen sekä niistä tiedottaminen avoimesti. Esimerkiksi puuttuminen johtajien epäeettiseen, visiossa kuvattujen tavoitteiden kannalta kyseenalaiseen käytökseen tai päätöksentekoon => lisätään luottamusta ja vahvistetaan vision uskottavuutta.
7. Vuorovaikutteisuus. Oikeus esittää kysymyksiä visiosta ja kyseenalaistaa se. Esimerkiksi henkilöstön antama palaute vision sisällön korjaamiseksi tai päivittämiseksi.

Jotta visio voisi toteutua, se tarvitsee työntekijöiden hyväksynnän. Ilman hyväksyntää muutosprosessin kaksi seuraavaa vaihetta eivät onnistu. (Kotter 1996, 85.)

2.5.5 Anna henkilöstölle laajat valtuudet

Kotterin (1996, 99) mukaan ”jos työntekijöillä on yhteinen päämäärä, sen toteuttamiseksi tarvittavien toimenpiteiden käynnistäminen on helpompaa”. Organisaatio ei kykene uudistumaan, mikäli henkilöstö ei pääse vaikuttamaan muutosten ideointiin ja toteuttamiseen. Tämä edellyttää valtuuksien sekä vastuun antamista työntekijöille. (Kotter 1996, 87.) Jotta henkilöstö kykenee osallistumaan muutosten toteuttamiseen, tarvitaan lisäksi muutoksia organisaation rakenteisiin, työntekijöiden taitoihin, tieto- ja hallinnollisiin järjestelmiin sekä esimiestyöskentelyyn (Kotter 1996, 88).

Rakenteellisten esteiden raivaaminen organisaatiossa tarkoittaa puuttumista esimerkiksi siiloutumiseen eli yksikköjen väliseen heikkoon yhteistyöhön. Organisaatorakenne voi vaikeuttaa yksikkörajat ylittävää yhteistyötä työntekijöiden välillä ja heidän ideoittensa edistämistä, mikä voi vuorostaan estää muutostoimenpiteiden toteutuksen. Siksi tarvittavat rakenteelliset muutokset tulee tehdä ajoissa. (Kotter 1996, 89-91; Kotter & Rathgeber 2014, 126.)

Muutosten aikaansaaminen merkitsee työntekijöille uusien tietojen ja taitojen oppimista. Yleinen virhe muutoshankkeissa on se, että osaamisen kehittämisessä tarvittavaa koulutusta järjestetään liian vähän, liian myöhään tai se on vääräntyyppistä. Pelkkä teknisten taitojen opettaminen ei riitä, vaan tarvitaan myös sosiaalisten taitojen vahvistamista ja asenteiden muuttamista. Lisäksi, koulutusten kautta syntyneiden uusien toimintamallien toteutusta pitää valvoa sekä pyytää henkilöstöltä palautetta, jotta toimintamalleissa esiintyvät ongelmat havaitaan ajoissa ja korjataan. Onnistuminen uusien toimintamallien käyttöönotossa vaikuttaa myönteisesti työntekijöiden asenteisiin ja sitä kautta muutosvision toteutumiseen. Antamalla valtuuksia työntekijöille vahvistetaan heidän ja työnantajan välistä luottamusta sekä työntekijöiden uskoa omaan kykyihinsä. (Kotter 1996, 92-93.)

Järjestelmien muuttamisella vaikutetaan siihen, miten työntekijät kokevat hyötyvänsä muutoshankkeesta. Järjestelmillä tarkoitetaan mm. suorituksen arviointia, palkkausjärjestelmää, ylennyksiä, työntekijöiden urapolkujen suunnittelua, rekrytointia, prosesseja (esim. strateginen suunnittelu) sekä organisaation johtamisessa käytettäviä tietojärjestelmiä. Niiden pitää kannustaa työntekijöitä ja tukea organisaatiota uudistusten aikaansaamisessa. (Kotter 1996, 94-95.) Tietojärjestelmiä on muutettava, jotta niiden sisältämiä tietoja voitaisiin käyttää hankkeen tulosten mittaamisessa (Kotter 1996, 109).

Alaisten käskyttäminen ja valvominen ovat yleisiä esimiesten piirteitä. Tällaiset esimiehet ovat esteitä organisaatioiden uudistamiselle. Heidän toimintaansa puuttumista (esim. työtehtävien vaihtamista, aseman muuttamista tai erottamista) rajoittavat usein johtoportaan henkilösuhteet. Heidän toimintatapoihinsa pitää kuitenkin puuttua rohkeasti heti muutoshankkeen alkuvaiheessa, koska heille on vaikeaa osoittaa luottamusta alaisiaan kohtaan ja myöntää alaisilleen valtuuksia sekä vastuuta. He voivat toiminnallaan myös tukahduttaa muutoshankkeissa tarvittavaa henkilöstön luovuutta sekä aloitteellisuutta. (Kotter 1996, 96-98.)

2.5.6 Lyhyen aikavälin onnistumiset

Suurissa muutoshankkeissa yleinen tilanne on se, että useita muutosprosessin vaiheita on käynnissä samanaikaisesti (Kotter 1996, 105). Muutoshankkeen onnistuminen koko-

naisuutena edellyttää myös aiempia eli lyhyen aikavälin saavutuksia. Niihin tulisi yltää noin puolen tai puolentoista vuoden sisällä hankkeen alkamisesta. Tämä tarkoittaa, että tuloksia on saavutettava jo muutosprosessin ensimmäisten vaiheiden aikana. Yksittäiset onnistumiset alkuvaiheessa eivät kuitenkaan riitä, vaan niitä tarvitaan koko hankkeen elinkaaren ajan. Lyhyen aikavälin onnistumisia tarvitaan luomaan uskottavuutta hankkeen kannattavuudelle. Kannattavuus sisältää mm. resurssien käytön arviointia suhteessa tuloksiin. (Kotter 1996, 102-105; Kotter & Rathgeber 2014, 126.)

Onnistumisten rakentamiseksi ohjaava tiimi laatii muutoshankkeelle välitavoitteet. Niiden toteuttamiseksi perustetaan projektit, joille valitaan ohjaavasta tiimistä vastuuhenkilö tai -henkilöt. Projekteille tehdään varaukset organisaation budjettiin ja varataan tarvittavat resurssit. Projektien edistymistä sekä tavoitteiden toteutumista seurataan säännöllisesti ohjaavassa tiimissä sekä organisaation johtoryhmässä (esim. kahden kuukauden välein). Välitavoitteisiin yltäminen ja säännöllinen konkreettisten, näkyvien tulosten esittäminen kannustavat ihmisiä muutostoimenpiteiden jatkamisessa. (Kotter 1996, 104, 107.) Lisäksi lyhyen aikavälin tulokset (Kotter 1996, 106)

- ohjaavat visioon sekä strategiaan tehtäviä muutoksia
- vaikuttavat keski- ja ylimmän johdon hankkeelle antamaan tukeen
- vaikuttavat hankkeen rahoitukseen (budjetointi)
- toimivat vastavoimana muutoshankkeen epäilijöille.

Lyhyen aikavälin tulosten aikaansaaminen aiheuttaa ihmisille paineita ja väsymystä, mahdollisesti myös turhautumista. Siksi prosessin edetessä on tärkeää motivoida ihmisiä. Esimerkiksi henkilöstölle tulee muistuttaa muutosten tarpeellisuudesta ja niiden päämäärän merkityksestä organisaatiolle. (Kotter 1996, 110.) Muutosprosessin ensimmäiset kuusi vaihetta toimivat perustana 7. vaiheessa toteutettaville suurille muutoksille. Yhteistä menestyksellisille muutoshankkeille on se, että niissä yhdistyvät sekä ihmisten että asioiden taitava johtaminen. (Kotter 1996, 112-113.)

2.5.7 Vakiinnuta parannukset, tuota lisää muutoksia

Suurten muutoshankkeiden riskinä on hiipua ennen kuin ne saavuttavat päämääränsä. Tilanne voi johtua esim. hankkeen avainhenkilöiden vaihtumisesta, johtajien väsymisestä tai organisaation vajoamisesta liialliseen tyytyväisyyteen, jossa ihmiset voivat tulkita muutostöiden tulleen tehdyiksi. Vaarana on tällöin se, että muutoksia vastustavat voimat voivat onnistua pysäyttämään prosessin ja sen aiemmissa vaiheissa tehty työ valuu hukkaan. (Kotter 1996, 115-116.)

Muutosprosessin ollessa kesken, sen aikana toteutetut muutokset ovat lähes poikkeuksetta hauraita sekä vakiintumattomia. Muutosten jatkamisen ja prosessin lopullisen tavoitteen hiipuminen estetään pitämällä muutosprosessin vauhti päällä. Mahdollinen tyytyväisyyden tunne ei pääse tuolloin leviämään, eikä rikkomaan ihmisten tietoisuutta prosessin kesken-eräisyydestä. (Kotter 1996, 117; Kotter & Rathgeber 2014, 126.)

Yritysten tilanne on muuttunut radikaalisti niiden kansainvälistymisen ja kilpailun voimistumisen myötä. Tässä muuttuneessa tilanteessa organisaatioiden yksiköiden on ollut pakko tiivistää yhteistyötään mm. tuotteiden valmistuksen nopeuttamiseksi. Samalla yksiköiden eri toimintojen välinen riippuvuus on kasvanut. Riippuvuus vaikeuttaa muutostöiden toteuttamista, koska yhden yksikön muuttaminen heijastuu välittömästi muiden yksiköiden toimintaan. (Kotter 1996, 117-118.) Organisaation toimintojen riippuvuuksista johtuvia ongelmia ratkaistaan (Kotter 1996, 118-119, 121-122):

1. Selkeyttämällä visiota (=muutosten tarkoitusta) ja strategiaa.
2. Määrittämällä syyt, jotka voivat estää vision toteuttamista yksiköissä, joiden toimintaa halutaan muuttaa. Välitetään uudistetun vision sisältö em. yksiköiden avainhenkilöille.
3. Keskittymällä keinoihin, joilla voidaan rikkoa tyytyväisyyden tilanne em. yksiköissä.
4. Toteuttamalla muutoksia huolella suunnitellussa järjestyksessä ja pienissä vaiheissa eli sarjana pieniä projekteja, yhden suuren projektin sijaan. Nämä projektit yhdistetään tarvittaessa muutosprosessin aiemmissa vaiheissa käynnistettyihin projekteihin.
5. Karsimalla tarpeettomia riippuvuuksia (esim. tehtaan uuden työntekijän palkkaamiselle vaaditaan lupa pääkonttorista) 5. vaiheessa kuvatuilla tavoilla, kuten rakenteellisilla muutoksilla. Riippuvuuksien karsiminen vaikuttaa tulevaisuuteen eli helpottaa tulevien muutoshankkeiden toteuttamista sekä organisaation uudistamista.

On tavallista, että edellä kuvatussa tilanteessa joudutaan toteuttamaan suunniteltua enemmän muutostoimenpiteitä. Niihin on myös jouduttu uhraamaan enemmän aikaa kuin oli arvioitu. Nämä kokemukset kuitenkin lisäävät tietoa, auttaen siten vastaavien toimenpiteiden suunnittelua jatkossa. (Kotter 1996, 122.)

Organisaation tasolla prosessin etenemistä turvataan lisäämällä muutosten toteutusvauhtia ja projektien määrää sekä tiukentamalla projektien tavoitteita. Hankkeen alkuvaiheessa suunnitellut suuret toimintamallien muutoksiin (esim. asiakaspalvelun kehittäminen ja kilpailukyvyyn vahvistaminen) keskittyvät projektit käynnistetään. Käynnistetään organisaation strategian uudistaminen. Samalla sekä päivitetään että toteutetaan 5. vaiheessa kuvattuja muutostarpeita, kuten henkilöstön kouluttamista, työntekijöiden ylennyksiä ja järjestelmien muuttamista. Prosessin seitsemännessä vaiheessa toteutetut toimenpiteet voi-

vat olla massiivisia, sisältäen esim. 10-20 samanaikaista muutosprojektia. Tällaisen kokonaisuuden onnistunut hallitseminen edellyttää, että johtajat delegoivat em. projekteihin liittyviä ihmis- ja asiajohtamisen tehtäviä alaisilleen organisaation eri tasoilla. Tällöin johtajat pystyvät keskittymään muutoshankkeen johtamiseen. Siinä painottuu *leadership* eli ihmisten johtaminen. Sen lisäksi johtoryhmä vastaa muiden päälinjausten suunnittelusta hankkeen eri osa-alueilla. Näin varmistetaan, että johtoryhmä kykenee johtamaan kokonaisuutta ja samalla tukemaan muutosprojektien läpivientiä. (Kotter 1996, 122-124.) Muutosprosessin 7. vaihe voi ääritapauksessa kestää vuosikymmenen verran. Vaihe voi vaatia satojen tai tuhansien ihmisten osallistumista. (Kotter 1996, 125.)

2.5.8 Juurruta uusia toimintatapoja yrityskulttuuriin

Organisaation uusien toimintamallien vakiinnuttaminen ratkaisee muutosprosessin lopputuloksen. Tavoitteeseen yltäminen edellyttää näiden toimintamallien istuttamista organisaatiokulttuuriin. Arvot ovat osa kulttuuria. Uusien toimintamallien tulisi olla yhteensopivia kulttuurin kanssa. Muutosprosessin viimeisessä vaiheessa onnistumista punnitaan siten, pystyvätkö uudet toimintatavat korvaamaan kulttuuriin juurtuneet vanhat toimintatavat. (Kotter 1996, 127-128, 130, 132; Kotter & Rathgeber 2014, 127.) Tämän tehtävän onnistuminen riippuu ihmisten johtamistaidosta (Kotter 1996, 129).

Kulttuurilla tarkoitetaan ihmisten keskuudessa vallitsevia toimintanormeja sekä yhteisiä arvoja. Toimintanormit ovat toimintamalleja, jotka säilyvät niin pitkään kuin niitä opetetaan ihmisjoukon uusille jäsenille. Arvojen asema on pysyvämpi: ne säilyvät ihmisyhteisössä, vaikka sen jäsenet vaihtuvat. Siksi arvoja on vaikeinta muuttaa. Normeja ja arvoja kutsutaan myös sosiaalisiksi voimiksi. (Kotter 1996, 129-130.)

Kulttuuria voidaan käsitellä sekä organisaatiotasoisena että yksikkökohtaisena (esim. myyntiosasto) piirteenä (Kotter 1996, 130). Muutosprosessissa kulttuuri on haastava kohde, koska (Kotter 1996, 130)

- sitä on työlästä muuttaa
- se vaikuttaa voimakkaasti ihmisten toimintatapoihin
- se on näkymätöntä, mikä estää suoran puuttumisen.

Organisaatioiden kykenemättömyys uudistua johtuu osaltaan kulttuurista. Eli siitä, että ne rekrytoivat omaan kulttuuriinsa sopivia ihmisiä ja kasvattavat nämä ihmiset noudattamaan tiettyjä, organisaation hyväksymiä toimintatapoja. Siksi toimintatapojen kyseenalaistaminen tai niiden objektiivinen, kriittinen käsittely koetaan kielteisesti. (Kotter 1996, 130-132.)

Lähdettäessä uudistamaan organisaation kulttuuria muutosprosessissa pitää valita muutosta tukevat nk. avainarvot. Kulttuuria muutettaessa olennaista on se, että uusia arvoja ei voi opettaa ihmisille pakottamalla. Siksi ne pitää sovittaa organisaation vakiintuneeseen kulttuuriin. Toteutettavien muutosten perusteluiden pitää olla sekä rationaalisia että tunteisiin vetoavia. Jos organisaatiokulttuurin peruspilari on ollut asiakaslähtöisyys, kulttuurin muuttamisen perusteena voi olla asiakkaiden tarpeiden muuttuminen. Tällaisessa tilanteessa uusien toimintamallien tarpeellisuus ja hyödyt voidaan todistaa muutosprosessin aikana esim. asiakastytyväisyyskyselyiden avulla. (Kotter 1996, 132-134.) Hyötyjä korostamalla voidaan perustella, miksi muutos on tarpeen eli organisaation vanha kulttuuri ei nykytilanteessa enää toimi (Kotter 1996, 135). Kulttuurinmuutosta tuetaan myös esimerkiksi (Kotter 1996, 135)

- estämällä vanhojen arvojen ja normien levittäminen uusille työntekijöille
- ylentämällä ainoastaan niitä ihmisiä, jotka kannattavat muutoksia
- vaihtamalla organisaation keskeisiä, vanhoihin menetelmiin, normeihin sekä arvoihin tukeutuvia henkilöitä
- nimittämällä organisaation johtoon ihmisiä, joilla ei ollut kytköstä vanhaan kulttuuriin.

Kulttuuria voidaan muokata vähitellen eli prosessin yksittäinen vaihe kerrallaan. Vaikka kulttuurin muuttaminen on mahdollista aloittaa jo prosessin alussa, kulttuurin muutos ja uusien toimintatapojen juurtuminen voivat toteutua vasta muutosprosessin viimeisessä vaiheessa. Niiden aikaansaamisen edellytyksenä on, että uudet toimintamallit on osoitettu vanhoja paremmiksi. (Kotter 1996, 136-137.)

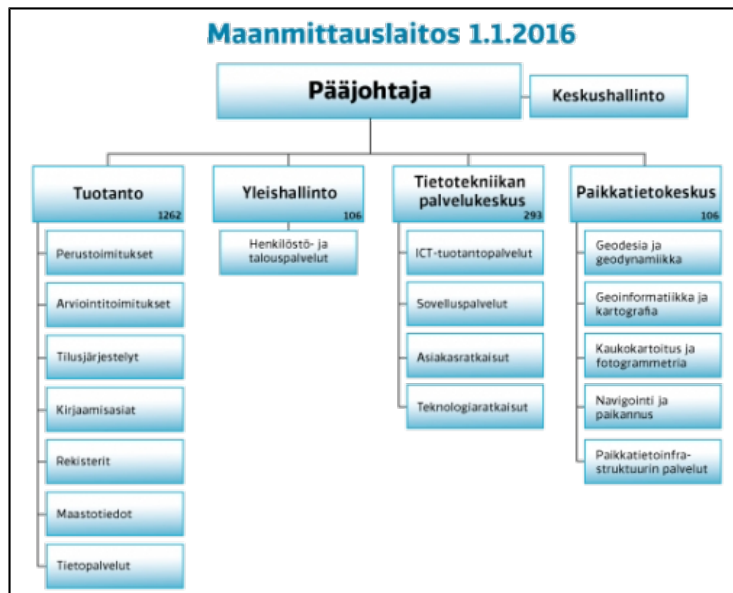
3 Opinnäytetyön kohdeorganisaatio ja kehittämismenetelmä

Tämä opinnäytetyö kohdistui Mitpan Vallilan toimipaikassa käytössä oleviin laadunvarmistuskäytäntöihin. Opinnäytetyön empiirisessä vaiheessa tutkin toimipaikan laadunvarmistuskäytäntöjen heikkouksia ja vahvuuksia. Lisäksi halusin kerätä henkilöstöltämme kehitysideoita. Keräämieni tietojen avulla halusin selvittää, millaisia muutoksia henkilöstömme haluaisi tehdä laadunvarmistuskäytäntöihimme.

3.1 Kohdeorganisaatio

Maanmittauslaitos on valtionhallintoon kuuluva virasto. Laitoksella on yli 200-vuotinen historia. Maanmittauksen ja Maanmittauslaitoksen juuret ulottuvat Ruotsin vallan aikaan ja vuoteen 1633. Silloin ensimmäinen maanmittari lähetettiin Ruotsista töihin Suomeen. Suomen siirryttyä Venäjän keisarikunnan alaisuuteen 1800-luvun alussa Suomen hallintorakenteisiin tehtiin muutoksia. Venäjän keisari vahvisti Päämaanmittauskonttorin perustamisen 14.11.1812. Päivämäärää pidetään Maanmittauslaitoksen syntymäajankohtana. (Maanmittauslaitos 2012; Wikipedia 2016a; Wikipedia 2008; Wikipedia 2016b.)

Maanmittauslaitoksen tehtäviin kuuluu maanmittaustoimitusten suorittaminen (esim. maa-alueiden lohkomiset) ja kartta-aineistojen tuottaminen. Virasto ylläpitää lainhuutoja, kiinnityksiä ja muita kiinteistöjen tietoja. Muita viraston keskeisiä tehtäviä ovat paikkatiedon tutkiminen ja soveltaminen sekä tietojärjestelmien kehittäminen. (Maanmittauslaitos 2016.)



Kuva 8. Maanmittauslaitoksen organisaatio (Maanmittauslaitos 2016)

Vuodenvaihteessa 2015 toteutetun Maanmittauslaitoksen, Tiken (Maa- ja metsätalousministeriön tietopalvelukeskus) ja Geodeettisen laitoksen yhdistymisen jälkeen Maanmittaus-

laitoksella on lähes 2000 työntekijää. Laitoksella on toimipaikkoja 37 paikkakunnalla, ympäri Suomea. Laitoksen organisaatio (kuva 8) koostuu keskushallinnosta ja neljästä toimintayksiköstä: Tuotanto, Yleishallinto, Tietotekniikan palvelukeskus ja Paikkatietokeskus. Tietotekniikan palvelukeskuksessa työskentelee vajaat 300 työntekijää. (Maanmittauslaitos 2015; Maanmittauslaitos 2016.)

3.2 Aineistonkeruumenetelmä ja sen perustelut

Sen jälkeen, kun olin valinnut opinnäytetyön kohteen eli toimipaikkani laadunvarmistuskäytännöt, valitsin avainkäsitteet, joiden perusteella keräisin materiaalin teoreettisen viitekehyksen kirjoittamista varten. Teoriaosan kirjoittamisessa käytin ohjelmointia, ohjelmistotuotantoa, laadunvarmistusta, IT-projektinhallintaa ja muutosjohtamista käsittelevää kirjallisuutta sekä painamattomia lähteitä (mm. korkeakoulujen materiaali ja Wikipedian artikkelit).

Aineistonkeruuseen valitsin kvalitatiivisen menetelmän. Aineiston keräämiseksi toteutin kaksi kyselyä (liite 1): ensimmäisen kyselyn sähköpostitse (kevät 2015) ja toisen kyselyn Webropol-työkalulla (talvi 2017). Jälkimmäinen kysely sisälsi ensimmäiseen verrattuna kaksi esitietokysymystä: vastaajan tulosityksikkö ja hänen tehtäväroolinsa. Alun perin olin laatinut kyselyyn kahdeksan aihetta. Tiivistin aiheiden määrän viiteen, jotta vastaaja ei kokisi vastaamista liian aikaa vieväksi. Valitsin kyselyyn avoimet kysymykset Likert-asteikon sijaan, koska halusin ohjata ihmiset itse miettimään ja ehdottamaan, miten he haluavat toimintaamme kehittää. Kyselyyn vastaaminen olisi työkavereilleni yksi kanava tuoda esille kehitysideoitaan.

Tutkimuskysymyksiin vastatakseni otin tarkasteltavaksi työkavereideni kirjaamat kehittämis ehdotukset sekä laadunvarmistuskäytäntöjemme heikkoudet ja vahvuudet. Ne olisivat hyödyllistä tietoa työnantajalleni ja tärkeää pohjatietoa jatkotoimenpiteiden suunnittelulle laadunvarmistuksen kehittämiseksi.

4 Opinnäytetyön tulokset

Tutkimukseen sisältyi kaksi kyselyä. Niistä ensimmäisen toteutin huhtikuussa 2015. Tuoloin lähetin sähköpostitse kyselyn Sovelluskehitys-vastuualueen työntekijöille. Jälkimmäisen, laajemman kyselyn toteutin tammi-helmikuun vaihteessa 2017 Webropolilla. Sen vastaanottajat oli valittu otantamenetelmällä siten, että tavoitteena oli kerätä kattavasti vastauksia eri tehtävärooleissa työskenteleviltä ihmisiltä. He edustivat neljää eri vastuualuetta Vallilan toimipaikassa, pois lukien edellä mainittu Sovelluskehitys-vastuualue, joka oli jo ollut kyselyn kohteena.

Ottaen huomioon kyselyiden vastaajien kokonaismäärän (19) ja kyselyiden vastausten analysointiin osallistuneiden määrän (11), opinnäytetyön aineiston tuottamiseen osallistui noin neljäsosa (~25 %) Mitpan Vallilan toimipaikan 115 työntekijästä. He edustivat 11:tä tehtäväroolia. Työnantajani päätöksen mukaisesti kyselyiden vastaukset ovat luottamuksellisia. Vastaukset löytyvät liitteistä 2 ja 3.

4.1 Kyselyn 1 vastaajat ja vastausaktiivisuus, kevät 2015

Kyselyyn vastasi neljä (4) henkilöä Sovelluskehitys-vastuualueen 16:sta työntekijästä.

Vastaajien profiilit olivat seuraavat:

- Yksi testausvastaava.
- Kaksi teknistä koordinaattoria.
- Yksi Scrum-mestari.

Lähettämäni kyselyyn (sähköposti) sain neljä vastausta, mikä tarkoitti 25 %:n vastausaktiivisuutta (4/16 henkilöä). Vastausaktiivisuus jäi alhaiseksi.

Olin kuitenkin erittäin tyytyväinen saamiini vastauksiin. Ne olivat pohdiskelevia, monesta näkökulmasta laadunvarmistusta käsitteleviä ja nostivat esiin kipupisteitä, joihin Tietotekniikan palvelukeskuksen johdon pitäisi kiinnittää huomiota. Myös esitutkimukseni palautteessa Soveltava tutkimus ja kehittäminen-opintojaksolla opettajani painotti kyselyn vastausten sisältöä ja sitä, että vastauksia oli saatu usealta tehtäväroolilta. Kyselyn tulosten analysointi toteutettiin Maanmittauslaitoksen Sovelluskehitys-vastuualueella työpajamuotoisesti helmikuun lopulla 2017.

4.2 Kyselyn 2 vastaajat ja vastausaktiivisuus, talvi 2017

Kyselyyn vastasi viisitoista (15) henkilöä 23:sta kyselyn vastaanottaneesta työntekijästä Vallilan toimipaikassa. Vastaajista kahdeksan (8) edusti Teknologiaratkaisut-tulosyksikköä ja seitsemän (7) Asiakasratkaisut-tulosyksikköä. Vastaajien profiilit olivat seuraavat:

- Kaksi ETL-kehittäjää.
- Kolme IT-palveluvastaavaa.
- Yksi ratkaisuarkkitehti.
- Kolme projektipäällikköä.
- Yksi raporttikehittäjä.
- Yksi sovellus- tai tietojärjestelmäarkkitehti.
- Yksi testaaja.
- Kolme tietokantavastaavaa.

Lähettämäni kyselyyn (Webropol) sain 15 vastausta, mikä tarkoitti 65 %:n vastausaktiivisuutta (15/23 henkilöä). Se on hyvä tulos tämän tyyppisessä kyselyssä. Kyselyn tulosten analysointi toteutettiin Maanmittauslaitoksen Sovelluskehitys-vastuualueella työpajamuotoisesti 2017 helmikuun lopulla.

4.3 Kyselyiden vastausten analyysi

Sovin esimieheni kanssa keväällä 2016, että Sovelluskehitys-vastuualueen työntekijät suorittavat analyysin kyselyiden vastauksista. Lähetin kyselyiden vastaukset ja niiden analysointipyynnön kysymyksineen vastuualueen henkilöstölle 20.2.2017. Analysoinnin tulosten keräämiseksi järjestin työpajan Sovelluskehitys-vastuualueen palaverin yhteydessä 28.2.2017. Työpajan aikana kirjasin ylös työntekijöiden suullisesti esittämät vastaukset heille etukäteen lähettämiini kysymyksiin. Kysymykset olivat seuraavat:

1. Mikä on mielestäsi laadunvarmistuskäytäntöjemme tärkein kehityskohde (=mitä pitää muuttaa tai mitä uutta pitää luoda)? Perustele valintasi. Kehityskohde voi koskea esim. työmenetelmiä tai työn organisointia.
2. Mikä on mielestäsi laadunvarmistuskäytäntöjemme suurin vahvuus (=mitä säilytetään)?

Osa vastuualueen työntekijöistä ei päässyt osallistumaan työpajaan. Siksi analyysin vastauksista osa on kerätty sähköpostitse.

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
1	<p>Vastausten perusteella katselmoinnissa vaikuttaisi olevan epäselvyyksiä, erityisesti uusille työntekijöille asia on jäänyt epäselväksi (eivätkä katselmointien tärkeys ja niiden tuomat hyödyt myöskään ole hahmottuneet). Se, milloin ja miten katselmoiteja tulisi tehdä tuntuu olevan epäselvää. Katselmointien tuonti sujuvaksi osaksi ketterän kehityksen menetelmiä olisi tärkeä kehityskohde, koska katselmoinnit edesauttavat laadukkaiden sovellusten kehittämistä. Tämä edellyttäisi sitä, että mietitään, miten saadaan tarvittavat resurssit asialle varattua (varaukset pitäisi olla Ressussa); lisäksi pitäisi saada käytännöt siihen, miten katselmointia tehdään ketterässä kehityksessä. Nykyiset toimintatavat sopivat esim. dokumenttien osalta paremmin vesiputousmallin toimintaan. Ehkä jonkinlainen jatkuva katselmointi, johon laskettaisiin käytettäväksi jonkin verran työaika? Esimerkiksi Confluence-dokumentteihin voisi kirjata mahdollisia puutteita aina kun niitä havaitaan, eikä siten, että yritetään käydä kerralla koko dokumentti läpi. Projektille pitäisi myös sopia vastuuhenkilö, joka huolehtisi Confluenceen tehtävistä päivityksistä. Samoin pitää katsoa sopivat välit, joissa tehdään koodikatselmoiteja ja tarkkaillaan jatkuvasti tilannetta automaattisissa seurantavälineissä. Palveluryhmämallin</p>	-	<p>Tähän asiaan kiinnitin huomiota vastausten joukossa: testaa- jien substanssin osaaminen ja perehtyneisyys järjestelmiin ja tietokantoihin oli selvästi arvokasta osaamista. Monet järjestelmät ovat niin monimutkaisia, että jos testaaja hallitsee vain tekniset asiat, on tehokas ja oikeisiin asioihin keskittyvä testaaminen hankalaa, oli automatisointia käytetty tai ei. Aika mielenkiintoinen asia maailmassa, jossa kaikki pitäisi tarvittaessa voida ulkoistaa. Paras yhdistelmä laadunvarmistuksessa vaikuttaisi siis olevan se, että henkilöllä on sekä teknistä että järjestelmien substanssin tuntemusta.</p>

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>myötä voisi ehkä olla mahdollisuus saada tätä selkiytettyä?</p> <p>1) Muutama kysymys:</p> <p>a) Onko testaajillemme missään vaiheessa ohjeistettu, miten tietokantoja pitäisi testata? Tai onko olemassa ohjeistus, jota voisi vain käydä läpi (siis muutakin ohjeistusta kuin pelkkä Journal-tilien tarkastaminen)?</p> <p>b) Ohjeiden koostaminen johonkin Confluencessa? Nyt ei varmaan kukaan tiedä, mistä eri ohjeistukset löytyvät, tai onko niitä ylipäätään olemassa. Ehkä voitaisiin ylipäätään koota linkkejä laadunvarmistuksen ohjeisiin vaikka testauskillan sivuille?</p> <p>2) Muita kommentteja:</p> <p>a) Malli tietoturvatestauksen suorittamiselle on ollut suunnitteilla jo jonkin aikaa, ja se kuuluu yhden palveluryhmämme tavoitteisiin lähiaikoina. Siitä ei ehkä ole vielä laajemmin kerrottu, mutta asia etenee kyllä.</p> <p>b) Tämä kommentti kuulosti mielenkiintoiselta: "Onko meillä yhtenäistä testauskäytäntöä, jonka perusteella kaikki sovellukset testattaisiin ainakin periaatteessa samalla tavalla? Jos on, niin sitä</p>		

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>ei juuri huomaa. Tuntuu, että sovellukset testataan eri tavoin: toiset tarkemmin, toiset vähemmän tarkasti. En tiedä, riippuuko tämä siitä, osaako asiakas vaatia."</p> <p>Kuulostaa aika mielenkiintoiselta. Ehkä tässä enemmän on ongelmana se, että toisissa järjestelmissä ei ole aikaa testata tarkemmalla tasolla ja toisissa siihen annetaan aikaa? Enemmän taitaa olla kyse projektipäällikön/palveluvastaavan toiminnasta: osataanko kertoa asiakkaalle, että testaukseenkin tarvitaan aikaa.</p> <p>c) Tietovarastojen testaus? Pitäisikö näihin perehtyneitä testaajia (tai muulla nimikkeellä toimivia) olla enemmänkin? Myös huomiot siitä, että toteuttajalla ei voi olla riittävää tietoa tietosisällöstä, herättävät kysymyksen, pitäisikö tietovarastoja testata eri tavoin kuin nyt tehdään?</p> <p>d) Määrittelyiden puutteet vaikuttavat olevan edelleen iso kysymys, vaikka kyselyiden välillä on ollut 2 vuotta.</p> <p>e) Käytettävyydestä oli hyvä kommentti: "Asiakkaat eivät ehkä tilauksissaan osaa ottaa kantaa käytettävyyden tärkeyteen. Näin se saattaa jäädä liian pienelle painoarvolle."</p>		

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>Tämä on juuri näin. Pyydetään tekemään se vanha lomake uudestaan sähköisesti ymmärtämättä, että juuri tässä tilanteessa olisi mahdollista saada muutoksia aikaan. Vaikka meillä olisi asiasta omaakin osaamista, ei sitä osata hyödyntää projekteissa, vaan turvaututaan aina konsultteihin. (Asiakkaalle helpompi myös saada uudet ideat läpi, jos ne esittää joku muu kuin oma työntekijä, mikä on aika surullista.)</p>		
2	<p>Mielestäni testausta ja laadunvarmistusta pitäisi integroida enemmän ohjelmakooditasolle.</p>	<p>- Testaajien olisi hyödyllistä osata ohjelmoida ja lukea sovel-luskoodia. Näin olisi helpompi ja nopeampi tehdä relevantteja testejä ja hyödyntää tehokkaammin automaatiotestausvälineitä.</p> <p>- Tarvitsisimme myös muutenkin lisää arkikitehtuuria ja ohjelmointia osaavaa henkilöstöä, jotta resurssit riittäisivät laadukkaiden koodikatselmointien tekemiseen. Koodikatselmointeja pitäisi tehdä nykyistä enemmän.</p>	<p>Suurin vahvuutemme lienee se, että testauksen merkitys ja tärkeys on meillä hyvin ymmärretty. Myös testausmenetelmien kehittämiseen panostetaan ja uusia välineitä on otettu käyttöön.</p>

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
3	Vastaukseni on: Code review.	<p>- Tämä on myös mainittu liitteenä olevassa dokumentissa ”1. Katselmoiteja pitäisi istuttaa enemmän mukaan kehitysprosessiin, testauksen tavoin. Kehitystiimin sisällä pitäisi toteuttaa kevyitä katselmoiteja uusien ominaisuuksien valmistuksessa.”</p> <p>- Mielestäni paras hetki tehdä code reviewiä on pienissä osissa. Esim. kehitystiimissä aina toinen kehittäjä lukee läpi toisen henkilön commitin ennen kuin se hyväksytään. Tällöin saadaan suurimmat ”purkkakoodit” pois, ja koodissakin keskitytään laatuun, kun joku toinen raukka joutuu sen lukemaan aina läpi, jolloin tehdään luettavampaa koodia.</p>	<p>- Testausta on automatisoitu menestyksekkäästi, tämä tulisi säilyttää ja jatkaa edelleen toimintaa.</p> <p>- Esim. Robot-testejä on tehty korvaamaan manuaalista työtä, näitä testejä ajetaan automaattisesti ja systemaattisesti Jenkinsistä, jolloin kerralla hyvin tehty testi korjaa tulevaisuudessa monen monta manuaalitestitapausta. Tämä myös pakottaa määrittelyjen/sovelluksen muuttuessa muuttamaan automaattitestejä, jolloin uutta määrittelyäkään ei voi kiireessä jättää testaamatta.</p>
4	Projekteissa, joissa tehdään uusia sovelluksia tai jatkokehityksessäkin, niin	-	Substanssitietämys.

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>katselmointia pitää tehdä eri vaiheissa ei vasta projektin valmistuttua.</p> <p>Määrittelyjen tekeminen jo heti alussa: "Yritetään usein miellyttää asiakasta liikaa, eikä uskalleta sanoa, että vaatimukset/määrittelyt vs. resurssit/aikataulu ei tule toimimaan. Jos kerran halutaan tehdä laatua, se on huomioitava heti alusta alkaen."</p>		
5	<p>Jo ihan aiempien kehittäjien tekemiä sovelluksia jatkokehittäneenä voisi todeta, että koodikatselmoinnissa olisi parantamisen varaa, mikä näkyy myös 2017 vuoden kyselyn tuloksissa.</p>	-	<p>Yksikkötesteihin on panostettu ja yksikkötestausta tulisi painottaa ja kouluttaa vieläkin enemmän. Se vähentäisi manuaalista työtä sekä osoittaisi heti, jos jokin hajoaa kehittäessä. Vähintäänkin nykyinen taso tulisi säilyttää.</p>
6	<p>Yleisiä puutteita: riippuvuuksien haku tietojärjestelmiä toteutettaessa, vanhentuneet määrittelyt ja puuttuva testausautomaatio. Mihin testausautomaation käyttöönotto perustuu eri tietojärjestelmissä?</p>	<p>Ohjelmistojen laadun parantaminen ja työn sujuvuus.</p>	<p>Halu ja mahdollisuus ottaa käyttöön uusia välineitä. Esimerkiksi Elvin ja EPR:n väki ollut säännöllisesti yhteydessä ja kyselty neuvoja CI-työkaluista.</p>
7	<p>Tietovarastojen testauksen kehittämätömyys.</p>	<p>Pitää kiinnittää huomiota, koska tämän osa-alueen kehittäminen vasta alkuvaiheessa. Tiedon laa-</p>	<p>Asenne testaukseen sovelluskehityspuolella ja ammattitaito niillä, joilla taitoa on.</p>

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
		tuun panostaminen.	
8	Testausautomaatio. On välineitä ja menetelmiä. Ketterä testaus = testausta ja laadunvarmistusta tulee tehdä töiden aloittamisesta lähtien.	Ei mene itsestään eteenpäin, vaan QA-ihmisillä on velvollisuus edistää uuden teknologian & menetelmien käyttöönottoa. Oma-aloitteisuus ja aktiivisuus edistää.	Löytyy CI-ympäristö, jossa valmiita työkaluja. On erikoistuneita henkilöitä, jotka osauttavat auttaa ympäristön hyödyntämisessä. Muitakin työkaluja: Confluence ja Jira. Ohjelmistotuotantomallista löytyy ohjeita, dokumenttipohjia ja malleja. Hyviä QA-käytäntöjä voisi esitellä esim. Maanantai-seurassa.
9	Miten testausautomaatiota on meillä otettu käyttöön ja miten automaatiota voidaan laajentaa? Entä suorituskykytestauksen, tietoturvatestauksen ja käytettävyydestestauksen tilanne?	Testauksen osalueiden käyttöönoton epäselvyys.	Substanssiosaaminen. Testaajat taitavia. Pitäisi tarjota myös aikaa tehdä työ hyvin.
10	<p>- Testauksen mukaantulo määrittelyvaiheessa jää liian vähäiseksi: testausuunnitelma saatetaan tehdä, mutta manuaalisten testitapausten teko jää ja tulee vastaan sitten, kun kehitysajon on jo "elänyt" ja ensimmäisen sprintin muutokset pitäisi testata. Voi olla, ettei kehitysajonossa tällöin ole ilmaistu riittävän selkeästi tai tarkalla tasolla sitä, mitä pitäisi testata.</p> <p>- Katselmointiin ei kukaan velvoita - päinvastoin tuntuu kuin siihen (varsin-</p>	-	<p>- Ohjelmistotuotantokäsikirja ja Confluncen sivupohja antavat hyvät raamit.</p> <p>- JIRA on kätevä työkalu.</p> <p>- Esimerkiksi Elvin automaatio-osaamista voisi nostaa esille. Elvissä ensin panostettiin Robot-testaukseen</p>

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>kaan koodikatselmointiin) ei saisi käyttää vähiä resursseja.</p> <p>Muuta:</p> <ul style="list-style-type: none"> - Miten ketterissä sprinteissä parhaiten tuotaisiin esiin, mikä on tilanne laadunvarmistuksen suhteen: daily scrumissa, demoissa? - Integraatiotestauksen sisältöä ja suorittamista voisi esitellä tarkemmin, esim. jossakin työpajassa. 		<p>ja sitten alettiin parantamaan määrittelyitä.</p>
11	<p>Tärkein kehityskohde on automaation kehittäminen. Tietojenkäsittelytiede ja sen sovellukset ovat kehittyneet riittävästi, että varsinainen ATK, eli automaattinen tietojenkäsittely voisi vihdoinkin lunastaa niitä toiveita, joita sille asetettiin jo vuosikymmeniä sitten. Manuaalisen työn vähentämistä pitää tehdä kaikkialla siellä, missä se on mahdollista ja järkevää. Tämä ei tarkoita ainoastaan testausautomaatiota, vaan myös kaikkea muuta työtä, jota teemme.</p> <p>Automaatio tuntuu herättävän Mitpassa monenlaisia tunteita, myös pelkoa ja epävarmuutta tulevaisuudesta. Toisaalta organisaatioiden kulttuureista ja muutosvastarinnasta tiedetään, ettei muutokset ole helppoja ja että niiden onnistumiseen vaikuttavat hyvin monet seikat. Automaatiota pitäisikin mielestä-</p>	<p>Perustelut edellisessä kohdassa. Jos aikataulut painaa päälle, panostetaan edes jonkin yksittäisen QA:n osa-alueen eteen.</p>	<p>Työkaluja ja työmenetelmiä olemassa. Voitaisiin nostaa esille projekteja, joissa on tehty hyvää QA-työtä.</p> <p>Laadunvarmistukselle on käsittääkseni johdon tuki ja sitä on lähdetty kehittämään. Ilman sitä ei työllä ole mitään mahdollisuuksia. Laatuun merkittävästi vaikuttavia automaatiotyökaluja on olemassa ja niitä voi ottaa käyttöön.</p> <p>Lopuksi vielä mainittava, että laatu kuuluu meille kaikille. Mikään</p>

#	1a. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde	1b. Laadunvarmistuskäytäntöjen tärkein kehittämiskohde, perustelut	2. Laadunvarmistuskäytäntöjen suurin vahvuus
	<p>ni ikään kuin markkinoida enemmän, työntekijöiden pitäisi nähdä automaatio tuomassa helpotusta omaan tekemiseen, eikä uhkana omalle positiolle tai ammattitaidolle.</p> <p>Resurssien puute ja jatkuva kiire. Tehdään hyvää laatua alusta pitäen -> esim. ensimmäisestä buildistä lähtien -> automaatio auttaa tässä.</p> <p>Vastauksista voi päätellä ja minusta muutenkin tuntuu, että Mitpassa tehdään vain hyvin pieni osa siitä työstä, joka pitäisi tulla tehdyksi (manuaalisesti tai automaatiota hyödyntäen), että järjestelmien laadukkuudesta voitaisiin todellisuudessa edes jossain määrin varmistua. Jos ajatellaan, että teemme vaikka (hypoteettisesti) vain 20 % kaikesta työstä, joka pitäisi tehdä, niin automaatiota hyödyntäen voisimme ylettyä samalla työntekijämäärällä vaikka 80 %:iin. Tämä olisi mahtava kehitys ja järjestelmiemme laatu paranisi ja uskoakseni myös viihtyvyys suurimmalla osalla, kun voisivat tehdä työtään hyvin. Tätä ajatusta voisi käyttää markkinointimateriaalina automaatiomyönteisen kulttuurin levittämisessä organisaatioon.</p>		<p>erillinen testaus- tai laatuporukka ei voi liimata laatua sovellukseen, vaan lopullinen laatu muodostuu pienistä valinnoista jatkuvasti jokaisessa prosessin vaiheessa.</p>

5 Johtopäätökset

5.1 Aineiston analyysi

Analyysini tutkimuksen tuloksista (kahden kyselyn vastaukset ja niistä tehdyt analyysit Sovelluskehitys-vastuualueella):

1. Määrittely ja laadunvarmistus. Määrittelyt:

- Kirjoitettu osittain pinnallisesti.
- Joissakin järjestelmissä vanhentuneita.
- Jätetty päivittämättä ylläpitovaiheessa.
- Joidenkin ihmisten päässä, ei dokumentoituna.
- Miten voidaan - tällaisten puutteiden vallitessa - odottaa laadukasta testausta?

2. Testaus.

Yksikkötestaus:

- Systemaattisuus testien tekemisessä puuttuu.
- Testejä on ajoittain vaikea tulkita.
- Jätetään tekemättä aikataulupaineiden vuoksi.

Integrointitestaus:

- Testaamisesta puuttuu järjestelmällisyys.
- Integrointitestauksen asema muuttuu yhä kriittisemmäksi järjestelmien monimutkaistuesssa ja rajapintojen määrän kasvaessa.

Järjestelmätestaus:

- Testausta vaikeuttavat keskeneräiset, puutteelliset ja vanhentuneet määrittelyt.
- Testausta pitäisi päästä tekemään nykyistä aiemmin eli heti toteutuksen valmistumisen jälkeen. Käytännössä tähän tavoitteeseen on vaikea päästä, koska järkevä testauskokoisuus yleensä edellyttää usean ohjelmisto-osan valmistumista.
- Testausta vaikeuttaa puuttuvat rajapintamäärittelyt. Ne pitäisi olla saatavilla kaikista järjestelmistä.
- Jos projektin aiemmat vaiheet ovat myöhässä, valitettavan usein myöhästymistä korjataan siten, että lyhennetään testaukseen varattua aikaa.

Hyväksymistestaus:

- Ei jää riittävästi aikaa → ei ehditä tekemään kattavaa testausta.

- Myös tätä testauksen vaihetta olisi mahdollisuus automatisoida.

Muut testaustehtävät:

- Testauksen automatisoinnissa on päästy etenemään. Tulisi arvioida järjestelmäkohtaisesti, milloin ja missä laajuudessa automatisointia kannattaa tehdä. Automatisointia ei kannata tehdä keskeneräisille toiminnallisuuksille, koska tällöin testejä joudutaan toistuvasti päivittämään, mistä seuraa ylimääräistä työtä.
- Kuormitustestaus tulee suorittaa kaikille laajoille järjestelmille. Niillä on mm. suuri yhtäaikainen käyttäjäkuorma ja paljon rajapintoja toisiin järjestelmiin.
- Tietoturvatestauksen osaamistamme pitäisi vahvistaa.

3. Käytettävyyden varmistaminen.

- Henkilöresurssit eivät ole riittävät.
- Loppukäyttäjän näkökulmaa ei huomioida riittävästi.
- Osa asiakkaista ei ymmärrä käytettävyyden merkitystä.

4. Katselmoinnit.

- Katselmoiteja ei toteuteta systemaattisesti.
- Osassa katselmoiteja ei noudateta katselmointiprosessia (voi johtua epäselvästä ohjeistuksesta). Esimerkiksi tuotoksen kommentit kuuluu kirjata ennen tarkastuskokousta ja kokouskutsu tulee lähettää ajoissa.

5.2 Kehittämisehdotukset kohdeorganisaatiolle

Keräämäni aineiston - kahden kyselyn vastaukset ja niistä tehdyt analyysit Sovelluskehitys-vastuualueella - pohjalta laadin Maanmittauslaitoksen Tietotekniikan palvelukeskuksen johdolle alla listatut kehittämisehdotukset Vallilan toimipaikan laadunvarmistuskäytäntöjen kehittämiseksi (ehdotukset 1-16). Osa ehdotuksista pohjautuu ohjelmistotuotannon työntekijänä tekemiini havaintoihin ja kehitysideoihini (ehdotukset 17-22). Kehittämisehdotuksista pääosa käsittelee Maanmittauslaitosta. Listauksessa käyttämäni numerointi on tarkoitettu kehittämisehdotusten yksilöimiseksi, eikä kuvaamaan ehdotusten tärkeysjärjestystä.

1. Laatuajattelu. Aineiston perusteella toimipaikassamme pitää kehittää kokonaisvaltaista laatuajattelua. Se tarkoittaa, että tietojärjestelmän laatuun vaikuttaa järjestelmän elinkaaren kaikissa vaiheissa tehty työ, alkaen esitutkimuksesta ja päättyen tuotantoonvientiin, viime vaiheessa tietojärjestelmän käytöstä poistamiseen. Tietojärjestelmän laatu ei ole siten pelkästään testaajien vastuulla, vaan laatuvastuu kuuluu sekä tilaajan että järjestelmätoimittajan henkilöstölle.

2. Toteutuksen ja testauksen laadun sekä työn sujuvuuden parantamiseksi nostettava määrittelyiden tarkkuutta. Ominaisuuksia pitää määrittellä tarkemmin, jotta sovelluskehittäjillä ja testaajilla on tarvittavat, riittävän yksityiskohtaiset tiedot ohjelmiston toteuttamista sekä testien suunnittelua ja suorittamista varten. Määrittelyt pidettävä ajan tasalla. Tilaajan pitää huomioida, että määrittelyiden pitää olla ajoissa valmiita. Prosessin tulee edetä siten, että ensin tehdään määrittely kirjallisesti eli dokumentoidaan järjestelmään tehtävä muutos. Vasta tämän jälkeen toteutetaan ja testataan. Tämä on ainoa keino pitää määrittelyt ajan tasalla, varsinkin ylläpitovaiheessa. Siis ei enää suullista "speksaamista". Sekä oman organisaatiomme että asiakkaiden henkilöstön pitää ymmärtää vastuunsa määrittelyiden tarkkuudessa: epäselvät ja puutteelliset määrittelyt heijastuvat kielteisesti tietojärjestelmien rakentamiseen ja ylläpitoon.
3. Yksikkötestien laatua parannettava. Yksikkötestien kattavuus (esim. lausekattavuus) ei kerro, testaavatko testit oikeita asioita. Yksi keino korjata tilanne on ottaa yksikkötestien tarkastaminen mukaan koodikatselmoiteihin.
4. Ketkä valvovat integrointitestausta? Huolellisuus integrointitestauksessa vaikuttaa erittäin paljon testiversion laatuun ja järjestelmätestaajien työhön. Tietojärjestelmien sisäisten ja ulkoisten riippuvuuksien laajentuminen lisää integrointitestauksen merkitystä. Se on jäänyt huomioissa järjestelmä- ja yksikkötestauksen varjoon.
5. Järjestelmätestaus. Järjestelmätestaukselle varattua aikaa ei saa lyhentää. Testaajat tulisi ottaa aikaisemmin mukaan projekteihin, mieluiten jo määrittelyvaiheessa. He pääsisivät tällöin perehtymään järjestelmien liiketoimintalogiikkaan ja testien suunnittelu päästäisiin aloittamaan nykyistä aiemmin.
6. Hyväksymistestaus. Asiakkaiden pitää kantaa vastuunsa hyväksymistestauksesta. Toisaalta Mitpalla on oltava myös valmius tukea asiakkaita hyväksymistestauksen suunnittelussa.
7. Muut testaustehtävät. Tietoturva- ja suorituskykytestaukseen sekä tietokantojen ja tietovarastojen testaamiseen toivottiin lisäpanostuksia. Niiden toteuttamiseksi tarvitaan nykyistä kattavampi ja yksityiskohtaisempi testausohjeistus.
8. Testauksen automatisointiin kannattaa panostaa. Työyhteisössämme on jatkossakin rajalliset henkilöresurssit. Koska järjestelmät muuttuvat laajemmiksi ja monimutkaisemmiksi, tarvitsemme laadunvarmistusvelvoitteidemme täyttämiseksi työvälineitä, jotka vähentävät manuaalista testaamista. Siksi tarvitsemme automati-

sointia. Osaamista, ohjeita, koulutusta ja henkilökohtaista opastusta on saatavilla, joten automatisointiin perehtymisen kynnyksen pitäisi olla alhainen.

9. Katselmointien säännöllinen toteuttaminen kiinteäksi osaksi ohjelmistojen kehittämistä. Riippumatta katselmoitavasta tuotoksesta (esim. sovelluskoodi, tietokantojen koodi, tietokantaobjektit ja määrittelydokumentit) ja katselmointien virallisesta asemasta ohjelmistotuotantoprosessissamme, katselmointien toteutuksessa on vakavia puutteita. Katselmointien tekemättä jättäminen heikentää työn laatua, lisää tietojärjestelmien kehittämisen kustannuksia ja viivästyttää töiden etenemistä sekä projekteissa että ylläpidossa. Dokumenttien osalta katselmoinnissa kannattaa hyödyntää Confluencen kommentointiominaisuutta, jolloin kommentteja - virhehavainto, kysymys, vastaus kysymykseen ym. - voi kirjoittaa suoraan dokumenttiin sitä luakiessa. Koodikatselmointien pitäisi olla jatkuva, joissakin tapauksissa päivittäinen tehtävä. Parikoodaus XP-menetelmää jäljitellen auttaisi löytämään ohjelmoinnissa tehdyt virheet nopeammin ja opettaisi sovelluskehittäjille oikeaa ohjelmointityyliä. Siksi parikoodaus todennäköisesti vähentäisi refaktoroinnin määrää.
10. Katselmointien seurantaan saatava ryhtiä. Seuranta on ollut aiempina vuosina organisoitua ja säännöllistä. Tuolloin katselmointien tilanteesta raportoivat testaajat/testausvastaavat ja raportointia valvoi Tiken testauspäällikkö. Aineiston perusteella seuranta on tuuliajolla. Tästä on seurannut mm. se, että katselmoiteja jää tekemättä ja katselmointikommenteista huolimatta tuotos jää korjaamatta. Katselmointiprosessiin ja sen ohjeisiin kaivattiin selkeyttämistä. Yksi keino tarpeen täyttämiseksi voisi olla katselmoinnin prosessikuvauksen lisääminen projektisuunnitelman dokumenttipohjaan Confluencen. Silloin kuvaus tulisi ehkä aktiivisemmin luettua kuin projektisuunnitelman liitteenä (=yleensä erillisenä tekstitiedostona).
11. Dokumentointi. Lean-ajattelun mukaisesti dokumentaatiota tulisi tuottaa vain se määrä, joka on riittävä toimimaan tietolähteenä laadukkaalle työskentelylle. Leanin mukaan dokumentaatiota ei tule "tuottaa varastoon". Dokumentoinnin riittävyttä pohdittaessa pitää kuitenkin muistaa se, että dokumentoinnin määrän vähentäminen ei saa tarkoittaa dokumentoinnin laadusta tinkimistä. Käyttäjävaatimusten tuottamiseksi on joissakin projekteissamme alettu kirjoittaa käyttäjätarinoita. Pienillä täydennyksillä niistä syntyisi valmiita testitapauksia yksikkö-, järjestelmä- ja hyväksymistestaukseen. Menetelmä edellyttää opiskelua sekä tilaajalta että toimittajalta, mutta se keventää dokumentoinnin työkuormaa ja nopeuttaa testauksen käynnistämistä. Omien kokemusteni pohjalta projekteissamme laaditut käyttäjätarinat nykyisessä muodossaan ovat pelkkiä korkean tason kuvauksia. Ne eivät kuvaa tietojärjestelmän ominaisuuksia riittävällä tarkkuudella eli yksityiskohtaisesti. Siten

ne eivät - tällä hetkellä - tue sovelluskehittäjän ja testaajan työtä ohjelmistokomponenttien sekä testien suunnittelussa. Toinen kehittämiskohde koskee raportointia. Osa ohjelmistotuotantomme työvälineistä (esim. Jira ja Ressu) sisältää raportointitoiminnon. Sitä pitää osata hyödyntää. Raporttien tekeminen käsin on Leanin vastaista eli hukkaa (engl. waste) tuottavaa toimintaa, eikä istu tämän päivän sovelluskehitykseen, jossa pyritään minimoimaan manuaalisen työn osuus.

12. Yhteiskäyttöiset järjestelmät ja viestintä. Kun yhteiskäyttöisiin järjestelmiin tehdään muutoksia, niistä pitää tiedottaa koko organisaatiolle.
13. Hyvien käytäntöjen ja ohjeiden jakaminen. Projekteissamme tapahtuu jatkuvaa Leanin periaatteiden mukaista toiminnan kehittämistä. Ongelma on siinä, että hyviksi todetut käytännöt eivät leviä. Kyseinen tieto jää liian usein vain yksittäisen projektiryhmän tietoon ja siten hyödyntämättä organisaatiotasolla. Tarvitaan keskitetty tietolähde, johon kirjattaisiin hyviä ohjelmistokehityksen käytäntöjä. Sama koskee ohjelmistotuotannon ohjeita. Keskitetty tietolähde voisi olla esim. Confluencen luotava yhteenvetosivu. Yhteenvetosivu voisi olla itsenäinen osansa Confluencessa tai osa Ohjelmistotuotannon käsikirjaa. Käyttäjät voisivat lisätä sivulle linkkejä projekteissa luotuihin ohjelmistotuotannon ohjeisiin ja projektien tuottamiin hyviin käytäntöihin. Kyseinen ratkaisu helpottaisi tietojen löytämistä. Tietoja osattaisiin tällöin paremmin hyödyntää kaikkialla Mitpassa.
14. Ympäristöt. Teknisen ympäristön perustaminen tulisi tehdä nykyistä helpommaksi ja nopeammaksi. Toimipaikastamme puuttuu integraatiotestiympäristö.
15. Käytettävyys. Tietojärjestelmäprojektin lopputuloksen kannalta yksi tärkeimmistä arviointikriteereistä on käyttäjätyytyväisyys. Osa käyttäjätyytyväisyydestä syntyy hyvästä käytettävyydestä. Se tarkoittaa, että käyttäjä kokee tietojärjestelmän helpokäyttöiseksi, jolloin järjestelmän käyttö on selkeää ja loogista. Yksi keino varmistaa käyttäjien ja asiakkaidemme tyytyväisyys toteuttamiemme järjestelmiä kohtaan on tehdä käytettävyydestä pakollinen tehtävä ohjelmistotuotantoprosessissamme. Tällä hetkellä käytettävyydestä tehdään vain satunnaisesti, vaikka sen hyödyt ovat selkeät. Mitpan henkilöstössä on käytettävyysosaamista (käyttöliittymäsuunnittelu, käytettävyyden arviointi ja käytettävyydestä) ja sitä tulisi hyödyntää nykyistä aktiivisemmin.
16. Henkilöstön palaute ja palautteeseen reagointi. Kahden tähän tutkimukseen sisältyneen, eri vuosina (2015 ja 2017) toteutetun kyselyn tuloksista voidaan päätellä, että joillakin laadunvarmistuksen osa-alueilla esimiehet eivät ole ryhtyneet näkyviin

toimenpiteisiin henkilöstön ilmoittamien ongelmien korjaamiseksi. Henkilöstön mukaan ongelmiin sisältyvät mm. puutteet määrittelyjen laadussa, puutteet katselmoitien suorittamisessa ja seurannassa sekä riittämättömät henkilöresurssit laadunvarmistuksessa. Ongelmien korjaamatta jättäminen heikentää työilmapiiriä ja henkilöstön luottamusta esimiestyöhön. Tilanteen korjaamiseksi esimiesten tulee esitellä julkisesti henkilöstölle toimenpiteet, joilla he aikovat ratkaista ongelmia. Ongelmien ratkaisumalleja tulee suunnitella yhdessä henkilöstön kanssa. Näitä malleja voidaan käsitellä esim. yksikköjen kuukausipalavereissa.

17. Testauskohteiden priorisointi on heikkoa erityisesti vesiputousmallisissa projekteissa. Priorisointi on työohje, joka määrittelee testauskohteiden tärkeysjärjestyksen (esimerkki: järjestelmän osat A, D ja K testattava jokaisen testiversion yhteydessä). Priorisointivastuu on yleisesti jätetty testajille, vaikka vastuu kuuluu järjestelmän tilaajalle/omistajalle (esim. Scrum-projektissa tuoteomistajalle). Tilanteen korjaaminen mm. nopeuttaisi testajan perehtymistä uuteen järjestelmään ja testauksen aloittamista. Nämä hyödyt korostuvat projekteissa, joissa testajilla ei ole aiempaa, kyseiseen järjestelmään liittyvää substanssietämystä.
18. Työn organisoinnin kehittäminen. Esimerkiksi testajan työkuorman ollessa liian suuri hänen on pakko tehdä työnsä nopeimmalla mahdollisella tavalla ja keskittyä testien suorittamiseen ja bugien eli sovellusvirheiden raportointiin. Tuollaisessa työpaineessa uusista työvälineistä ja niiden ominaisuuksista ei ole apua, koska ei ole aikaa välineiden opiskeluun. Ongelma ei siis ratkea työvälineillä, vaan työn organisointia muuttamalla. Tässä muutoksessa työtehtävien (projektityö, ylläpitotyö, viraston sisäiset työt ym.) priorisointi on avainasemassa. Se ei voi olla pelkästään työntekijän vastuulla, vaan se on yksi esimiesten perustehtävistä. Priorisointi Maanmittauslaitoksen sisällä ei kuitenkaan riitä, vaan sitä pitäisi tehdä säännöllisesti ja nykyistä suuremmassa mittakaavassa: virastojen välisenä yhteistyönä (katso kehittämissuositus 20).
19. CASE-työkalut ja ohjelmistohankinnat. Valtionhallinnossa virastot saavat tällä hetkellä hankkia ohjelmistotyövälineitä suhteellisen itsenäisesti. Tämä johtaa väistämättä hallinnonaloilla esim. päällekkäisiin lisenssikustannuksiin. Toisaalta koordinoinnin puute välineiden hankinnassa voi aiheuttaa välineiden yhteentoimimattomuutta. Esimerkiksi ohjelmistotuotannon työvälineiden hankintaa ja käyttöä pitäisi arvioida kokonaisuutena. Virastojen tulisi syventää yhteistyötään ohjelmistohankinnoissa. Virastojen tulisi kartoittaa ohjelmistotarpeitaan yhdessä ja tehdä ohjelmistoissa yhteishankintoja. Ministeriön tehtävänä olisi tarkastella hallinnonalansa ohjelmistohankintoja kokonaisuutena, arvioida hankintojen kokonaisyhteyksiä sekä

koordinoida hankintojen valmistelua. Tavoitteena olisi päällekkäisten hankintojen estäminen ja edistää ohjelmistojen yhteentoimivuutta. Työvälineiden yhdenmukaistaminen hallinnonalalla toisi kustannussäästöjä ja helpottaisi esim. ihmisten työskentelyä eri tietojärjestelmien kehittämisen parissa, koska työvälineet olisivat tuttuja. Esimerkiksi virastoilla tulisi olla yhteinen Confluence ja Jira. Tällöin tiedot olisivat keskitetyksi yhdessä välineessä ja helpommin virastojen henkilöstön saatavilla, mikä sujuvoittaisi työntekoa. Tietojen saatavuutta rajattaisiin käyttöoikeuksien avulla.

20. Tietojärjestelmien kehittäminen hallinnonalalla, kehittämiskohteiden priorisointi ja resursointi. Ministeriön tulisi olla tietojärjestelmien kehittämistä koordinoiva toimija hallinnonalalla. Ideaalitulanteessa hallinnonalan virastot kykenevät keskenään priorisoimaan tietojärjestelmien kehittämistä. Mikäli virastojen yhteistyö ei toimi, ministeriö viime kädessä päättäisi - hallinnonalan keskusorganisaationa - esim. projektien tärkeysjärjestyksestä. Koska tietojärjestelmät ovat nykypäivänä organisaatioiden ydintoiminto, sen riittävä resursointi on turvattava. IT-palvelutuotantomme on toiminut ja toimii jatkossakin rajallisten henkilöresurssien varassa. Henkilöstömme ylikuormittumisen estämiseksi ministeriön pitää kantaa vastuunsa toiminnan koordinoijana:

1. Osallistua aktiivisesti virastojen tietojärjestelmien kehittämistarpeiden kartoittamiseen ja priorisointiin.
2. Valvoa kehittämistarpeiden toteutusta.
3. Tarpeen mukaan tietojärjestelmien kehittämistä priorisoidaan uudelleen ja muuteen resurssien (aika, raha, ihmiset jne.) kohdentamista.

21. Organisaation tavoitteet ja johdon tuki. Johtajien ja lähiesimiesten pitää kertoa henkilöstölle selkeästi työyhteisön tavoitteet. Heidän pitää myös itse toimia esimerkkinä eli osoittaa henkilöstölleen - sanoillaan ja ennen kaikkea teoillaan - työskentelevänsä yhteisten tavoitteiden saavuttamiseksi. Sanoihin ja tekoihin sisältyy heidän muutostoimenpiteille osoittamansa tuki. Vaikka muutosten toteuttamisen (esim. uuden työmenetelmän käyttöönotto) operatiivinen vastuu yleensä kuuluu työntekijöille, kuten työnjohtajille (esim. projektipäälliköt), muutoksia ei voida toteuttaa työyhteisössä ilman organisaation johdon ja lähiesimiesten tukea.

22. Toiminnan jatkuva kehittäminen. Organisaation menestymisen perusedellytys on sen sitoutuminen toimintansa jatkuvaan kehittämiseen. Se ei toimi, ellei jatkuvaa kehittämistä juurruteta osaksi organisaation kulttuuria. Mitpan toiminta perustuu jatkossakin rajallisiin henkilöresursseihin. Palvelutuotannon turvaamiseksi työntekijöillä pitää olla mahdollisuus osaamisensa kehittämiseen ja opiskeluun - työajalla.

Omien kokemusteni pohjalta nykyinen tuki työntekijän korkeakouluopinnoille on heikko Mitpassa. Esimerkiksi palkallisen opintovapaan myöntämisessä tulee huomioida työntekijän elämäntilanne ja siihen liittyvät yksityiselämän velvoitteet (esim. pienten lasten hoitaminen), palvelusaika kyseisellä työnantajalla, työntekijän opintojen yhteys työnantajan toimialaan ja tehtäviin sekä työntekijän opinnoissaan työnantajalleen tuottama hyödyllinen tieto, kuten opintojaksojen harjoitustöiden ja opinnäytetyön sisältö. Kyseisiä tekijöitä ei huomioitu omalla kohdallani hakiessani palkallista opintovapaata tukemaan perheen, työn ja opiskelun yhteensovittamista. Hyvän, alaisen tarpeet huomioivan johtajuuden näkökulmasta palkallista opintovapaata tulisi käyttää julkisella sektorilla välineenä tukemaan työntekijän työkykyä ja työssä jaksamista opiskeluaikana. Siten palkallinen opintovapaa olisi myös keino vahvistaa työntekijän työviihtyvyyttä ja sitouttaa työntekijää työnantajaansa.

6 Pohdinta ja jatkotutkimusehdotukset

Tämä opinnäytetyö oli rajattu käsittelemään vain Mitpan Vallilan toimipaikan laadunvarmistuskäytäntöjä. Osassa kyselyiden vastauksista sekä vastausten analyysissä esitettiin hyviä, koko ohjelmistotuotantoa koskevia ehdotuksia. Aineisto sisältää siten kehitysideoita, jotka ovat yleishyödyllisiä koko Mitpassa, kuten panostaminen testausautomaatioon, määrittelyjen tarkkuuteen ja katselmointeihin. Siksi aineiston hyötyjä tulee tarkastella laajemmasta näkökulmasta kuin ainoastaan Vallilan toimipaikkaan rajattuina. Koska maisteriopintojeni keskeinen teema on ollut johtamistaidot, olen laadunvarmistuksen kehittämisen rinnalla tarkastellut myös johtamisen laatua sekä työnantajallani että hallinnonalallamme.

Teoriaosan lähdemateriaalissa käsiteltiin ristiriitaisesti laadunvarmistus-käsitettä. Osa lähde-eteoksista erotti toisistaan laadunvarmistuksen ja testauksen. Mielestäni kyseinen erotte- lu on keinotekoinen sekä vanhentunut. Omien kokemusteni pohjalta esimerkiksi katsel- moinnit ja testaus tukevat toisiaan. Siksi ohjelmistojen laadunvarmistusta tulee mielestäni - myös terminologian osalta - tarkastella kokonaisuutena. Siihen sisältyvillä tehtävillä on sama, yhteinen tavoite. Eli huolehtia siitä, että ohjelmiston käyttäjän ohjelmistolle asetta- mat odotukset täyttyisivät. Oma määritelmäni onkin: laatu on odotusten täyttämistä.

Kuten käyttämässäni lähdekirjallisuudessa todettiin, termien katselmointi, tarkastus ja läpi- käynti merkityksessä esiintyy kirjavuutta tietotekniikka-alalla. Toimipaikassani käytetään - ainakin toistaiseksi - vain katselmointi-termiä. Katselmointeja toteutetaan tuotokselle, esim. käyttöliittymäkuvaukselle ja tietojärjestelmän lähdekoodille, sekä projektin vaiheelle.

Laadunvarmistuksen uusien menetelmien käyttöönnotto on edennyt epätasaisesti. Tes- tauksen automatisointia on tehty jo usean vuoden ajan, mutta sen laajentuminen on ollut hidasta. Vielä hitaampaa on ollut katselmointien käyttöönnotto. Puutteet katselmointien to- teuttamisessa johtavat osaltaan siihen, että tekemättä jääneet, epäselvät ja puutteelliset määrittelyt vaikeuttavat, hidastavat tai estävät sujuvaa työntekoa ohjelmistotuotannossam- me. Laadunvarmistuksemme tiettyjen osa-alueiden rapautumiseen pitää reagoida. On syytä esimerkiksi pohtia, onko työyhteisössämme keskitytty viime vuosina liikaa ketterän sovelluskehitysmallin ja DevOps-mallin (ohjelmistojen kehitys- ja julkaisutoiminnan yhdis- täminen ketteryyden periaatteita noudattaen) edistämiseen ja tämä on toteutunut laadun- varmistuksen seurannan ja kehittämisen kustannuksella.

Aineistossa kuvattujen laadunvarmistuskäytäntöjemme heikkouksien, vahvuuksien ja ke- hittämisehdoiden pohjalta kirjoitin kehitysehdotukset. Esittelin ne - sekä työkalureideni että omani - luvussa 5.2 (Kehittämisehdotukset kohdeorganisaatiolle). Organisaation toimin- nan kehittäminen perustuu usean osa-alueen huomioimiseen, kuten huolella suunniteltui-

hin prosesseihin sekä työskentelyä tukeviin työmenetelmiin ja työvälineisiin. Inhimillisestä näkökulmasta organisaation toiminnan laatu riippuu:

1. Työntekijöiden motivaatiosta työtään kohtaan, ammattitaidosta, huolellisuudesta sekä ammattitilpeydestä eli halusta tehdä työnsä hyvin.
2. Työnantajan kiinnostuksesta ja kyvystä varmistaa työntekijöille edellytykset tehdä työnsä hyvin. Näihin edellytyksiin sisältyvät mm. riittävät henkilöresurssit, koulutusmahdollisuudet sekä esimiesten arvostus alaistensa työtä kohtaan ja esimiesten myönteinen, kannustava asenne alaisten haluun kehittää osaamistaan. Edellä luetellut tekijät vaikuttavat mm. henkilöstön työtehokkuuteen, työyhteisön ilmapiiriin sekä työntekijöiden työiihtyvyyteen ja työssä jaksamiseen. Ollessaan toimivia kyseiset tekijät vaikuttavat myönteisesti organisaation palvelukykyyn, palveluiden laatuun ja asiakastyytyväisyyteen.

Tutkimuksessani on käsitelty ohjelmistotuotantomme laadunvarmistuksen kehityskohteita ja vahvuuksia. Mielestäni suurimmat vahvuutemme ohjelmistotuotannossa ovat henkilöstön hyvä substanssitetämys, pitkä kokemus hallinnonalan tietojärjestelmien kehittämisestä ja työntekijöiden halu kehittää omaa osaamistaan sekä organisaation toimintaa. Työvälineistämme esimerkiksi jatkuva integraatio (CI) ja siihen integroidut työkalut, kuten testikattavuudet ja automaattinen lähdekoodin katselmointi, ovat lisänneet ohjelmointityön laadun läpinäkyvyyttä ja helpottaneet sovelluskehittäjien työn valvontaa testausvastaavan työssäni. Edellä mainitsemieni vahvuuksien säilymisestä pitää huolehtia myös jatkossa.

Maa- ja metsätalousministeriö teki marraskuussa 2016 päätöksen Elintarviketurvallisuusviraston (Evira) ja Maaseutuviraston (Mavi) yhdistämisestä. Yhdistämishankkeeseen sisältyy myös Mitpan asiakasvirastoille palveluja tuottavan yksikön eli Vallilan toimipaikan toimintojen siirto uuteen nk. suurvirastoon. Sen toiminta alkaa 1.1.2019. Päätös tarkoittaa toimipaikkani henkilöstölle toista suurta organisaatiomuutosta vain muutaman vuoden sisällä. Työyhteisössäni on pitkään yritetty sovittaa yhteen asiakasvirastojen tarpeita ja yksikkömme rajallisia henkilöresursseja. Palvelutasomme varmistaminen on viime vuosina vaikeutunut henkilöstön vaihtuvuuden vuoksi. Organisaatiomuutosten on lukuisissa tutkimuksissa (esim. PSYRES-tutkimushanke) todettu vaikuttavan kielteisesti mm. henkilöstön työhyvinvointiin ja työtehtäviin sitoutumiseen. Tuntematta yksityiskohtaisesti henkilöstömme vaihtuvuuden syitä on kuitenkin selvää, että toistuvat organisaatiomuutokset työyhteisössämme eivät edistä työiihtyvyyttä, eivätkä päivittäisistä velvoitteistamme suoriutumista. Ohjelmistojen laadunvarmistus on osa kyseisiä velvoitteita. Keräämäni aineiston perusteella henkilöstömme on tunnollista, ammattitaitoista sekä työhönsä ja toiminnan kehittämiseen sitoutunutta. Toisaalta aineistosta nousee esille mm. huoli henkilöstön ylikuormit-

tumisesta. Työntekijät ovat työnantajan tärkein pääoma. Sen säilyminen on varmistettava myös Eviran ja Mavin yhdistämishankkeen yhteydessä.

Käsittelin luvuissa 2.4 ja 2.5 muutosjohtamista. Se on ajankohtainen osaamisalue hallinnonalallamme Eviran ja Mavin yhdistämishankkeen johdosta. John Kotter painottaa Muutos vaatii johtajuutta-klassikkokirjassaan johtajuuden eli ihmisten johtamistaidon tärkeyttä nykyaikaisessa organisaatiossa. Omien kahden vuoden kokemusteni pohjalta Maanmittauslaitoksen esimiestyössä painottuvat asiajohtamisen taidot. Asiantuntijaorganisaation johtamisessa 2000-luvulla asetelman pitää olla päinvastainen eli johtamisessa pitää painottaa esimiehen henkilöjohtamisen taitoja. Moderni johtamismalli perustuu - esimiehen autoritäärisyyden sijaan - tunneälyyn, alaisen työn edellyksistä huolehtimiseen sekä valmentamiseen eli oppimiseen kannustamiseen ja osaamisen kehittämiseen. Maanmittauslaitoksessa tulisikin laatia vuosittain painopisteet niin henkilöstön kuin esimiesasemassa olevien henkilöiden osaamisen kehittämiseksi.

Kuinka tärkeänä seurannan kohteena Vallilan toimipaikan ja Maanmittauslaitoksen johto pitää henkilöstön työhyvinvointia? Opinnäytetyöni tuloksista välittyy työntekijöiden tyytymättömyyttä, osittain myös turhautumista. Tulkitsen niiden kuitenkin liittyvän työn esteisiin, eikä edellä mainittuun, käynnissä olevaan muutoshankkeeseen. Henkilöstön reaktiot kertovat, että toimipaikkamme toiminnan tietyillä osa-alueilla ei ole tapahtunut henkilöstön kaipaamia muutoksia. Siksi Eviran ja Mavin yhdistämishanke luo lisähaasteen toimipaikkamme työhyvinvoinnille: työn esteitä pitäisi raivata kiireellisesti, mutta samanaikaisesti pitäisi valmentaa ja tukea henkilöstöä keskellä organisaatiomuutosta. Muutostilanteissa korostuvat esimiesten henkilöjohtamisen taidot. Työntekijöiden tarpeiden yksilöllinen huomiointi ja sitä tukevat esimiesten toimenpiteet ovat mielestäni tällä hetkellä riittämättömiä. Tässä tilanteessa viraston johdon tulisi panostaa työtyytyväisyyden aktiivisempaan seurantaan. Koska organisaatiomme on muutostilanteessa, vuosittain toteutettavaa valtionhallinnon työtyytyväisyyskyselyä tulisi täydentää vähintään yhdellä, keväällä järjestettävällä henkilöstökyselyllä. Se kohdistettaisiin Eviran, Mavin ja Mitpan Vallilan toimipaikan henkilöstölle. Työhyvinvointia mittaavissa kyselyissä pitää numeeristen vastausten lisäksi tarkastaa ja huomioida avoimet vastaukset: ne kertovat henkilöstön työhyvinvoinnista paljon yksityiskohtaisemmin kuin numerot. Lisäksi muutosten kohteena olevalle henkilöstölle pitäisi avata yhteinen keskustelufoorumi (esim. extranettiin), jossa työntekijät voisivat esittää kysymyksiään, näkemyksiään ja ideoitaan, myös anonyymisti. Tärkeintä yhdistämishankkeessa on henkilöstön osallistaminen. Kuten John Kotter muistuttaa edellä mainitussa teoksessaan, organisaatiomuutos ei ole johdon, vaan koko työyhteisön projekti. Muutoksia ei ole mahdollista toteuttaa onnistuneesti, mikäli henkilöstö ei pääse osallistumaan muutosten suunnitteluun ja toteutukseen.

Maisteriopinnot merkitsivät minulle tasapainoilua perheen, työn ja opiskeluelvoitteiden välillä. Toteutin opinnäytetyöni, aineopintojen tavoin, kokonaisuudessaan vapaa-ajallani. Tilanteeni oli poikkeus omassa opiskelijaryhmässäni ja lisäsi merkittävästi kuormitustani kuluneen kahden vuoden aikana. Vaikka minulla ei ollut mahdollisuutta tehdä opinnäytetyötäni työajalla, en supistanut tutkimukseni kohdetta. Halusin pitää kiinni tutkimusaiheestani, koska vastaavaa, ohjelmistotuotannon laadunvarmistukseen keskittynyttä tutkimusta ei oltu toteutettu työyhteisössäni kuluneen yhdeksän palvelusvuoteni aikana. Koin tärkeäksi sen, että henkilöstömme ääni eli mielipiteet ja ideat laadunvarmistuskäytäntöjemme kehittämiseksi tuodaan esille. Aineistoa kertyi runsaasti, mutta ilahduttavinta oli tietenkin sen sisältö eli työntekijöiden pohdinta ohjelmistotuotantomme laadunvarmistuksen kehittamisestä. Tutkimukseni yksi tavoite oli kerätä kattavasti vastauksia eri tehtävärooleissa työskenteleviltä asiantuntijoiltamme. Tuokin tavoite toteutui.

Johtamisen näkökulmasta mielenkiintoinen jatkotutkimusaihe olisi vuoden 2015 organisaatiomuutoksen (Maanmittauslaitoksen, Tiken ja Geodeettisen laitoksen yhdistäminen) sekä tulevan Evira-Mavi-Mitpa-fuusion vaikutukset henkilöstömme työhyvinvointiin. Toinen tutkimuskohde koskisi tässä opinnäytetyössä esiin nousseita laadunvarmistuskäytäntöjen puutteita. Kyseisessä jatkotutkimuksessa selvittäisiin, millaisilla toimenpiteillä laadunvarmistuskäytäntöjemme puutteita on yritetty korjata ja miten niissä on onnistuttu. Lisäksi tulisi esittää ratkaisumalleja mahdollisten jäljellä olevien puutteiden/ongelmien korjaamiseksi. Vaikka tutkimusideani laadunvarmistuskäsikirjasta ei toteutunut, pidän aihetta edelleen tärkeänä. Laadunvarmistuksen toimintamallien yhdenmukaistamisen tulisi olla osa Evira-Mavi-Mitpa-yhdistämishanketta. Siksi laadunvarmistuskäsikirjan voisi ottaa uudelleen esille tutkimusaiheena yhdistämishankkeen edetessä.

Lähteet

Auer, A., Auer L., Heinäsmäki M., Hölttä J., Kalliala E., Laanti M., Laine K., Lekman L., Miinalainen P., Naski H., Piiparinen T., Puhakka H., Pyhäjärvi M., Pääkkönen T., Räisänen S., Sora H., Taipale M., Talvio J., Tanninen A., Toikkanen T., Toivola T., Toro K., Valsta A., Väyrynen V. & von Weissenberg M. 2013. Ketterää kehitystä. Finn Lectura. Helsinki.

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

Haikala, I. & Märijärvi, J. 2004. Ohjelmistotuotanto. Talentum. Helsinki.

Helsingin yliopisto 2009. Arkkitehtuuri pienessä mittakaavassa. Luettavissa: <https://www.cs.helsinki.fi/u/laine/arkki/s09/pdf/oa-04-komponentit-rajapinnat.pdf>. Luettu: 8.4.2015.

Helsingin yliopisto 2002. Ohjelmointikielet. Luettavissa: <http://www.ling.helsinki.fi/kit/kurssit/verkkokurssit/korpuskurssi/ohjelmointikieli.shtml>. Luettu: 11.4.2015.

Ilmarinen. Onnistunut muutos - tukea onnistuneen muutoksen suunnitteluun ja läpivientiin. Luettavissa: <https://www.ilmarinen.fi/siteassets/liitepankki/tyohyvinvointi/opas-onnistunut-muutos.pdf>. Luettu: 28.3.2017.

Juuti, P. & Virtanen, P. 2009. Organisaatiomuutos. Otava. Helsinki.

Jyväskylän yliopisto 2004. Katselmoinnit. Luettavissa: http://www.mit.jyu.fi/opetus/kurssit/jot/2005/kalvot/katselmoinnit_05.pdf. Luettu: 15.4.2015.

Jyväskylän yliopisto 2003. Käytettävyys. Luettavissa: <http://appro.mit.jyu.fi/soveproj/kaytettavyys/#TOC3>. Luettu: 28.4.2015.

Kasurinen, J. 2013. Ohjelmistotestauksen käsikirja. Docendo. Jyväskylä.

Kauppalehti 2014. Esimiehet muutosjohtamisen etulinjassa. Luettavissa: <http://blog.kauppalehti.fi/muutoksen-ammattilaiset/esimiehet-muutosjohtamisen-etulinjassa>. Luettu: 9.4.2017.

Keva. Esimiehenä muutoksessa - esimiehen opas muutokseen. Luettavissa: http://esimiehenopasmuutokseen.fi/esimiehena_muutoksessa. Luettu: 9.4.2017.

Kotter, J. 1996. Muutos vaatii johtajuutta. Rastor. Helsinki.

Kotter, J. & Rathgeber, H. 2014. Jäävuoremme sulaa: muutos ja menestyminen kaikissa olosuhteissa. Talentum Media. Helsinki.

Lehtimäki, T. 2006. Ohjelmistoprojektit käytännössä. Readme.fi. Helsinki.

Lescelius, S. 2010. Suorituskykytestauspalvelun kehittäminen. Luettavissa: http://www.theseus.fi/bitstream/handle/10024/15733/Lescelius_Santtu_Suorituskykytestauspalvelun_Kehittaminen.pdf?sequence=1. Luettu: 7.4.2015.

Mattila, P. 2008. Otollinen tilaisuus: miten tarttua muutokseen. Talentum. Helsinki.

Maanmittauslaitos 2012. Maanmittauslaitos 200 vuotta-verkkonäyttely. Luettavissa: <http://www.maanmittauslaitos.fi/nayttelyt/10631/teos/10615>. Luettu: 15.10.2016.

Maanmittauslaitos 2015. Maanmittauslaitos 2000-2015. Luettavissa: <http://www.maanmittauslaitos.fi/yhteen2015>. Luettu: 15.10.2016.

Maanmittauslaitos 2016. Toiminta ja tehtävät. Luettavissa: <http://www.maanmittauslaitos.fi/toiminta/organisaatio>. Luettu: 15.10.2016.

McConnell, S. 2002. Ohjelmistotuotannon hallinta. Edita Publishing. Helsinki.

Murch, R. 2002. IT-projektinhallinta. Edita Publishing. Helsinki.

Pirinen, H. 2014. Esimies muutoksen johtajana. Talentum. Helsinki.

Pohjolainen, P. 2003. Ohjelmiston testauksen automatisointi. Luettavissa: http://www.cs.uku.fi/tutkimus/Teho/PenttiPohjolainen_Gradu.pdf. Luettu: 28.4.2015.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Docendo. Jyväskylä.

Ponteva, K. 2012. Muutoksessa. Sanoma Pro. Helsinki.

Tampereen teknillinen yliopisto 2007. Luento 6: käytettävyyden hyötyjä ja arviointimenetelmiä. Luettavissa: <http://www.cs.tut.fi/~kaper/syksy07/luennot/S07L6.pdf>. Luettu: 28.4.2015.

Tampereen yliopisto 2000. Käytettävyys ja verkko-oppiminen. Luettavissa: http://www.sis.uta.fi/ipopp/ipopp2000/AhtinenKokkonen/kaytettavaus_2.html. Luettu: 28.4.2015.

Teknillinen korkeakoulu 2001. Tietokone ja ohjelmointikieli. Luettavissa: <http://www.cs.hut.fi/~lma/kevytOLO/s2001/Materiaalia/Luku-3.html>. Luettu: 11.4.2015.

Wikipedia 2015a. Ohjelmistotuotanto. Luettavissa: <http://fi.wikipedia.org/wiki/Ohjelmistotuotanto>. Luettu: 4.4.2015.

Wikipedia 2015b. Functional requirement. Luettavissa: http://en.wikipedia.org/wiki/Functional_requirement. Luettu: 7.4.2015.

Wikipedia 2015c. Non-functional requirement. Luettavissa: http://en.wikipedia.org/wiki/Non-functional_requirement. Luettu: 7.4.2015.

Wikipedia 2013a. Konekieli. Luettavissa: <http://fi.wikipedia.org/wiki/Konekieli>. Luettu: 11.4.2015.

Wikipedia 2015d. Lähdekoodi. Luettavissa: <http://fi.wikipedia.org/wiki/L%C3%A4hdekoodi>. Luettu: 11.4.2015.

Wikipedia 2015e. Ohjelmointikielen kääntäjä. Luettavissa: http://fi.wikipedia.org/wiki/Ohjelmointikielen_k%C3%A4%C3%A4nt%C3%A4j%C3%A4. Luettu: 11.4.2015.

Wikipedia 2015f. Software quality assurance. Luettavissa: http://en.wikipedia.org/wiki/Software_quality_assurance. Luettu: 10.4.2015.

Wikipedia 2015g. Software review. Luettavissa: http://en.wikipedia.org/wiki/Software_review. Luettu: 14.4.2015.

Wikipedia 2015h. Audit. Luettavissa: <http://en.wikipedia.org/wiki/Audit>. Luettu: 10.4.2015.

Wikipedia 2015i. Winston W. Royce. Luettavissa: http://en.wikipedia.org/wiki/Winston_W._Royce. Luettu: 16.4.2015.

Wikipedia 2013b. Prototyyppi. Luettavissa: <http://fi.wikipedia.org/wiki/Prototyyppi>. Luettu: 16.4.2015.

Wikipedia 2015j. Extreme programming. Luettavissa:

http://en.wikipedia.org/wiki/Extreme_programming. Luettu: 18.4.2015.

Wikipedia 2015k. Pair programming. Luettavissa: http://en.wikipedia.org/wiki/Pair_programming. Luettu: 18.4.2015.

Wikipedia 2015l. Lean. Luettavissa: <http://fi.wikipedia.org/wiki/Lean>. Luettu: 18.4.2015.

Wikipedia 2015m. Toyota Production System. Luettavissa:

http://en.wikipedia.org/wiki/Toyota_Production_System. Luettu: 18.4.2015.

Wikipedia 2015n. Kanban. Luettavissa: <http://fi.wikipedia.org/wiki/Kanban>. Luettu: 19.4.2015.

Wikipedia 2015o. Computer-aided software engineering. Luettavissa: http://en.wikipedia.org/wiki/Computer-aided_software_engineering. Luettu: 14.4.2015.

Wikipedia 2017. John Kotter. Luettavissa: https://en.wikipedia.org/wiki/John_Kotter. Luettu: 29.3.2017.

Wikipedia 2016a. Maanmittauslaitos. Luettavissa: <http://fi.wikipedia.org/wiki/Maanmittauslaitos>. Luettu: 15.10.2016.

Wikipedia 2008. Virasto. Luettavissa: <http://fi.wikipedia.org/wiki/Virasto>. Luettu 12.4.2015.

Wikipedia 2016b. Venäjän keisarikunta. Luettavissa:

https://fi.wikipedia.org/wiki/Ven%C3%A4j%C3%A4n_keisarikunta. Luettu: 15.10.2016.

Liitteet

Liite 1. Kyselylomakkeet

Kysely 1, kevät 2015.

Terve,

Aloitin viime tammikuussa opiskelun tietojärjestelmäosaamisen koulutusohjelmassa Haa-ga-Helia-ammattikorkeakoulussa. Osana Soveltava tutkimus ja kehittäminen-opintojaksoa toteutan laadunvarmistusta käsittelevän opinnäytetyön. Siihen sisältyy oheinen Sovelluskehitys-vastuualueelle osoitettu kysely, johon toivon mahdollisimman monen vastuualueemme työntekijän vastaavan.

Kyselyn tarkoituksena on selvittää nykyisten laadunvarmistuskäytäntöjemme vahvuudet ja heikkoudet, käytäntöjemme kattavuus sekä kerätä ideoita käytäntöjemme kehittämiseksi. Siten jokaisella vastauksella on suuri merkitys.

A) Tiivistelmä laadunvarmistuksesta.

Tässä kyselyssä ohjelmistotuotannon laadunvarmistus on jaettu kolmeen osa-alueeseen ja niiden alakäsitteisiin.

1) Testaus:

- yksikkötestaus
- integraatiotestaus
- järjestelmätestaus
- rajapintatestaus
- suorituskykytestaus
- hyväksymistestaus
- testausautomaatio.

2) Käytettävyyden varmistaminen:

- käytettävyyden arviointi (nk. heuristinen arviointi)
- käytettävyydestestaus.

3) Katselmoinnit:

- tuotoksen katselmointi
- koodikatselmointi (manuaalinen ja automatisoitu).

B) Kysymykset laadunvarmistuksesta.

1) Testaus.

a) Mitä ongelmia tai puutteita liittyy testaukseen?

b) Mitkä ovat vahvuutemme testauksessa?

2) Käytettävyyden varmistaminen.

a) Mitä ongelmia tai puutteita liittyy käytettävyyden varmistamiseen?

b) Mitkä ovat vahvuutemme käytettävyyden varmistamisessa?

3) Katselmoinnit.

a) Mitä ongelmia tai puutteita liittyy katselmointeihin?

b) Mitkä ovat vahvuutemme katselmoinneissa?

4) Mikä laadunvarmistuksen osa-alue mielestäsi puuttuu ja tulisi liittää osaksi ohjelmistotuotantoamme?

5) Miten kehittäisit nykyisiä laadunvarmistuskäytäntöjämme (esim. työmenetelmät, ohjeet tai työvälineet)?

Vastaukset pyydän lähettämään 14.4.2015 mennessä.


Vastauksista kiittäen,
Pekka

Kysely 2, talvi 2017.

Mozilla Firefox

https://www.webropol-surveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1 Tulosta kaikki:



Tutkimus ohjelmistotuotannon laadunvarmistuskäytäntöjen kehittämisestä - Vallilan toimipaikka

Opiskelen ylempää tradenomin tutkintoa tietojärjestelmäosaamisen koulutusohjelmassa, Haaga-Helia-ammattikorkeakoulussa. Opinnäytetyössäni tutkin laadunvarmistuskäytäntöjä Maanmittauslaitoksen Vallilan toimipaikassa. Tutkimukseeni sisältyy tämä Tekno- ja Assi-tulosyksiköille osoitettu kysely.

Kyselyn tarkoituksena on selvittää nykyisten laadunvarmistuskäytäntöjemme heikkoudet ja vahvuudet, käytäntöjemme kattavuus sekä kerätä ideoita käytäntöjemme kehittämiseksi. Siten jokaisella vastauksella on suuri merkitys. Kyselyyn voi vastata 5.2.2017 saakka.




Huom:
Tämä tutkimus käsittelee ohjelmistotuotannon laadunvarmistuksen käytäntöjä vain Maanmittauslaitoksen asiakasvirastoille tuotettavassa IT-palvelutuotannossa Vallilan toimipaikassa, Helsingissä. Muut Maanmittauslaitoksen laadunvarmistuskäytännöt on suljettu tutkimuksen ulkopuolelle.

Tässä kyselyssä ohjelmistotuotannon laadunvarmistus on jaettu kolmeen osa-alueeseen ja niiden alakäsitteisiin.

- 1) Testaus, esimerkkejä testaustehtävistä:
 - yksikkötestaus
 - integraatiotestaus
 - järjestelmätestaus
 - rajapintatestaus
 - suorituskykytestaus
 - tietoturvatestaus
 - hyväksymistestaus
 - testausautomaatio.
- 2) Käytettävyyden varmistaminen:
 - käytettävyydsiantuntijan suorittama käytettävyyden arviointi (nk. heuristinen arviointi)
 - käytettävyydestestaus.
- 3) Katselmoinnit:
 - dokumentin katselmointi
 - tietokannan katselmointi
 - koodikatselmointi (manuaalinen ja automatisoitu)
 - työvaiheen katselmointi (esim. projektin määrittelyvaiheen katselmointi).

Mozilla Firefox

https://www.webpolsurveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1    Tulosta kaikki: (nk. heuristinen arviointi)

3) Katselmoinnit:
- dokumentin katselmointi
- tietokannan katselmointi
- koodikatselmointi (manuaalinen ja automatisoitu)
- työvaiheen katselmointi (esim. projektin määrittelyvaiheen katselmointi).

Lisätietoja tutkimuksesta:
pekka.saarinen@maanmittauslaitos.fi

Kyselyn esitiedot

1. Tulosityksikkösi *

Asiakasratkaisut
 Teknologiaratkaisut




2. Tehtäväroolisi *
(tehtävärooli, jossa pääasiallisesti työskentelet)

ETL-kehittäjä
 IT-palveluvastaava
 Järjestelmäasiantuntija
 Projektipäällikkö
 Raporttikehittäjä
 Sovellus- tai tietojärjestelmäarkkitehti
 Tekninen koordinaattori
 Teknologiakoordinaattori
 Testaaja
 Testausvastaava
 Tietojärjestelmävastaava
 Tietokantavastaava

Kyselyn kysymykset

Mozilla Firefox

https://www.webpolsurveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1    Tulosta kaikki:


- Järjestelmäasiantuntija
- Projektipäällikkö
- Raporttikehittäjä
- Sovellus- tai tietojärjestelmäarkkitehti
- Tekninen koordinaattori
- Teknologiakoordinaattori
- Testaaja
- Testausvastaava
- Tietojärjestelmävastaava
- Tietokantavastaava

Kyselyn kysymykset

Testaus:




3. Mitä ongelmia tai puutteita liittyy testaukseen?

4. Mitkä ovat vahvuutemme testauksessa?



Mozilla Firefox

https://www.webpolsurveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1    Tulosta kaikki:

3. Mitä ongelmia tai puutteita liittyy testaukseen?

4. Mitkä ovat vahvuutemme testauksessa?




Käytettävyyden varmistaminen:

5. Mitä ongelmia tai puutteita liittyy käytettävyyden varmistamiseen?

6. Mitkä ovat vahvuutemme käytettävyyden varmistamisessa?

Mozilla Firefox

https://www.webpolsurveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1    Tulosta kaikki:

5. Mitä ongelmia tai puutteita liittyy käytettävyyden varmistamiseen?

6. Mitkä ovat vahvuutemme käytettävyyden varmistamisessa?




Katselmoinnit:

7. Mitä ongelmia tai puutteita liittyy katselmointeihin?

8. Mitkä ovat vahvuutemme katselmoinneissa?

Mozilla Firefox

https://www.webpolsurveys.com/Preview/PreviewQuestions.aspx?nocache=9473

Sivu: 1 / 1    Tulosta kaikki:

8. Mitkä ovat vahvuutemme katselmoinneissa?

Laadunvarmistuksen kattavuus:

9. Mikä laadunvarmistuksen osa-alue mielestäsi puuttuu ja tulisi liittää osaksi ohjelmistotuotantoamme?

Laadunvarmistuksen kehitysideat:

10. Miten kehittäisit nykyisiä laadunvarmistuskäytäntöjämme (esim. työmenetelmät, ohjeet tai työvälineet)?

Keskeytä

Lähetä

Liite 2. Kyselyn 1 vastaukset (kevät 2015)

Kohdeorganisaatio on määritellyt liitteen sisällön luottamukselliseksi.

Liite 3. Kyselyn 2 vastaukset (talvi 2017)

Kohdeorganisaatio on määritellyt liitteen sisällön luottamukselliseksi.