



TESTAUSTIIMIN OHJAAMINEN JA OSAAMISEN JOHTAMINEN

Terhi Stenroth

Kehittämistehtävä
Toukokuu 2011
Teknologiaosaamisen johtaminen
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

SISÄLTÖ

TIIVISTELMÄ

ABSTRACT

1	JOHDANTO.....	5
1.1	Työn taustat.....	5
1.2	Työn aihe ja tavoitteet	10
1.3	Tutkimuksen rakenne ja työn rajaus	12
1.4	Käsitteet ja käytettävät termit	13
2	OSAAMISEN JA OHJAAMISEN KEHITTÄMISMENETELMÄT	14
2.1	Johtaminen perustehtävän ja muutoksen kautta	14
2.2	Osaamisen johtamisprosessi.....	28
2.3	Ohjaaminen testauksen näkökulmasta.....	39
2.4	Kehys- ja kypsyyssmallit	54
2.5	Kanban	72
3	SOVELTUVUUS TESTAUKSEN OHJAAMISEKSI.....	78
3.1	Testaustiimin ohjaaminen.....	79
3.2	Testauksen osaamisen johtaminen	95
3.3	Testauksen kehittäminen.....	101
4	POHDINTA.....	104
	LÄHTEET.....	106
	LIITTEET.....	108

TIIVISTELMÄ

Stenroth, Terhi. 2011. Testaustiimin ohjaaminen ja osaamisen johtaminen. Kehittämistehtävä, Tampereen Ammattikorkeakoulu, Teknologiaosaamisen johtaminen. Tampere. 123 sivua. Liitteet 1 kpl.

Teen kehittämistehtäväni teknologiateollisuuden yritykselle. Tehtävän tavoitteena on arvioida osaamista, jota tarvitaan testaustyön suorittamiseksi testaustiimissä. Tarkoituksena on löytää keinot kehittää testaustiimin toimintaa ja testausprosessia siten, että työskentely tukisi tuotteiden laadunvarmistusta. Lisäksi pyritään tarkastelemaan päivittäisen työn ohjaamista ja testaajien osaamisen kehittämistä. Teoriaosuus aloitetaan käsittelemällä johtamista perustehtävän ja muutoksen kautta, sekä osaamisen johtamisprosessia. Testauksen näkökulmasta tarkastellaan testausstandardeja ja testauksen kehysmalleja, kuten TMMi ja TPI. Lisäksi arvioidaan Kanban-menetelmän toimivuutta testaustiimin päivittäisessä työskentelyssä.

Kehittämistehtävä osoittaa, että toimiva työskentely pohjautuu avoimeen ja aktiiviseen vuorovaikutukseen testaustiimin ja projektiryhmien kesken. Työskentely edellyttää testauksen tarkoituksen ymmärtämistä ja periaatteiden tietämistä. Onnistunut testaustiimin ohjaaminen puolestaan edellyttää testaustiimin osaamiskartan määrittelyä ja sen pohjalta osaamistarpeen arviointia. Testaustiimin osaamisen kehittämisessä tulee huomioida yksilön tarpeiden lisäksi tiimin tarpeet. Testausprosessin kehittämisen osalta voidaan käyttää hyödyksi kehysmalleja, mutta kehittäminen tulisi suorittaa hallitusti ja muutosten yhteydessä huomioida vaikutukset tiimin lisäksi yksittäisen työntekijän kykyyn omaksua muutos.

Asiasanat: testaus, osaamisen johtaminen, testauksen ohjaaminen

ABSTRACT

Stenroth, Terhi. 2011. Leading a testing team and managing their competence. Master's thesis, Tampere University of Applied Sciences, Degree Program in Strategic Leadership of Technology based business. Tampere. 123 pages. Appendices 1.

This Master's thesis is done for a company working in the technology branch. The purpose of this thesis is to find a way to estimate competence needed for testing operations. Also a way to improve operation of testing team and testing process is examined. Teams work should support the quality assurance. Leading a daily work and competence management of testers are observed too. The theory part describes leading via basic task, change process and competence process. Testing standards and testing models like TMMi and TPI are examined. Using Kanban tool in daily work is also estimated.

This thesis shows that efficient working requires open and active communication between a testing team and project groups. Working requires that the meaning of testing is known. Also principles of testing must be understood. A successful leading of a testing team needs a proper definition of a competence map. Therefore the competence need of the testing team must be estimated. Competence management of a testing team notices both the needs of an individual and the team. Testing process management can also use testing models. The development should be controlled even if the models are used. All changes must be noticed both in team's and individual's perspective. They both must have ability to accept changes.

Keywords: testing, competence management, test management

1 JOHDANTO

Testaustiimin johtaminen ja testaustyön ohjaaminen on haasteellista. Erityisesti sellainen ympäristö, jossa tiimillä on tehtävänä suoriutua useiden tuotteiden testaustyöstä samanaikaisesti ja testaajia on vain muutama, on työn organisointi vaativaa. Testaustiimi-ajattelun ollessa uusi eikä vertaistiimejä ei ole käytettävissä korostuu haasteellisuus. Yhteistyön ja osaamisen jakamisen kannalta haaste kasvaa, kun tiimiin kuuluvat testaajat ovat usean vuoden testauskokemuksen myötä tottuneet toimimaan pääosin itsenäisinä testaajina tai suoriutumaan vain tietyn osa-alueen testaustyöstä. Myös se, että testaajilla on eri henkilöesimiehet ja esimiehet työskentelevät tiimin ulkopuolella, asettavat omat vaatimukset tiimin osaamisen kehittämiseksi.

Kehittämistehtävällä halutaan etsiä vastausta testaustiimin ohjaamiseen ja osaamisen johtamiseen. Tiimin testausosaamisen kautta on tarpeen saada toimeksiantajan ja toimittajan näkökulmasta kaikkien eri projektien testausoiminnat suoritettua kattavasti. Työ on tehtävä käytettävissä olevilla resursseilla ja riittävän ajoissa ennen toimitusta. Samanaikaisesti on saatava täytettyä myös yksittäisen testaajan osaamisen kehittäminen niin yksilön itsensä kuin myös koko tiimin tarpeiden näkökulmasta.

1.1 Työn taustat

Tämä kehittämistehtävä laaditaan teknologiateollisuuden yritykselle, joka tarjoaa kotimaisille ja kansainvälisille asiakkailleen ratkaisuja ja palveluja verkkokeskeisiin johtamis-, tietoliikenne- ja koulutusjärjestelmiin sekä sähköisen liiketoiminnan verkottumiseen ja vahvaan tietoturvaan. Kehittämistehtävän kohteena on yksi yrityksen monista tiimeistä. Työntekijät toimivat yhdessä tai useassa projektiryhmässä. Lisäksi tiimiin kuuluu testaustiimi, jonka tehtävänä on suorittaa järjestelmätestausta. Yksikkö- ja integrointitestausta ovat projektiryhmien vastuulla.

Vuoden 2010 keväällä laadittiin opinnäytetyö aiheesta ”Testaustoimintojen uudistaminen ohjelmistokehitysprosessissa”. Työn tavoitteena oli kuvata testausprosessi ja arvioida kehityskohteet. Niiden perusteella päivitettiin testaustiimin käyttämä testausprosessi ja työskentelytavat eri projekteissa. Tutkimuksen tulokset olivat lupaavia. Uusi testausprosessi saatiin määriteltyä ja koekäytettyä tutkimuksen yhteydessä. Ennen käyttöönottoa prosessikuvausta viilattiin tutkimustulosten perusteella. Testausprosessi otettiin käyttöön testaustiimissä, uusi prosessikuvaus esiteltiin ja prosessi perehdytettiin testaajille. Työn julkaisun jälkeen on yrityksessä suoritettu laajamittainen toimintajärjestelmän päivitys ja jalkauttaminen. Testausprosessin yhtenevyys toimintajärjestelmän prosessikuvausvaiheeseen on varmistettu. Käyttöönoton jälkeen prosessimallia on tarkennettu käytännön työohjeiden muodossa. Ohjeistuksen avulla on tavoiteltu testaajien kykyä suoriutua työstään itsenäisesti ja tarvittaessa löytää tukea työn suorittamiseksi toisilta testaajilta ja dokumentoiduista ohjeista.

Testaustiimi osallistuu projektiryhmien työhön suorittamalla projektista riippuen staattista testausta jo projektin määrittelyvaiheesta asti. Staattinen testaus koostuu tuotteeseen toteutettavien ominaisuuksien määrittelyjen kommentoinnista ja testaustyössä huomioitavien tuotevaatimusten hyväksyntäkriteerien laadinnasta, sekä testaussuunnitelmien laadinnasta. Vaiheen tarkoituksena on testata tuleeko kaikki tarpeellinen huomioitua ominaisuuksien toteutuksessa. Näin saadaan vikojen määrä minimoitua suunnittelu- ja toteutusvaiheessa. Määrittelyvaihe nivoutuu tiiviisti kiinni suunnitteluvaiheeseen, jossa toteutus suunnitelmille jatketaan staattista testausta käytettävyydestä testauksen ja testitapauksien laadinnan muodossa. Tarkoituksena on noudattaa testauksen etupainotteisuutta. Toteutusvaiheen yhteydessä suoritetaan yksikkö- ja integrointitestaus. Testaustiimi osaltaan suorittaa dynaamisen järjestelmätestauksen testitapauksia vasten ja testaajan ammattitaitoon pohjautuvan tutkivan ja valmistelemättömän testauksen muodossa.

Ohjelmistokehitystyö on iteratiivista ja inkrementaalista, joten tuotetta kehitetään ominaisuuksittain. Joitakin laajempia ominaisuuksia hajotetaan pienempiin osiin

siten, että ominaisuutta kehitetään kokonaisuus kerrallaan. Käytännössä työskentelyn ja testauksen ohjaaminen prosessin mukaisesti monimutkaistuu testauksen ohjaamisen näkökulmasta, kun ajankäyttöä ei voi suunnitella tarkkaan vaiheittain. On osattava huomioida kussakin iteraatiossa toteutettavat ominaisuudet ja niiden sijoittuminen iteraatioon. Lisäksi on huomioitava, että samalla tavalla eteneviä projekteja on useita. Tosin iteraatioiden pituudet ja sisällön laajuudet poikkeavat toisistaan. Myös iteraatioiden suunnittelussa hyödynnettävä tehtävien keston arviointi on täysin riippuvainen tarpeiden ja vaatimusten tunnistamisesta ennalta ja kehittäjien työmääräarvioiden laadintaosaamisesta. On myös osattava varautua yllättäviin muutoksiin tai tapahtumiin. Näitä ovat muun muassa työntekijöiden mahdolliset poissaolot, uudet projektit tai lisäominaisuustarpeet käynnissä olevien projektien yhteydessä. Tehtävien arviointi ja sijoittaminen iteraatioon on kriittinen osa työskentelyä. Arvioinnin paikkaansa pitävyys on ehdoton edellytys testaustiimin työskentelyn ennakkoon suunnittelulle.

Ei siis riitä, että testausprosessi on saatu päivitettyä ja jalkautettua testaustiimiin. Pitää saada myös tukitoiminnot toimimaan yhdessä. Projektiryhmien tulisi olla kykeneviä huomioimaan muiden projektien vaikutus testaustiimin käytettävissä oleviin resursseihin ennen kuin projektin tehtäviä aikataulutetaan iteraatioille. Lisäksi heidän on saatava riittävästi tietoa testaustiimin kyvystä suoriutua sille asetetuista vaatimuksista niin toteutettavien ominaisuuksien osalta. Esimerkiksi testaustapoja valittaessa ja testattavuuskykyä arvioitaessa ja aikatauluja ja toteutusjärjestystä määriteltäessä on yhteistyön merkitys tärkeää. Ongelmia aiheuttavat myös projektiryhmien erilaiset prosessit, jotka eivät aina noudata yleisesti määriteltyä prosessia. Ryhmien vuorovaikutuksen onnistumisen edellytyksenä on kunkin ryhmän kyky noudattaa yhteisesti hyväksyty havaittuja käytänteitä. Lisäksi on noudatettava yrityksen toimintajärjestelmässä kuvattua prosessia ja työskentelyohjeita.

Työn onnistuminen vaatii avointa ja aktiivista vuorovaikutusta molempien ryhmien suunnalta. Kehitystyö vaatii vielä yhteistyön ja työskentelytapojen juurruttamista osaksi testaustyössä toimittaessa rajapinnassa kehittäjien, projektipääl-

liköiden ja muiden sidosryhmien kanssa. Tehokas ja kehittyvä tiimi vaatii testausprosessin ympärille selkeät työskentelyohjeet ja niiden noudattamisen. Näitä on jo laadittu ja prosessin tulkinta on yksiselitteinen työskentelyohjeista noudattaen. Tiimissä tarvitaan vielä työkalut arvioida, kuinka hyvin testaustyö kattaa kehitettävän tuotteen asettamat tarpeet erilaisille testaustavoille. Lisäksi on arvioitava testaustiimin kyky täyttää nämä tarpeet.

Testaustiimi toimii itsenäisenä ja mahdollisimman riippumattomana projektiryhmästä. Tällä toimintamallilla on pyritty löytämään mahdollisimman kriittinen tarkastelutapa kehitettäviin tuotteisiin ja aikaansaamaan testaajalle vahva tuki toisilta testaajilta. Kiireellisinä aikoina testaajien siirtäminen projektista toiseen vahvistaa suoriutumista testaustyöstä. Näin varmistetaan työn suorittaminen siten, että testaustyön keskeneräisyys ei estä tai pitkitä toimitusta. Lisäksi on lähdetty vahvistamaan hiljaisen tiedon jakamista testaajien kesken ja näin osaltaan vahvistetaan kunkin testaajan osaamista. Hiljaisen tiedon jakaminen on vasta alussa, eikä ole ollut suunnitelmallista tai järjestelmällisesti organisoitua. Käytännössä tiedon jako on ollut lähtöisin testaajista itsestään ja pääosin suppeasti jaettua tietoa. Tarve tiedon jakamiselle on kuitenkin tiedostettu jo useita vuosia sitten, kun yrityksessä perustettiin testaajalähtöisesti testausfoorumi. Tarkoituksena oli koota eri tiimeissä toimivat testaajat yhteen jakamaan osaamistaan. Foorumi on kokoontunut vaihtelevasti. Tämän vuoksi tiedon jako ei ole ollut riittävän aktiivista.

Tutkimuskohteena olevan testaustiimin osalta työn kehittäminen on kesken. Tiimi toimii itsenäisesti. Tiimille on määritetty testauspäällikkö, jonka tehtävänä on kehittää, ohjata ja organisoida tiimin toiminta. Testaajien osaamisen johtamiseen vaikuttaa kuitenkin se, että testauspäällikkö ei ole testaajien henkilöesimies. Testauspäällikön ja henkilöesimiesten välille ei ole kehitetty yhteistä työskentelyä. Henkilöesimiehet ja testauspäällikkö eivät jaa keskenään tietoa työntekijöiden päivittäisen työskentelyn onnistumisesta, palautteista tai osaamisen kehittämistavoitteista. Näkökulmat eivät näin ollen ollen huomioi yksilön, testaustiimin ja koko yrityksen tarpeita. Vuorovaikutuksen puute aiheuttaa väistämättä

tilanteen, jossa testaajien osaamista kehitetään yksilön ja henkilöesimiesten tuntemuksen mukaan. Huomioiduksi ei nouse testauksen tavoitteet koko tiimiä ja yritystä ajatellen. Yhteinen suunta on merkittävä siksi, että vuorovaikutuksella voidaan tukea yrityksen visiota ja strategiaa noudattamalla toimintajärjestelmää. Testaustiimin kehityksen yksiselitteinen suunta edesauttaa tuotteiden entistä parempaa testausta. Lisäksi vahvistetaan tuotteille asetettujen vaatimusten täyttämistä yksilön osaamista kehitettäessä niin tiimin kuin yksilön näkökulmasta.

On havaittavissa, että kaikilla tiimissä toimivilla ei ole ymmärrystä seuraavista asioista: mitä testaus tarkoittaa, mitä testaustyöhön kuuluu ja miten testausta tulisi kehittää. Vahvimmin puutteet ovat nähtävissä niissä, jotka toimivat testaustiimin ulkopuolella. On selvää, että myös testaustiimin sisällä tarvitaan osaamisen kehittämistä. Testaustiimin työn ymmärtäminen yksityiskohtaisesti vaatii panostusta. Tällä hetkellä erilaiset mielipiteet, vahvat persoonallisuudet ja testaustiimin resurssien pieni osuus suhteessa koko tiimin resursseihin aiheuttavat epävarmuutta suorittaa testaustyötä. Tilanne heikentää mahdollisuutta ohjata tiimiä vakaasti ja hallitusti. Testaustiimin toiminnan kehittymistä ei välttämättä tunnisteta tiimin ulkopuolella. Kehittäminen keskittyy usein sisäisten toimintamallien tehostamiseen. Rajapinnoissa toimiminen edellyttää, että kehitystoimintaan osallistuisi avoimesti myös projektiryhmät. Heidän olisi tarpeen tarkastella omia toimintojaan kriittisesti ja noudattaa toimintatapoja. Näin helpotetaan testaustiimin suoriutumista testaustyöstä.

Tutumpi tarkastelutapa on tällä hetkellä ollut se, että tehostamista toivotaan testaustiimin suunnalta helpottamaan projektiryhmien työskentelyä. Malli perustellaan usein asiakkaan tarpeilla. Kehittäminen on kuitenkin kaksisuuntaista. Onnistumisen edellytyksenä on molempien, niin testaustiimin kuin projektiryhmien kriittinen suhtautuminen omaan työskentelyynsä tiimien rajapinnoissa. Tällä hetkellä projektiryhmillä on vapaus suunnitella työskentely ryhmälle toimivalla tavalla kuitenkin siten, että he noudattavat yrityksen toimintajärjestelmää. Tällainen toimintamalli asettaa testaustiimin testaajille ison haasteen, koska heidän on opeteltava jokaisen projektin yhteydessä vuorovaikutustavat erikseen. Käy-

tännössä aikaa kuluu projektin alusta, kun ryhmäläiset etsivät keinoja jakaa tietoa keskenään. Monesti kiireinen projektiaikataulu aiheuttaa sen, että työhön ryhdytään nopeasti ja vuorovaikutuksen puute. Usein hankaluudet ratkaistaan sitä mukaa, kun ongelmat tulevat esiin. Projektiryhmät eivät myöskään tällä hetkellä jaa hyviä käytänteitä aktiivisesti keskenään. Testaustiimin toiveet tai ehdotukset toimintatavoiksi eivät aina saavuta toivottua reaktiota. Testaustiimin toiminnan näkökulmasta olisi toivottavaa, että koko tiimissä löydettäisiin yhtenevät käytännöt projektien läpivienniksi. Tällöin sekä testaustiimi että projektiryhmät huomioisivat toistensa tehtävät aktiivisesti, tutustuisivat toistensa työskentelyyn ja toisaalta etsisivät mahdollisimman yhtenäiset toimintamallit erityisesti vuorovaikutuksen osalta projektista riippumatta.

1.2 Työn aihe ja tavoitteet

Kehittämistehtävän aihe pohjautuu aiemmin laadittuun opinnäytetyöhön eli on sille jatkumo. Työn valmistumisen jälkeen nousi esiin seuraavat kaksi kysymystä, joihin työ ei antanut vastausta:

- Miten testausresurssit saadaan jaettua kaikkien projektien kesken siten, että testauksen määrittely, suunnittelu, toteutus ja tulosten analysointi saadaan vietyä läpi käytettävissä olevassa aikataulussa ja kattavasti?
- Miten projektien sisältöön ja aikatauluun tehdyt muutokset saadaan huomioitua testauksen resursoinnissa tehokkaasti?

Monipuolinen ja järjestelmällinen testaustyö sekä testausresurssien jakaminen eri projektien kesken ottaen edellyttää, että testaustiimissä on käytettävissä erityyppisten testausosa-alueiden osaajia. Tarpeelliset osaamisalueet vaihtelevat osaamisesta ja projektien tarpeista riippuen. Suurin tarve on testauksen perustehtävien osaajista, sekä muiden osa-alueiden erityisosaajista. Tiimin kehittyminen uusien projektien monimutkaistuessa vaatii testausosaamisen kasvattamis-

ta ja laajentamista. Näin ollen testaustiimin jäsenten yksilöllisen kehittymisen tulisi täyttää myös koko testaustiimin kehittymistarpeet. Näin voidaan tukea testaustyön kehittämistä organisaatiossa.

Tämä kehittämistehtävä käsittelee testaustiimin ohjaamista ja tiimin jäsenten osaamisen kehittämistä. Näkökulmana on erityisesti tiimi, mutta myös yksilöiden ajatukset kehittymisen suhteen huomioidaan. Työn tarkoituksena on tarkastella testauksen perustehtävää ja arvioida millaisia osaamisia testaustiimissä tarvitaan. Kehittämistehtävä etsii vastauksia seuraaviin tutkimuskysymyksiin:

- Mitä testausosaamista tarvitaan testaus toimintojen läpiviemiseksi?
- Miten huomioidaan testaajien vahvuudet ja heikkoudet resurssijaon yhteydessä?
- Miten arvioidaan testaustiimin kehittymistarpeet organisaation näkökulmasta?
- Miten arvioidaan testaajien kehittymistarpeet yksilöinä ja tiimissä?
- Miten kehitetään testaajien keskinäistä vuorovaikutusta testaustyön suorittamisen yhteydessä?

Osaamisalueiden ja osaamisten määrittämisen kautta voidaan arvioida testaustiimin tämän hetkinen tilanne. Sitä kautta testaustiimin ohjaamiseksi on tavoitteena löytää työkalut testaustiimin kehittämisen suhteen ja ennen kaikkea osaajien riittävä varmistaminen projektien tarpeet huomioon ottaen. Lisäksi saadaan arvokasta tietoa yksilön kehittämiseksi, sekä mahdollisesti koko tiimin käytettävissä olevien resurssien riittävyyden arvioimiseksi. Projektien onnistunut läpivienti edellyttää, että myös testaustiimi kykenee reagoimaan projekteissa tapahtuviin nopeisiin muutoksiin mahdollisimman pienellä viiveellä ja käyttämällä työn suorittamisessa erityisosaajia. Tähän kyetään helpommin, kun testaustiimi pysyy kohdistamaan kehittämisen niille osa-alueille, joita testaustyön läpiviemisessä tarvitaan kriittisimmin ja riittävän ammattitaidon omaavia osaajia näihin alueisiin löytyy riittävästi.

1.3 Tutkimuksen rakenne ja työn rajaus

Testauksen osaamisien määrittäminen ja toimeksiantajan testaustiimin osaamistason arviointi on kokonaisuudessaan erittäin laaja aihe. Testaustiimin osaaminen on myös kehittynyt kehittämistehtävän laadinnan aikana paljon. Kehittämistehtävää on rajattu siten, että työ kattaa ainoastaan testaustiimin osaamistarpeiden määrittämisen eri näkökulmista. Tarkoituksena on hahmottaa keinot testaustiimin osaamisen jatkuvan arvioinnin ja kehittämisen tueksi. Varsinaisen tiimin osaamisen arviointi ja sen perusteella tehtävä osaamistarpeen määrittely jätetään kehittämistehtävän ulkopuolelle. Samoin tavoiteltujen osaamisten määrittely yksityiskohtaisesti on rajattu työn ulkopuolelle, koska kyseiset yrityskohtaiset teknologiaratkaisut eivät sovellu välttämättä yleisesti yritysten tarpeeseen.

Aiheen tarkastelu suoritetaan ohjelmistotestauksen näkökulmasta. Kehittämistehtävä jakautuu neljään osaan. Ensimmäisessä osassa (1 Johdanto) kuvataan tutkimuksen taustat, tavoitteet ja työn rajaus. Toisessa osassa (2 Osaamisen ja ohjaamisen kehittämismenetelmät) tarkastellaan teorian pohjalta erilaisia näkökulmia testauksen osaamisalueiden ja osaamisten määrittämiseksi. Tarkasteltavat näkökulmat on jaettu testausnäkökulmaan ja yleiseen osaamisen johtamisen näkökulmaan. Testauksen näkökulmasta merkittävimmät tarkasteltavat asiat ovat ISEB:n testausstandardit. Erityksen tarkastelun kohteena ovat määritelmät ja ohjeet testauksen ohjaamiseksi ja johtamiseksi. Toisaalta etsitään vastauksia myös testauksen tunnettujen kehittämismallien kautta, kuten TMMi, TPI, CTP ja STEP. Yleisen osaamisen johtamisen näkökulmasta keskitytään aluksi perustehtävän tarkasteluun ja muutoksen johtamiseen. Samassa yhteydessä keskitytään aiheiden merkitykseen tiimin ohjaamisessa. Tämän jälkeen kuvataan osaamisen johtamisprosessi ja etsitään keinot määritellä testauksen osaamiset tätä prosessimallia käyttäen.

Kehittämistehtävän kolmas osa (3 Soveltuvuus testauksen ohjaamiseksi) kokoaa teorian yhteen etsien soveltuvat osat standardeista ja kehittämismalleista.

Näiden pohjautuen määritellään testauksen osaamiskartta. Lisäksi kuvataan keinoja testaustiimin ohjaamiseksi ja osaamisen johtamiseksi. Viimeinen osa (4 Pohdinta) keskittyy kokoamaan työn tulokset yhteen. Teorian ja johtopäätösten perusteella arvioidaan tulosten vastaavuutta esitettyihin tutkimuskysymyksiin. Lisäksi arvioidaan osaamisprosessin loppuun viemisen, testaustiimin osaamisen arvioinnin käynnistämistä ja näin ollen testaustiimin ohjaamisen ja johtamisen keinoja saatujen tulosten pohjalta.

Kehittämistehtävän tiedon hankinnassa hyödynnetään erilaisia tapoja. Tutkimuksen teoriapainotteinen osa koostuu lähdeaineistona käytetyistä kirjoista, artikkeleista, Internet-lähteistä ja erilaisista seminaareista sekä koulutuksista opitusta tiedosta. Tutkimuksen taustat pohjautuvat aiemmin tehtyihin opinnäytteihin ja niiden pohjalta tehtyihin muutoksiin toimeksiantajan testaustiimissä. Arviointimenetelmien soveltuvuuden yhteydessä hyödynnetään samaa lähdeaineistoa kuin teoriapainotteisessa tutkimuksessa. Omat pohdinnat sekä havainnot, joita olen kehittämistehtävän laadinnan yhteydessä suorittanut testaustiimissä, täydentävät tehtävää. Työn edetessä on joitakin opittuja asioita hyödynnetty epäsystemaattisesti käytännön työskentelyssä. Sitä kautta on saatu palautetta testaustiimin jäseniltä työskentelyn ja vapaamuotoisten keskustelujen yhteydessä.

1.4 Käsitteet ja käytettävät termit

Dokumenttiin liittyvät käsitteet ja termit on kuvattu erillisessä liitteessä (Liite 1).

2 OSAAMISEN JA OHJAAMISEN KEHITTÄMISMENETELMÄT

Kehittämismenetelmät kattavat kaksi näkökulmaa. Näistä ensimmäinen on yleinen tiimien osaamiseen ja johtamiseen liittyvä näkökulma, joka pitää sisällään myös tarkastelun yksilön ohjaamisesta ja organisaation ohjaamisesta. Tarkastelua täydennetään näkökulmalla, joka tukee tiimin ohjaamista ja johtamista ohjelmistotestauksen osalta. Tarkastelu suoritetaan huomioiden testausstandardit huomioiden, sekä erilaiset testauksen kehitys- ja kypsyysmallit.

2.1 Johtaminen perustehtävän ja muutoksen kautta

Työelämän vaatimukset ovat kasvaneet ja työntekijöihin vaikuttaa voimakkaasti työntekijöiden osaamisvaatimusten kasvu. Tehokkuutta on pyritty maksimoimaan ja samanaikaisesti työn suorittamisen tulos- ja laatuvaatimukset ovat myös kasvaneet. Työtä on suoritettava jatkuvasti muuttuvassa ympäristössä ja viime vuosikymmenen aikana muutosnopeus on kasvanut. Tehostamistoimenpiteet ovat vaikuttaneet siten, että työntekijöiden kiire ja työn suorittamisen nopeus ovat korostuneet entisestään. Kun tiedonvälitys on parantunut, ei jaettavaa tietoa välttämättä jalosteta riittävästi ennen tiedon jakamista. Tiedontulvasta on hankala erottaa oleellinen asia. Myös vuorovaikutus- ja yhteistyötaitojen merkitys työyhteisössä on korostunut, jotta tiedon ja osaamisen jakamista saataisiin kasvatettua. Samalla jokaisen työntekijän tulisi kyetä tulkitsemaan tieto yksiselitteisesti, jolloin vuorovaikutus muuttuu haastavammaksi. Käytännössä kuitenkin työelämän kasvaneet vaatimukset ovat aiheuttaneet epävarmuutta työntekijöihin. Yksilön tarve säilyttää oma arvonsa työmarkkinoilla on kasvanut kilpailun kasvun ja pätkätöiden vuoksi. Työtä ei voida enää käsittää vain työnä, vaan kyseessä on työstä riippumatta ammatti. (Järvinen, 2009 & 2010).

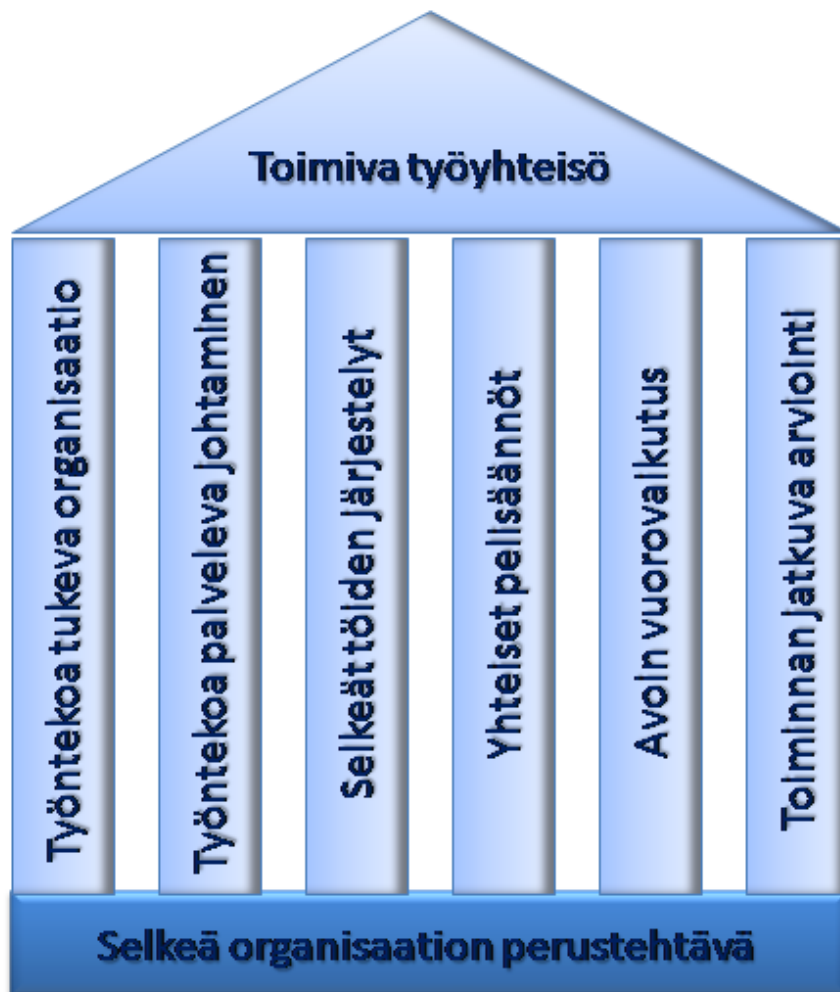
Johtaminen perustehtävän kautta

Työelämän haasteista riippumatta jokaisen työyhteisön tulisi olla toimiva. Esi- miesten suurena haasteena on saada työntekijät ymmärtämään oma perustehtävänsä ja suorittamaan sitä. Tarkoituksena on välttää tilannetta, jossa varsinainen perustyö muuttuu työntekijöiden omien olettamusten ja näkökantojen perusteella. On selvää, että työntekijöiden ajatuksia tarvitaan kehittämään työmenetelmiä, mutta varsinainen perustyö ei muutu tehostamiskeinojen myötä. Näiden rajojen ylläpitoon tarvitaan esimiestä, joka varmistaa perustyön tekemisen ja priorisoi työskentelyä. Esimiehellä on selkeämpi kyky nähdä kokonaisuus kuin työntekijällä, joka toimii suppeammalla alueella työyhteisössä. (Järvinen, 2009 & 2010).

Perustehtävän määrittäminen ei ole aina helppoa. Kyseessä on työ, joka pitää suorittaa tarvittavan tuotoksen aikaansaamiseksi. Perustehtävä voi olla esimerkiksi rakennustyömaalla vesierityksen teko tai ohjelmistokehitystyössä asiakasvaatimukseen perustuvan toiminnon suunnittelu ja koodaaminen, jonka tuotoksena syntyy toiminto. Työyhteisöt, joissa tuotos käsitetään abstraktina, saattaa perustehtävän hahmottaminen olla hankalampaa. Jokaisella työntekijällä kuitenkin on perustehtävä, jota suorittamaan hänet on työyhteisöön palkattu. Koko tiimin toiminnan onnistumisen kannalta on äärimmäisen tärkeää, että jokainen tiimissä työskentelevä. Tiimin kanssa rajapintojen kautta työskentelevien tulisi ymmärtää tiimin perustehtävä samalla tavoin tehtävää suorittavan tiimin kanssa. (Järvinen, 2009 & 2010).

Toimiva työyhteisö ei rakennu kuitenkaan yksittäisten perustehtävien varaan, vaan työyhteisössä on oltava selkeästi määritelty kaikki tarvittava työ. Jokaiselle työlle on määriteltävä myös työn suorittaja. Jos joitakin työtehtäviä jää perustehtävien ulkopuolelle, muodostuu harmaa alue. Prosessien linkittyminen ei ole tällöin perustehtävien suorittamisen kautta aukoton. Käytännössä tällaiset aukkokohdat aiheuttavat epätasapainoa työyhteisössä ja muodostavat ongelmia tiimien sisällä ja rajapinnoissa. Yhteistyö ei toimi hyvin ja kaikkea työtä ei aika-

tauluteta. Se saattaa vaikuttaa suoraan toimitusaikatauluihin ja hankintoihin. Selkeät ja kattavat perustehtävät vaativat tuekseen työntekoa tukevan organisaation ja selkeät töiden järjestelyt. Myös yhteiset pelisäännöt, jota yhteisössä noudatetaan, on laadittava. Lisäksi tarvitaan avointa vuorovaikutusta ja toiminnan jatkuvaa arviointia. Erityisesti työntekoa palveleva johtaminen on tärkeää. Toimiva työyhteisö on kuvattu seuraavassa kuvassa (Kuva 1). (Järvinen Pekka, 2009 & 2010).



Kuva 1 Toimiva työyhteisö (Järvinen Pekka, 2009 & 2010).

Jos perusta ei ole koko tiimillä hallussa tai jokin toimivaa työyhteisöä tukevista pilareista ei ole ohjattu ja hallittu, ei työyhteisö voi toimia riittävän hyvin. Jokaisen osa-alueen tasapainoinen tukeminen edellyttää organisaatiolta, tiimiltä ja

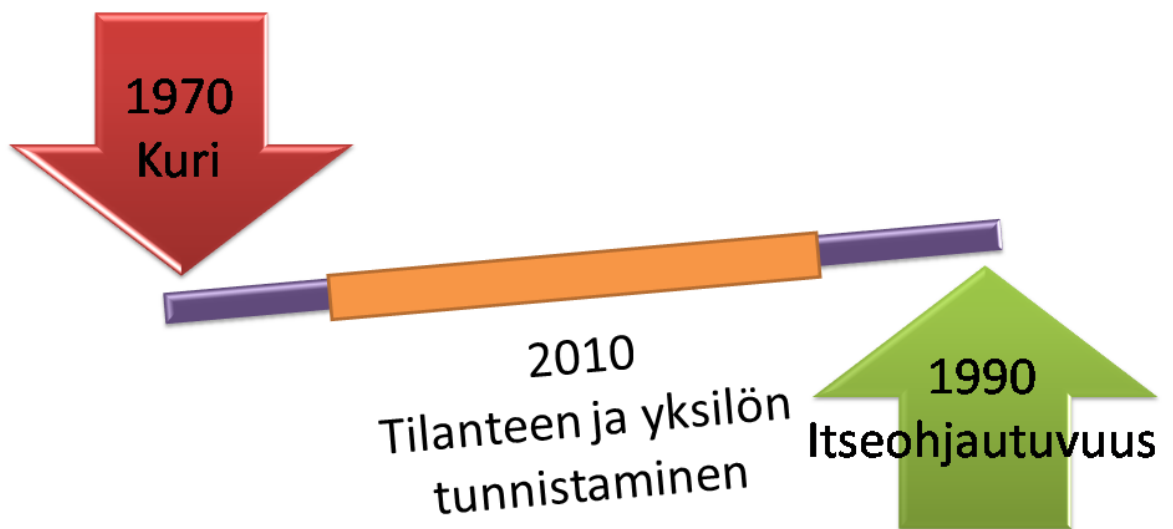
yksilöltä sitoutumista ja yhteistyötä. Tällöin työyhteisössä työskentely voi olla sulavaa. Kaiken perustana on perustehtävän suorittaminen annettujen ohjeiden ja yhdessä sovittujen toimintatapojen mukaisesti. Usein näin ei kuitenkaan ole.

Yksittäisen työntekijän tai tiimin erilaiset tulkinnat aiheuttavat väistämättä ongelmia koko työyhteisön onnistumiseen ja tasapainoon työn suorittamisen kannalta. Tällainen tilanne voi olla esimerkiksi tehtävien suorittamistapa tai yhteisten pelisääntöjen noudattamatta jättäminen. Usein tasapainon puuttuminen aiheuttaa toisissa työntekijöissä tarvetta seurata yhteisistä pelisäännöistä poikennutta työntekijää. Perustehtävän suorittaminen hämärtyy. Erityisen alttiina ovat uudet työntekijät, jotka vasta yritykseen palkattuina yrittävät oppia työyhteisön toimintatavat. Uusi työntekijä tarvitsee aktiivista ja säännöllistä palautetta työstään, sekä ohjausta työn suorittamiseksi. Siinä missä palautteen anto on tärkeää uudelle työntekijälle, on sitä annettava myös pidempään työyhteisössä työskennelleille. Palaute vaikuttaa suoraan työntekijän motivaatioon ja vuorovaikutuksen ylläpitoon. (Järvinen, 2009 & 2010).

Aika ajoin esimiehen tehtävänä on palauttaa työyhteisölle muistiin heidän perustehtävänsä. On kuvattava kokonaistavoite ja kyseisen tiimin osuus kokonaistavoitteessa. Näin saadaan työntekijät paremmin sitoutettua työhön ja ymmärtämään yhteisön merkitys koko yrityksen menestyksen kannalta. Työyhteisön ongelmista 90 %:a johtuvat siitä, että perustehtävä ja sitä tukevat toiminnot eivät ole kunnossa. Työyhteisö ei toimi tehokkaasti, ilmapiiri on huono tai motivaatio alhainen. Ongelman ilmaantuessa työntekijät tulkitsevat asiaa ja jakavat sen muiden työntekijöiden kanssa. Ryhmäkeskustelut ilman kaikkia osapuolia usein mutkistavat asioita. Kaikki nämä ongelmat on ratkaistavissa selkeällä perustehtävän määrittämisellä, noudattamisella, organisaation tuella ja yhteisillä pelisäännöillä, sekä esimiestyöskentelyllä. Ennen kaikkea aktiivinen ja suora vuorovaikutus, sekä ongelmien selvittäminen ennen niiden paisuttelua parantaa ilmapiiriä ja pitää työyhteisön toimivana. Vain 10 % ongelmista johtuu henkilöistä itsestään tai heidän henkilökohtaisista ongelmistaan. Riippumatta syystä on esimiehen seurattava ja nähtävä mahdolliset ongelmakohdat riittävän ajoissa.

Riittävän ajoissa selvitettyillä ongelmilla mahdollistetaan työyhteisön palauttaminen noudattamaan perustehtävää. (Järvinen, 2009 & 2010).

Eri tiimien yhteistyö edellyttää sitä, että esimiehet tukevat ja keskustelevat avoimesti keskenään ja sopivat toisiinsa liittyvistä asioista yhdessä. On tärkeää, että työyhteisö ja esimies ymmärtävät myös muiden tiimien toiminnan ja kyvyn toimia osana yritystä. Esimiehen on oltava läsnä työntekijöille. Aina läsnäolo ei tarkoita fyysistä läsnäoloa päivittäin, vaan olemalla työntekijöiden käytettävissä. Lisäksi tulee osoittaa vahvasti kiinnostus työyhteisölle ja etenemiselle perustehtävän suorittamisessa. Nykypäivän ongelmana työyhteisössä on johtamisen puuttuminen. 1970-luvulla tiimejä johdettiin organisaatiossa ylhäältä päin erittäin voimakkaassa kurissa (vrt. Kuva 2).



Kuva 2 Johtamisen kehittyminen

Työntekijät noudattivat täysin annettuja ohjeita, eikä heidän mielipiteitään välttämättä huomioitu. Pari vuosikymmentä myöhemmin siirryttiin toiseen ääripäähän, jossa organisaatiot siirtyivät täydelliseen itseohjautuvuuteen. Tämän seu-

rauksena 2000-luvulla tiimien sisällä käytiin valtataisteluista siitä, kenen mukaan tiimissä toimittiin ja vahvimmat ohjasivat tiimejä. Perustehtävää ei enää muistettu ja johtajuus joutui kriisiin. (Järvinen, 2009 & 2010).

Nyt 2010-luvulla on havaittu, että esimiehiä tarvitaan - johtamista tarvitaan. Ongelmana on kuitenkin se, että esimiehet eivät toimi vain esimiesten roolissa. Työ on heille sivutyötä, jolloin aikaa johtamiseen ei löydy. Lisäksi tiimeissä roolit, vastuut ja tehtävät ovat epäselviä. Työntekijöitä nimitetään ja heille annetaan uusia tehtäviä. Työyhteisön informointi, sekä selkeiden vastuiden ja valtuuksien määrittäminen jää kuitenkin tekemättä. Käytännössä niin esimies kuin muu työyhteisö kuluttavat merkittävästi aikaa etsiessään rajojaan. Ongelmia tuottavat myös esimiesten valitsemistavat esimiehen tehtävään. Usein tehtävään valitaan henkilö, joka halutaan palkita. Hän ei välttämättä ole kuitenkaan soveltuvin työtehtävään ja erityisesti toimimaan työntekijöiden opastajana ja ohjaajana. Kun hänet nimitetään tehtävään, jää hän yksin. Esimiestyö vaatii vahvaa tukea muulta organisaatiolta. Tehtävän vastaanottanut työntekijä pitää kouluttaa tehtävää johtamiskoulutuksella ja perehdyttää yrityksessä esimiesten käytössä oleviin toimintatapoihin. (Järvinen, 2009 & 2010).

Ennen kaikkea tehtävän vastaanottavan tulee olla motivoitunut ja johtamisen pitää olla yrityksessä tasalaatuista johtajista riippumatta. Esimiehen tulee olla kiinnostunut alaisistaan ja saada ryhmä tekemään tulosta. Hänen tulee myös ymmärtää, että hänen tehtävänsä ei rajoitu ainoastaan perustehtävästä suorittamisen valvontaan ja työnteon edellytyksistä huolehtimiseen. Hänen vastuullaan on päivittäin työyhteisön henkisestä tasapainosta huolehtiminen. Mahdolliset ristiriidat ja ongelmat on saatava selvitettyä ajoissa, eivätkä saa vaikuttaa työn suorittamiseen. Lisäksi esimiehen tulee eri tavoin haastaa henkilöstö arvioimaan ja kehittämään toimintaa asiakaspalvelulähtöisesti. Asiakkaan roolissa voidaan nähdä tilaaja, yrityksen johto ja ennen kaikkea sisäistä toimintaa tarkasteltaessa ryhmän rajapinnoissa työskentelevät toimijat. (Järvinen, 2009 & 2010).

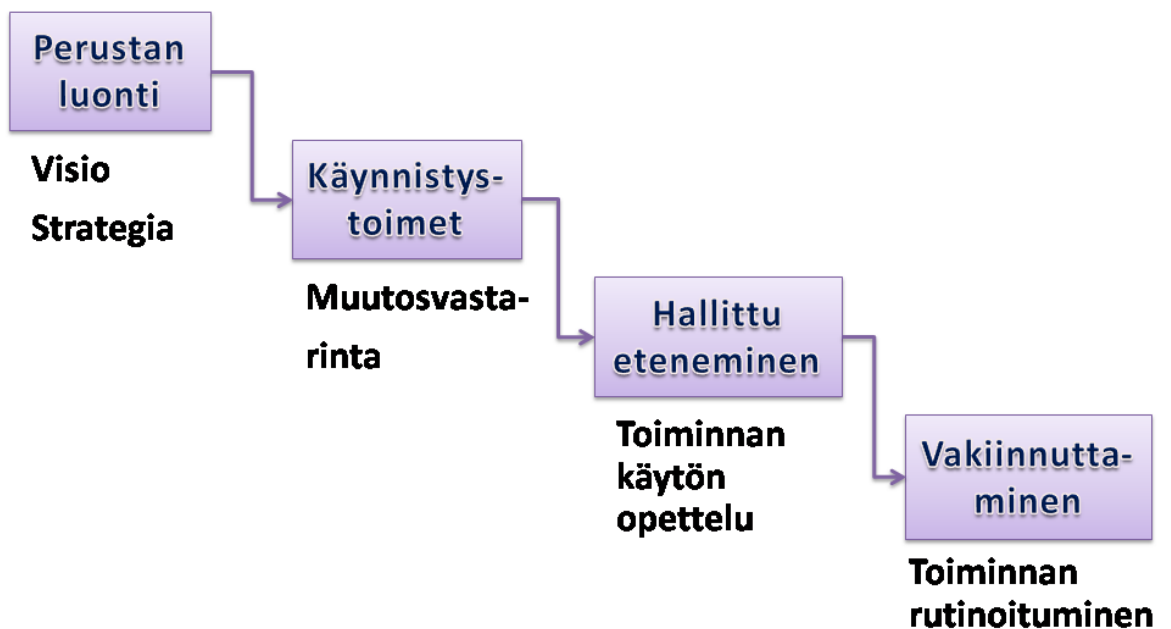
Suuri haaste nykyjohtamisella on itseohjautuvuuden säätelyssä. Esimiesten tulisi liikkua 70-luvun kurin ja 90-luvun itseohjautuvuuden välillä huomioiden yksilöiden, tiimien ja käsiteltävän asian luonne. Tiimit eivät toimi ilman vahvaa johtamista, jossa huomioidaan tehtävän ja yksilön keinot toimia itseohjautuen. Myös nykyorganisaatioissa paljon käytetty matriisiorganisaatio aiheuttaa hankaluutta, jos organisaation tehtäviä ja työkakoja ei ole määritelty selkeästi. Hyvän esimiehen tulisi olla jämäkkä, joka tekee selkeät päätökset. Hänen tulee ottaa huomioon yrityksen tarpeet ja valvoa niiden toteutumista. Työntekijöille hänen tulee olla ymmärtäväinen ja kuunnella heitä. Tiimin ohella on otettava huomioon myös yksilön tarpeet, rakennettava luottamus tiimin kanssa. Lisäksi on otettava avoimesti vastaan ehdotuksia työntekijöiltä toiminnan parantamiseksi ja valittava toimintatavat tilannekohtaisesti. Käytännössä esimies on kaikkien kaveri, joka joutuu tekemään työnantajan kannalta tiukkojakin ratkaisuja. Hän ei saa kuitenkaan olla liian lähellä työntekijöitä, mutta on oltava aina käytettävissä. (Järvinen, 2009 & 2010).

Esimestä voidaan kuvata säiliönä, joka ottaa vastaan tiedon ja tunteet työyhteisöltä. Näitä ovat esimerkiksi ongelmat, ristiriidat ja pelot. Hän prosessoi asiat ja pyrkii jäsentelemään niitä ymmärrettävämpään muotoon. Hän antaa palautteen työntekijöille ja selvittää asioita heidän kanssaan. Lisäksi hänellä on hajulukko, koska hän ei saa liikaa näyttää omia tunteitaan työyhteisössä. Onnistuakseen työssään hän tarvitsee kuitenkin kollegan tai esimiehen, jolle hän voi omat huolensa purkaa. Omat puutteensa hyväksyvänä, innostava ja riskejä ottavana hänellä on oltava kyky käsitellä pettymyksiä. Tunne omasta kyvykkyydestään, innostavat visiot ja hyvä todellisuuden taju ja tietynlainen varovaisuus tarvittaessa auttavat onnistumaan. Esimiestyöskentelyn perustana on esimiehen oma kyky ottaa ja ansaita johtajuus omaa persoonaansa ja auktoriteettiaan käyttäen. Tässä onnistuakseen hän tarvitsee organisaatiolta muodollisen vallan ja vastuut. Lisäksi on saatava jatkuva tuki työlle ja henkilöstön hyväksyntä. (Järvinen, 2009 & 2010).

Muutoksen johtaminen

Perustehtävän suorittamisen kehittäminen ja tehostaminen organisaatiossa vaatii usein muutosta toimintatapoihin. Lisäksi tarvitaan erittäin avointa vuoro-vaikutusta eri toimijoiden kesken. Nopeasti kehittyvän teknologian seurauksena on muutos tullut osaksi yritysten jokapäiväistä työtä. Näin yritys kykenee vastaamaan markkinoiden haasteisiin ja osaltaan ylläpitämään omaa kilpailukykyään. Muutoksen läpivienti on usein hidasta. Tähän on usein syynä tiimin jäsenten erilainen kyky mukautua muutokseen. Muutoksen ja sen vaikutuksien puutteellinen kuvaaminen eri osapuolille voivat hidastaa läpivientiä. Myös testaus- tiimin ohjaamisessa ja kehittämisessä on muutos läsnä jatkuvasti, jolloin muutoksen läpiviennin hankaluudet on kyettävä tunnistamaan. (Järvinen, 2009 & 2010).

Muutoksen vaiheet voidaan jakaa seuraavasti: perustan luonti, käynnistystoimet, hallittu eteneminen ja vakiinnuttaminen (Kuva 3).



Kuva 3 Muutoksen vaiheet

Muutoksen perimmäisenä lähtökohtana nähdään johdon vahva sitoutuminen tavoitteisiin ja strategiaan. Tämän lisäksi perustan luonti on äärimmäisen tärkeää. Yrityksellä on oltava selkeä visio ja näkemys siitä, mitä muutoksella tavoitellaan. Lisäksi on tunnistettava miten muutos tullaan yrityksessä viemään läpi mahdollisimman selkeästi ja suoraviivaisesti. Käynnistystoimien aikana saattavat herätä ensimmäiset pelkotunteet muutoksen vaikutuksista. Pelon erilaiset tyypit jakautuvat aiheettomiin ja aiheellisiin pelkoihin. Aiheettomat pelot ovat sellaisia, jotka saadaan poistettua korvaamalla turhat huhut ja olettamukset totuudella. Aiheelliset pelot voidaan jakaa myös kahteen osaan: niihin joihin voidaan vaikuttaa ja niihin joihin ei voida vaikuttaa. Käytännössä jälkimmäisiä on vain siedettävä. Ensimmäisiä varten on pohdittava toimenpiteet. (Järvinen, 2009 & 2010).

Käyttöönottoaiheessa esiintyy yleensä suurin muutosvastarinta. Tämä on hyvin negatiivis-sävytteinen sana. Merkitys voidaan kuvata myös siten, että kyseessä on yksinkertaisesti halu ymmärtää muutos ja siihen liittyvät syyt ennen kuin muutos voidaan hyväksyä. Organisaation kehittymisen kannalta on tärkeää ymmärtää muutosvastarinta ja saada se minimoitua mahdollisimman aikaisessa vaiheessa. Toisaalta muutosvastarinta on opittava myös hyväksymään. Jos muutosta ei millään muotoa vastusteta, ei sitä myöskään oteta tosissaan. Tällöin varsinaista muutosta ei voi tapahtua. (Järvinen, 2009 & 2010).

Muutosvastarinta lähtee syvimmillään yksilön omasta sisimmästä. Ihmisellä on tarve tietää, mistä muutoksessa on kyse. Muutoksen täytyy olla mielekäs. Toisaalta ihminen haluaa tietää, voiko hän itse vaikuttaa muutokseen. Näitä tarpeita kutsutaan mielekkyysspyrkimykseksi ja itsemääräämispyrkimykseksi. Kolmas liittyy identiteettiin ja itsetuntoon. Ihminen tahtoo tietää, mikä on hänen arvonsa muutoksessa. Lisäksi hänen on kyettävä hahmottamaan miten toimia muutoksen jälkeen, sekä selviääkö hän muutoksesta. Jos tunne on näiden kysymysten läpikäytyä positiivinen, on työntekijä valmis muutokseen. Toisaalta työntekijä voi kokea hallinnan tunteensa katoavan ja mahdollisesti oman psyykkisen tasapainon horjuvan. Tämän kokemuksen seurauksena käynnistyy stressireaktio, josta

palautuminen vaatii työntekijältä uutta energiaa eli tarkempaa tietoa muutoksesta. (Järvinen, 2009 & 2010).

Selviytyminen stressin aiheuttamasta tilasta vaihtelee erilaisilla työntekijöillä. Toisella horjuntavaihe kestää lyhyen hetken ja hän voi siirtyä muutoksessa opetteluun piankin. Toisilla muutos vie pitkään ja he saattavat tarvita apua esimieheltä tai työtovereiltaan sopeutua tulevaan tilanteeseen ennen itsenäisen työskentelyn onnistumista. Joissakin tapauksissa stressi voi näkyä psyykkisenä, fyysisenä ja sosiaalisena. Käytännössä jokaisen tulisi oppia tunnistamaan oma stressireaktionsa, jotta voisi pysäyttää sen mahdollisimman nopeasti. Pahimmissa tapauksissa stressi aiheuttaa stressikierteen, joka voi johtaa vakavaan uupumiseen. (Järvinen, 2009 & 2010).

Muutoksen ristiriita syntyy usein siitä, että jokainen työntekijä toimii työssään omalla mukavuusalueellaan. Toisilla tämä alue on laaja, mutta toisilla suppeampi. Usein työntekijää kiinnostaa uusi asia, joka on varmaankin haasteellinen ajatuksen tasolla. Kun muutostarpeesta tulee todellista, syntyy automaattisesti epävarmuuden tunne. Tämä ristiriitaisuus ja työntekijöiden eriaikainen reagointi muutokseen aiheuttaa sen, että muutosten käyttöönotto on usein hidasta. Moneksi puoli vuottakin on lyhyt aika muutoksen läpiviemiseksi. Kun muutosta käydään konkreettisesti läpi, tuo tilanne väistämättä epävarmuutta, turvattomuutta ja poikkeamista opituista tavoista ja rutiineista. Uusi tilanne pakottaa ponnistelemaan ja oppimaan uutta, jolloin siirrytään epämukavuusalueelle. Oman mukavuusalueen kasvattaminen helpottaa toimimista muutostilanteissa. Tämä onnistuu parhaiten, kun työntekijä säilyttää ja kartuttaa osaamista, sekä rakentaa omaa kykyään omaksua ja siirtyä uuteen nopeasti. (Järvinen, 2009 & 2010).

Muutoksen johtamisessa on tärkeää, että vastaavat ilmiöt nähdään ensin itsessään. Vasta sen jälkeen ne voidaan nähdä muissa ja toimia tukena muutosta läpikäyvälle työntekijälle. Kun työntekijät kokevat epävarmuutta, luottavat he esimiesten vastauksiin ja epäselvyyden hälventämiseen. Käytännössä kuitenkin esimiehet käyvät itse läpi samaa myllerrystä. Muutoksesta riippuen esimiehetkin

ovat epätietoisia tulevaisuuden suhteen. Vaikka ihminen olisi kokenut jo useita muutoksia tai suuriakin muutoksia, on uuden muutoksen yhteydessä aina käytävä läpi epävarmuuden ja tunnemyllerryksen aiheuttamat vaikutukset. Joissakin tapauksissa lähiesimiehet saattavat itse estellä muutoksia. He vaikuttavat näin omiin alaisiinsa, koska kokevat muutoksen uhkana tai eivät itse sitä ymmärrä. Tämän vuoksi tärkeiden muutosten tapauksissa on tarpeellista selkiyttää muutos ymmärrettävään muotoon. Toisaalta voidaan ottaa muutosta viemään läpi sellainen henkilö, johon muutos ei tule koskemaan tai ainakin tukemaan lähiesimiestä työssään muutoksen läpiviemiseksi. (Järvinen, 2009 & 2010).

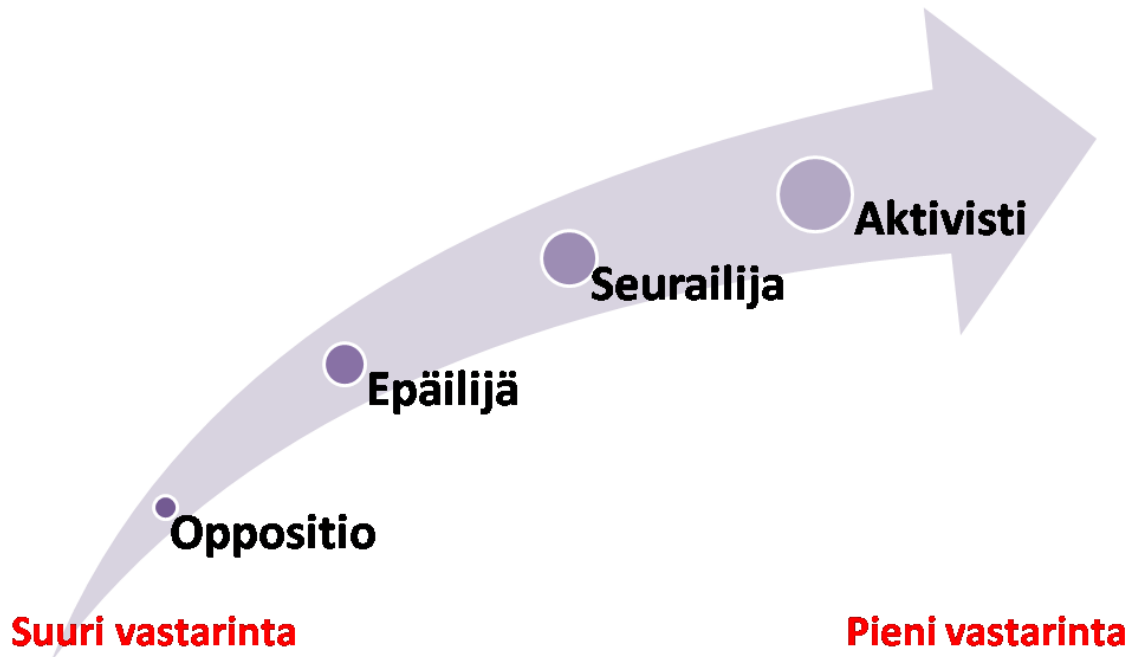
Uutta toimintatapaa ei noudateta ilman painetta, joten esimiehen on kyettävä satsaamaan paljon muutoksen läpivientiin ja oltava itse vahva. Kun pahin muutosvastarinta on ohi ja lyhyen muutosristiriidan läpikäyneet työntekijät pystyvät käsittelemään muutosta, siirrytään hallitun etenemisen vaiheeseen. Tässä vaiheessa muutosta opetellaan käyttämään. Vähitellen mukaan saadaan muut tiimin jäsenet heidän kyettyä käsittelemään muutoksen aiheuttama epämukavuus. Hyvän tuen ja opastuksen avulla muutos saadaan vietyä läpi. Viimeinen vaihe on vakiinnuttaminen. Käytännössä varmistetaan muutoksen noudattaminen ja muotoutuminen rutiininomaisena osaksi jokaisen toimintaa. (Järvinen, 2009 & 2010).

Muutoksia on erilaisia. Yrityksen sisäiset muutokset, joissa ei jouduta vähentämään henkilökuntaa, ovat yleensä helpompia kuin suuria irtisanomisia sisältävät muutokset. Suurissa muutoksissa saattaa työpaikalla alkaa kiertämään huhut ja pitkään jatkuvana työntekijät alkavat etsiä keinoja aliarvioida työtovereita esimiesten silmissä. Tällöin työilmapiiri kärsii ja saattaa aiheuttaa pysyviäkin haittoja vielä muutoksen jälkeen. Työnantajalle on haasteellista ja vaikeaa miettiä, missä kohdin tieto suuresta ikävästä muutoksesta annetaan. Epävarmuuden ajan pitäisi olla erittäin lyhyt, jotta työntekijä selviytyisi mahdollisimman lyhyellä epämukavuustilanteella. Työntekijälle pitää antaa myös aikaa sopeutua tulevaan tilanteeseen ennen muutoksen realisoitumista. Tärkeintä on kuitenkin se, että työntekijälle perustellaan selkeästi syyt irtisanomiselle. Ei riitä, että irtisa-

nomisesta informoidaan vain irtisanottavia, vaan yleiset perustelut irtisanomisille on kerrottava myös muulle henkilökunnalle. (Järvinen, 2009 & 2010).

Muutoksen onnistuminen edellyttää kykyä priorisoida muutostyö. Priorisoinnissa ja ajankäytössä huomioidaan normaalista työtehtävistä suoriutumisen muutoksen läpiviennin yhteydessä. On käytettävä arkista kieltä ja sellaisia sanoja, joita työntekijät ymmärtävät. Muutoksia saa olla vain rajallinen määrä kerrallaan. Lisäksi on huomioitava, että muutoksen läpivienti ei onnistu tiimin toimiessa itseohjautuvana. Muutos edellyttää aina esimiestä, joka ohjaa ryhmää. Työntekijöille on esiteltävä konkreettiset tavoitteet ja kyettävä näin pitämään paine alhaisena. Muutos aiheuttaa aina hinaajaefektin, jota voidaan käyttää hyödyksi. Sitä mukaa, kun työntekijät tulevat sinuksi muutoksen kanssa, heidät voi ottaa mukaan viemään muutosta läpi. Näin hinataan muita mukaan muutokseen. Hiitaammin muutokseen kykenevät vaativat pidempään perusteluja muutokselle, vaikka muut olisivatkin jo sitä toteuttamassa ja ymmärtäneet sen. (Järvinen, 2009 & 2010).

Edellä kuvatut edellytykset vaikuttavat suoraan muutoksen onnistumiseen. Esimiehen on kyettävä tunnistamaan ennalta työyhteisöstä jokaiselle työntekijälle oma rooli muutoksessa. Näin heitä osataan käsitellä oikein ja heidän tarpeensa muutoksen läpiviemiseksi osataan huomioida (Kuva 4).



Kuva 4 Roolit muutoksen yhteydessä

Aktivistit ovat muutoksen vetureita, jotka omalla innokkuudellaan ja rutiinien karttamisella saattavat lietsoa myös vastarintaa. Tämä johtuu siitä, että kaikki eivät ole kykeneviä muutokseen. Esimiehen on osattava hillitä tällaisen työntekijän vauhtia. Samalla häntä on tuettava ja annettava riittävästi tilaa, jotta hän ei turhautuisi muutoksen hitauteen. Seurailijat ovat puolestaan suurin ryhmä. He ovat hiljaisia ja käyttävät paljon maalaisjärkeä muutosta arvioidessaan. He vaativat esimieheltään kannustusta ja innostusta muutokseen. Lisäksi he odottavat selkeitä arkisia näyttöjä muutoksen vaikutuksista ja hyödyistä. On tarpeen pitää huolta siitä, että seurailijat eivät jää aktiivisimpien jalkoihin. Epäilijät omaavat pitkän kokemuksen ja tunnustetun aseman. He ovat lojaaleja työnantajaansa kohtaan. Monesti he kokevat, että muutos ei ole sopusoinnussa vanhan arvopohjan kanssa. Epäilijät ovat usein passiivisia ja pessimistejä muutoksen suhteen. (Mattila. 2008).

Vielä jyrkempää mielipidettä ylläpitävät oppositiota edustavat. He eivät ole aloitteellisia ja heillä on taipumusta haitantekoon tai kielteiseen viestintään. Oppositiossa oleva pyrkii olemaan mielipidevaikuttaja. Tällaisten työntekijöiden kohdal-

la esimiehen on opittava puuttumaan toimintaan. Hänen on saatava haitanteon sijaan ohjattua työntekijät tehokkaaseen työhön ja unohtamaan haitanteko. Viimeinen ihmistyyppi on opportunisti. Tämä ryhmä on harvinaisin ja heillä on korkea omaksumiskyky ja tehokkuus. He eivät piittaa säännöistä ja keskittyvät pääosin oman edun tavoitteluun. Esimiehen tehtäväksi jää työskentelyn ohjaaminen oikeaan suuntaan. Heidän tehtävänä on puhalttaa peli poikki, jotta ilmapiiri ei menisi pilalle. Kaiken kaikkiaan muutoksen onnistumiseksi tarvitaan perusteellista suunnittelua, tiedonjakoa, jatkuvaa kärsivällistä tukea ja asioiden perustelua. Esimiehen on keskusteltava, rohkaistava, tuettava, koulutettava, siedettävä ja alennettava ihmisten painetta ja suoritettava seuranta muutostilanteessa. (Mattila. 2008).

Oman tiimin ja yksilön myönteisen suhtautumisen lisäksi muutos edellyttää työympäristöltä ja muulta organisaatiolta vahvaa tukea. Organisaation näkökulmasta on asennoiduttava siten, että jokaiselle työntekijälle muodostetaan selkeä kuva oman työn ja organisaation uudistumisen tarpeesta, taustoista ja tavoitellusta suunnasta. Suhteet esimieheen ja työtovereihin pidetään päivittäisessä työskentelyssä läheisinä, kannustavina ja avoimina. Työntekijälle taataan sopiva työkuorma ja työn osien yhdensuuntaiset tavoitteet. Tehtävät pidetään siten, että niiden tärkeysjärjestys on tiedossa. (Mattila. 2008).

Työntekijän osaamista ja intoa oppia uutta ylläpidetään. Esimies tukee häntä muutoksen yhteydessä. Toisaalta edellytykset työyhteisön valmiuteen muutoksessa on oltava kunnossa. Edellisten kohtien lisäksi on varmistettava henkilöstön ja taloudellisten panosten riittävyys. Merkittävässä asemassa on nopea ja selkeä päätöksentekokulttuuri. On varmistettava myös riittävä tiedotus ja johdon sekä esimiesten tuki muutostilanteessa. Muutoksen laajuus pitää olla selvästi rajattu ja hallittu, kuten myös kesto ja vaiheistus. Muutoksen vaikutus pitää olla selvitetty ja tiedossa liittyen

- työprosesseihin ja -menetelmiin
- työvälineisiin

- tietojärjestelmiin
- työnkuviin ja osaamisvaatimuksiin
- organisaatorakenteisiin
- vastuusuhteisiin
- palkitsemisjärjestelmiin ja -etuisuuksiin.

Muutosten hyödyt pitää tunnistaa. Lisäksi odotettu vaikutus henkilöstön määrään ja asemaan pitää olla tunnistettu, sekä yhdenmukaisuus ja tasapuolisuus eri osissa työyhteisöä. (Mattila. 2008).

Muutos on läsnä tiimeissä ja organisaatioissa jatkuvasti. Organisaation on kyettävä uudistumaan. Tämä kyky saadaan organisaation sisältä ja organisaatioiden on osattava verkostoitua. Tietojohtaminen on muutoksen eräs muoto. Sillä tähdätään siihen, että organisaatiossa kyetään käsittelemään inhimillistä pääomaa. Sitä kautta tuotetaan lisäarvoa muuttuen tietopääomaksi ja kilpailukyvyksi. Inhimillinen pääoma koostuu henkilöstön peruskoulutuksesta, muista tiedoista ja taidoista, sekä luonteenpiirteistä. Perusajatuksena on, että esimiehet ovat mahdollistajia. He luovat ja ylläpitävät toimintaympäristöä, jossa työntekijät voivat kehittää osaamistaan ja suunnata sen tiimin uudistamiseen. Käytännössä johtamista ei lopeteta, vaan painotusta muutetaan. Tietojohtamista ei voi käsitellä projektina, vaan jatkuvana muutosprosessina. Siinä innovaatiovoimaa lisätään jatkuvasti ja kaikilla tasoilla. Käytännössä tämä tarkoittaa myös sitä, että jokaisen tiimin jäsenen on otettava vastaavasti vastuuta omasta uudistumisestaan ja kehittymisestä. Kaikkea ei voi odottaa suoraa organisaatiolta. (Grönroos. 2006. 221 - 232).

2.2 Osaamisen johtamisprosessi

Oppiminen ja osaamisen kehittäminen edellyttävät valmiutta ottaa muutos vastaan ja oppimistapahtuman ymmärtämistä. Oppimisprosessin aikana yksilö vie itsessään läpi muutosprosessia hyödyntäen prosessin aikana hankkimiaan uu-

sia tietoja, taitoja, asenteita, kokemusta ja kontakteja. Oppimistyytlejä on erilaisia ja näiden ymmärtäminen yksilön osaamisen kehittämisessä on erittäin merkityksellistä. Aktiiviset osallistujat ovat ennakkoluulottomia, kokeilevia ja avoimia oppijia, jotka innostuvat helposti uudesta. Toisaalta aktiiviset osallistuvat saattavat kyllästyä helposti ja heidän ajatuksensa oppimisprosessissa toimivat helposti itsensä ympärillä. Vastapainoisesti harkitsevat tarkkailijat pysyvät taka-alalla, pohtivat ja havainnoivat tapahtumia. He keräävät tietoa, kuuntelevat ja seuraavat toimintaa sivusta. Käytännön toteuttajat ovat innokkaita kokeilemaan uutta ja etsivät ideoita. Vastaavasti loogiset ajattelijat ovat perfektionisteja, jotka analysoivat uutta tietoa. He pyrkivät maksimoimaan varmuuden, sekä soveltavat havaintojen perusteella tietoja keskenään. Oppimisen kannalta ideaalisimmat oppijat ovat sopivassa määrin jokaista oppijatyyppeä. He kykenevät hankkimaan tietoa laajasti ja soveltamaan sitä ymmärryksen ja toiminnan kautta. (Sydänmaalakka. 2007. 33 - 58).

Edellytyksenä oppimiselle on, että yksilö on halukas ja motivoitunut oppimaan uutta. Hänellä on oltava käytettävissään riittävästi aikaa tiedon hankintaan. Suuri osa tiedosta on dokumentoimatonta ja toisaalta tietoa on tarjolla liikaa. Yksilöllä on oltava kyky poimia saatavilla olevasta tiedosta oleellinen. Opittava asia on myös ymmärrettävä ja löydettävä aikaa, sekä tilaa uuden asian kokeilemiseen. Myös tiimin ja organisaation on oltava vastaanottavia ja halukkaita tukemaan uuden tiedon oppimista. Tiimissä on oltava selvillä yhteinen tavoite ja hyvät vuorovaikutustaidot, sekä kykyä jakaa tietoa tiimin jäsenten kesken. Organisaation kannalta esteenä saattaa olla keskustelutaitojen tai koordinoinnin puute. Myös todellisuuden väärintulkinta ja yksipuoliset näkökulmat asioiden suhteen haittaavat oppimista. Oppimistaidon kehittämiseksi on kyettävä systeemiajatteluun, jossa kokonaisuuksien ja asioiden välisiä vuorovaikutussuhteita kyetään näkemään. Lisäksi on kyettävä huomioimaan se, että yhden asian muuttuminen vaikuttaa usein myös muihin asioihin. Sisäiset mallit eli tapa nähdä maailmaa ja toimia eri tilanteissa on hahmotettava myös. Strateginen oppiminen on hallittava eli johdon ja henkilöstön kyky tulkita maailmaa. On osattava kyseenalaistaa

omat ajatukset ja hyödynnettävä opittua toteutustapoihin. (Sydänmaalakka. 2007. 33 - 58).

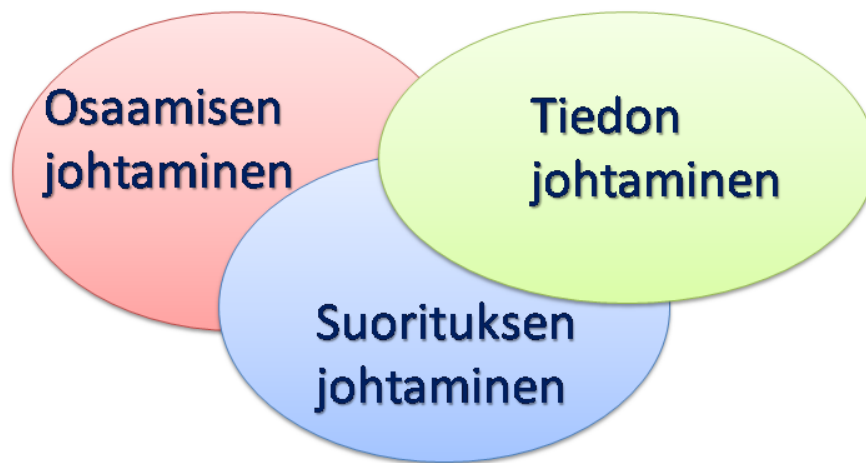
Oppivalla organisaatiolla on kyky sopeutua, muuttua ja uudistua ympäristön muuttuessa. Tulevaisuuden ihanneorganisaatio on älykäs organisaatio, joka osaa nähdä muutostarpeet hyvin aikaisin. Organisaatio oppii nopeasti ja vieden käytännön työhön oppimansa uudistuu jatkuvasti ja säilyttää kilpailukykyä. Älykkäässä organisaatiossa kannustetaan niin yksilöitä kuin tiimejä jatkuvaan oppimiseen. Sitä tuetaan selkeällä visiolla ja arvoilla, jotka ohjaavat toimintaa kohti kestävästä kehitystä selkeillä prosesseilla, suorituksen, tiedon ja osaamisen johtamisella. Organisaatio huolehtii henkilöstön osaamisesta ja sitouttavat työntekijät panostamalla heidän hyvinvointiinsa muun muassa selkeillä palautejärjestelmillä. Tiimit toimivat lähellä asiakasta. Ensisijainen tavoite on asiakastyytyväisyyden ylläpito ja tehokas henkilöstö- ja tiimijohtaminen. Organisaation tukena on oltava ajanmukainen ja nopean uudistumiskyvyn omaava teknologia, joka tukee työn suorittamista. (Sydänmaalakka. 2007. 218 - 261).

Lähitulevaisuudessa organisaatioiden tulisi keskittyä johtamisessa visiojohtamiseen, strategiajohtamiseen, suorituksen johtamiseen ja itsensä johtamiseen. Visiojohtamisella luodaan tarkoitus toiminnalle ja strategisella johtamisella suunta toiminnalle. Suorituksen johtamisella ohjataan päivittäistä toimintaa ja itsensä johtamisella varmistetaan yksilön voimavarat työn suorittamiseksi. Johtamismallit huomioiden on älykkään organisaation kehittymisen kannalta oleellista kyetä limittämään osaamisen johtaminen, tiedon johtaminen ja suorituksen johtaminen keskenään toimivaksi kokonaisuudeksi. Näillä keinoin tuetaan sekä organisaation, tiimin, että yksilön kehittymistä. (Sydänmaalakka. 2007. 218 - 261).

Osaamisen kehittäminen perustuu yrityksen visioon ja strategiaan. 1980-luvulla strategia painottui kustannusjohtamiseen. 90-luvulla painotus siirtyi jatkuvaan toiminnan kehittämiseen ja tehostamiseen hyödyntäen Lean-tyyppistä johtamista, prosessijohtamista ja laatujohtamista. Lean-johtamisella tarkoitetaan keskittymistä asiakkaalle lisäarvoa tuottavaan toimintaan, jossa pyritään mahdolli-

simman suoraviivaisiin toimintaketjuihin. Nyt 2010-luvulla johdon tärkeimpänä tehtävänä on hankkia, ylläpitää ja kehittää yrityksen osaamista eri osa-alueilla (Kuva 5). (Sydänmaalakka. 2007. 132 - 163).

Organisaation oppiminen > Ympäristön muutos



Kuva 5 Osaamisen johtamisen osa-alueet

Osaaminen ja tieto koetaan tärkeimmiksi menestystekijöiksi. Kilpailuetu saavutetaan osaamista, prosesseja ja tietojärjestelmiä kehittämällä. Työ tehdään niin yksilön, tiimin kuin koko organisaation tasolla. Kyseessä on osaamisen, tiedon ja suorituksen johtaminen. (Sydänmaalakka. 2007. 132 - 163).

Osaamisen, tiedon ja suorituksen johtaminen

Vision, strategian ja yrityksen muiden tavoitteiden näkökulmasta osaamisen johtaminen edellyttää ydinosamisten kartoittamista. Ydinosamiset ovat osaamisten, teknologioiden ja tietojärjestelmien yhdistelmä. Näiden ydinosamisten hyödyntämisen kautta yritys tuottaa asiakkaalle lisäarvoa. Tällaista osaamista on kilpailijoiden hankala kopioida. Ydinosamiset jaetaan osaamis-

alueisiin ja sitä kautta osaamisiin. Kehittäminen tapahtuu osaamisalueiden ja osaamisten kautta. Tarkastelunäkökulmat ovat yksilön, tiimin ja organisaation kannalta. Kokonaisuudessaan osaamisen johtaminen vaatii onnistuakseen jatkuvaa osaamistarpeiden arviointia ja niiden täyttymistä nykyisellä henkilöstöllä. Jatkuva osaamisen kehittäminen edellyttää, että henkilöstöä koulutetaan tai rekrytoidaan. Tarkoituksena on tällöin täydentää osaamista. Toisaalta on löydettävä kykyä luopua sellaisesta osaamisesta, jota toiminnan onnistuminen ei edellytä. (Sydänmaalakka. 2007. 132 - 163).

Osaamisen johtamisen tukena tarvitaan tiedon johtamista tiimin tasolla. Kyseisellä prosessilla luodaan, hankitaan, varastoidaan, jaetaan ja sovelletaan tietoa. Tiedon luominen on joko yksilöllistä oppimista tai ryhmässä ideointia sekä esimerkiksi työnkiertoa. Tiedon hankinta on esimerkiksi kursseilta, kirjoista ja Internetistä sisäistettyä tietoa. Toisaalta tieto voi olla yrityksessä jo olevaa tietoa, jota ei ole aiemmin tiedostettu. Tieto kerätään ja muutetaan jaettavaan muotoon. Tiedon luonnin ja hankinnan pohjalta voidaan yritykselle laatia selkeät ja parhaat käytännöt sisältävät prosessit. Jatkuva prosessien parantaminen edellyttää uuden tiedon ja olemassa olevan tiedon jatkuvaa kokoamista ja analysointia, sekä oleellisen tiedon hyödyntämistä prosesseissa. Varastoinnin osuus yrityksessä on merkittävä tiedon johtamisen kannalta. Tiedon jakamiseen on oltava selkeät ja helpot kanavat. Teknologian kehittymisen myötä kanavat ovat monipuolistuneet perinteisistä paperidokumenteista tietokantojen kautta esimerkiksi wiki-tyyppisiin Internet-sivustoihin. Tältä osin koko organisaatiolle on laadittava yksiselitteiset ja yhtenäiset tavat tiedon jakamiseen ja kuvaamiseen. Soveltaminen mahdollistaa tiimin kehittymisen tiedon jakamisen jälkeen tiedon hyödyntämisenä yksilön omassa työssä tiimin yhteistä päämäärää kohden. (Sydänmaalakka. 2007. 175 - 199).

Tiimin toiminnan kannalta on oleellista oppia keräämään sellaista tietoa, joka tukee organisaation strategiaa ja tavoitteita. Yksilön kannalta on merkityksellistä omata kyky ja halu oppia uutta, muuttua ja jakaa osaamaansa tiimissä, sekä vastaanottaa ja soveltaa olennaisia asioita tiimin muilta jäseniltä. Uuden tiedon

jatkuva soveltaminen edellyttää lisäksi selkeää yrityskulttuuria, joka tukee tiedon johtamista. Lisäksi on tuettava esimiesten johtamista, vuorovaikutustaitoja, palautteen antamista ja arvojen selkiyttämistä. Tiimissä on luotava sellainen ilma- piiri, jossa jokainen yksilö ymmärtää tiimin tavoitteet ja merkityksen koko orga- nisaation kannalta. Jokaiselle tiimin jäsenelle on määriteltävä järkevä rooli ja vastuut. Lisäksi tiimin eri jäsenten roolien tulisi nivoutua yhteen siten, että tiimin tehtävä tulee täytettyä tiimille asetettujen tavoitteiden mukaisesti. Avoin ja aktii- vinen vuorovaikutus tiimin jäsenten kesken on edellytys tiedon johtamisen on- nistumiseksi. Tämän luomiseksi on tiimin vetäjien huolehdittava yksilöiden tasa- puolisuudesta kehittämisestä, palkitsemisesta ja tiimin jäsenten pienestä vaihtu- vuudesta. Perehdyttäminen tiimin tehtäviin vaatii aikaa. Pitkäkestoisin työ on yhteisen luottamuksen rakentamisessa tiimin jäsenten kesken ja sitä kautta tu- loksen hyödyntämisessä yhteisissä tavoitteissa. (Sydänmaalakka. 2007. 175 - 199).

Suorituksen johtaminen liittyy päivittäiseen työhön, jota tiimeissä tehdään. Ky- seessä on tavoitteiden asettamista yksilötasolla. On määriteltävä yksilön tehtä- vät ja mittarit, joilla arvioidaan työstä suoriutumisen onnistumista. Samalla poh- ditaan keinoja, joilla yksilön osaamista kehitetään. Päivittäisen työn ohjaamisel- la yksilöä opastetaan ja ohjataan suoriutumaan hänelle asetetuista tehtävistä. Esimiehen tehtävänä on myös tehtävien priorisointi ja kokonaisuuden hallinta. Yhteistyö yksilöiden kanssa edellyttää vahvaa vuorovaikutusta esimiehen ja yksilöiden välillä. Lisäksi vaaditaan luottamuksellista suhdetta ja työn suorituk- sen oikeudenmukaista arviointia yksilöstä riippumatta. Yksilön työn tuloksia ar- vioidaan säännöllisesti ja sen pohjalta yksilön osaamista. Työtehtäviä kehitet- tään tukien sekä yksilön että tiimin jatkuvaa kehittymistä. Tavoitteena on myös yhteistyön rakentaminen ja parantaminen esimerkiksi päivittäisestä työtä tuke- villa suunnittelupalavereilla. (Sydänmaalakka. 2007. 107 - 130).

Merkittävässä asemassa on palautteen anto päivittäisen työn osalta. Lisäksi säännöllisin väliajoin kehityskeskustelussa annettu palaute määriteltyjen tavoit- teiden ja arvioiden osalta, sekä käytettävät palkitsemistavat ovat tärkeitä. Yksi-

löt ovat erilaisia tulkitessaan palkitsemistapoja. Näitä ovat kuitenkin peruspalkan ja tulospalkan suuruus, sekä muut palkan omaiset edut. Niitä ovat esimerkiksi puhelinetu, liikunta- tai kulttuurietuus tai lounasetuus. Rahallisten etuuksien vastapainoksi palkitsemistapoina voidaan pitää haastavat ja kehittyvät työtehtävät, sisäinen tai ulkoinen koulutus. Korkeammalla tasolla voidaan kokea palkitsemiskeinoina myös suullisen palautteen antaminen työstä. Toimiva työyhteisö, työympäristö ja siihen liittyvät asiat ovat myös palkitsemiskeinoja. Lisäksi yrityksen imagon kehittyminen ja arvot voivat osaltaan tuoda arvon tunnetta myös yksilölle ja yrityksen menestyksen myötä palkita yksilöä tunnetasolla. (Sydänmaalakka. 2007. 107 - 130).

Esimiehen rooli osaamisen johtamisessa

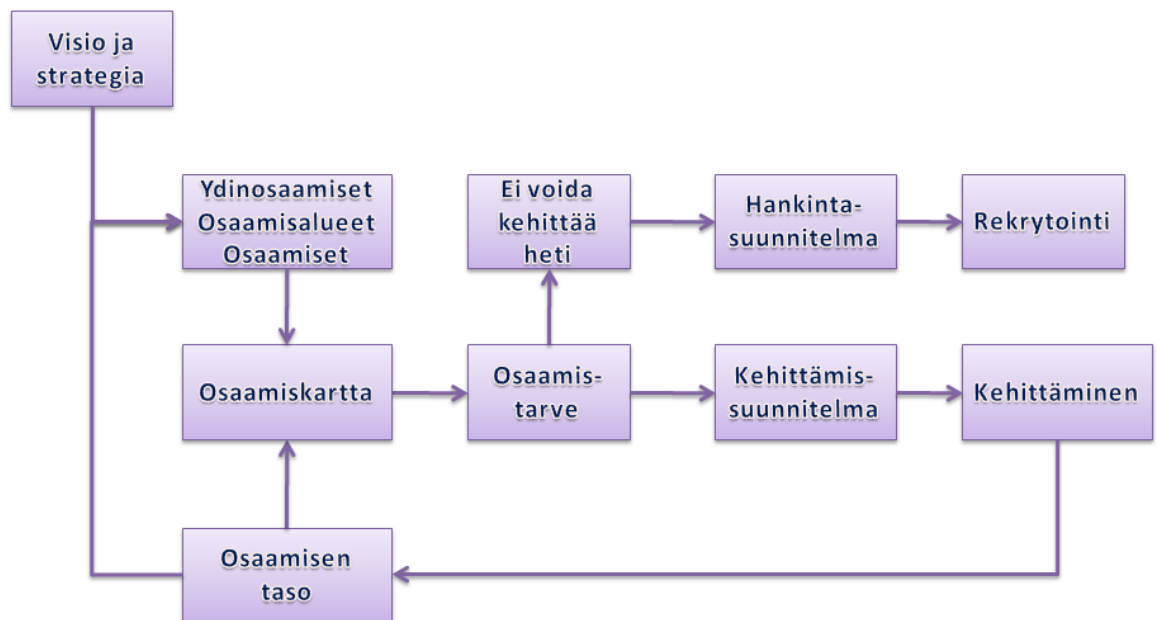
Osaamisen, tiedon ja suorituksen johtaminen edellyttävät esimieheltä kykyä toimia osaamisen johtajana. Esimiehen on oltava esimerkkinä, jolloin hän kehittää omaa ammattitaitoaan. On oltava innostunut työstään ja sitoutunut jatkuviin muutoksiin. Hänen on luotava oppimista edistävä ilmapiiri kehittämällä työyhteisön ilmapiiriä siten, että vuorovaikutustaitojen osaamista korostetaan ja tuetaan. Oppimisprosessien tukeminen edellyttää ryhmän kokonaisosaamisesta huolehtimisesta ja yksilön kehittymisen tukemisesta tiimissä. Tavoitteiden määrittäminen, laadun arviointi vaativat esimieheltä myös vastaavien kanavien luomista ja ylläpitoa. (Sydänmaalakka. 2007. 107 - 130).

Osaamisen johtamista tukeva esimiehen rooli on enemmänkin valmentaja. Esimies pyrkii luomaan sellaiset raamit, joissa työntekijöiden on helppo ja turvallinen tila etsiä ratkaisuja päivittäisen työn suorittamiseen noudattaen yhteisesti sovittuja toimintamalleja. Esimies ei anna valmiita vastauksia, vaan tukee yksilöä löytämään vastaukset kysymyksiin itse esimiehen tukemana tai opastamana. Lisäksi voidaan hyödyntää myös muita tiimin kanavia esimerkiksi tiimin muiden jäsenten osaamista hyödyntäen. Tämän sosiaalisen verkoston analysointi on tarpeen kartoittamaan yksilöiden kanavat yrityksen sisällä ja yrityksen ulkopuolella. Lisäksi on varmistettava, että tiimissä on toimiva varamiesjärjestelmä.

Tiimin toiminta ei voi olla yksittäisen tiimin jäsenen vastuulla, jolloin työn sujuvuus kärsii jäsenen estytyttyä suorittamasta tehtävänsä. (Sydänmaalakka. 2007. 107 - 130).

Osaamisen johtamisprosessi

Osaamisen johtamisen prosessi voidaan esittää seuraavan kuvan mukaisesti (Kuva 6).



Kuva 6 Osaamisen johtamisprosessi (Mayor, 2010).

Prosessi käynnistyy yrityksen vision ja strategian sekä tavoitteiden tarkentamisesta. Näiden pohjalta määritellään ydinosaamiset, joita yrityksessä tarvitaan strategian toteuttamiseksi. Ydinosaamiset pilkotaan osaamisalueiksi ja osaamiksiksi. Määrittelyn yhteydessä arvioidaan millaista henkilöstö-, laite-, tietojärjestelmä- ja teknologiaresursseja tarvitaan tarpeiden täyttämiseksi ja sitä kautta ydinosaamisen kattamiseksi. Osaamisten perusteella voidaan laatia osaamiskartta, johon kootaan yrityksessä tarvittavat osaamiset ja yrityksen käytettävissä

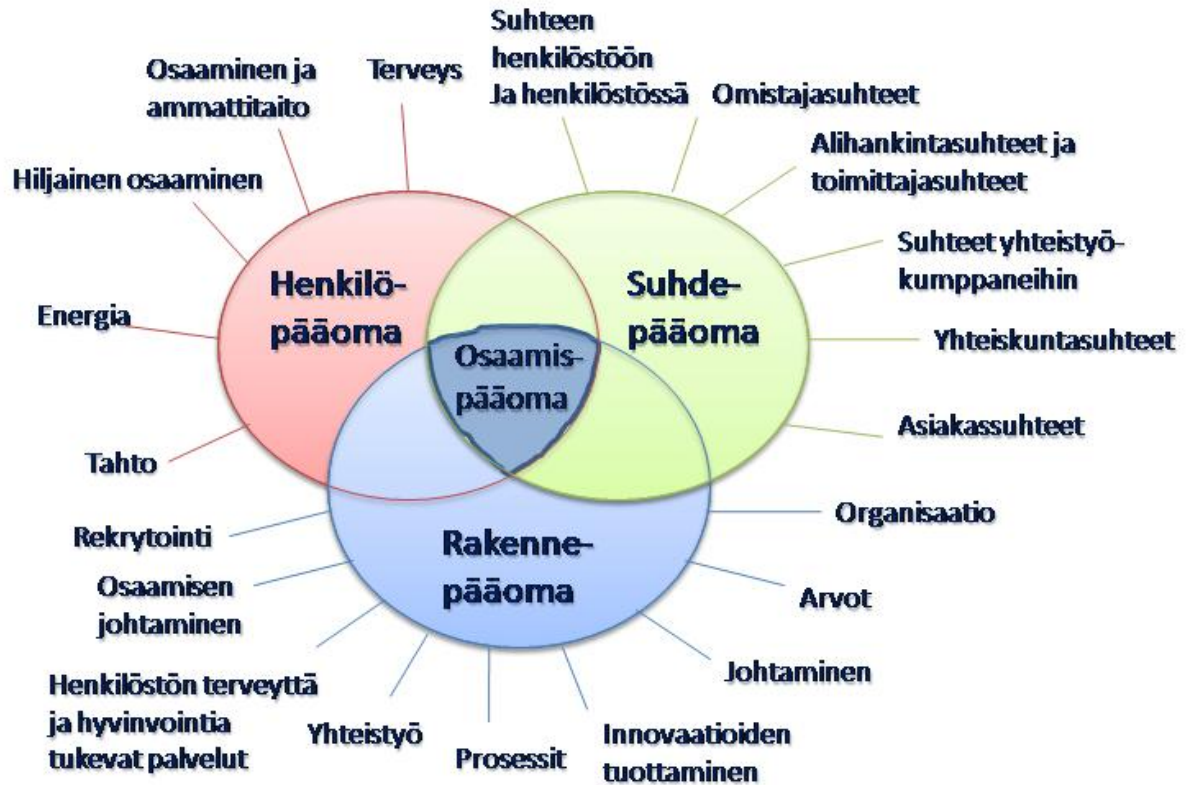
olevaan osaamiseen. Yleensä tämä osuus saadaan kartoitettua henkilöstölle suoritettujen kehityskeskustelujen perusteella tarkistamalla heidän osaamistasonsa. Osaamisille pitää määritellä mittarit, joiden perusteella henkilöstön osaamistaso voidaan arvioida. Lisäksi voidaan määrittää kullekin osaamiselle tarvittava osaamistaso. Osaamisia on yrityksessä useita satoja, joten määrittely on työlästä ja vaatii osaamisten priorisointia. Toiminnan kannalta oleelliset osaamiset ja niiden mittarit on kyettävä ottaa yrityksessä käyttöön. Tarkoitus on, että määritellyt osaamiset seurataan ja arvioidaan jatkuvasti. Osaamisten määrittelyssä voidaan käyttää hyväkseen myös asiakkailta saatua palautetta työstä, Sen perusteella arvioida, onko jotain yrityksellä puutteita esimerkiksi jonkin osaamisen osalta. Vahvistamalla tätä osaamista saadaan lisäarvoa asiakkaalle ja kasvatetaan asiakastyytyvyyttä. (Mayor, 2010).

Osaamiskartan määrittelyn perusteella saadaan analysoitua tarvittavan osaamisen ja käytettävissä olevan osaamisen erotus eli osaamistarve. Tämän analyysin perusteella on tarpeen priorisoida osaamistarpeet siten, että mahdollisimman pian täytettävät osaamiset voidaan erotella toisista osaamisista. Kriittisimpien osaamisten täyttäminen on yrityksessä välttämätöntä tiimin toimivuuden ja yrityksen toimivuuden kannalta. Analyysin perusteella laaditaan kehittämissuunnitelma, jonka mukaan yrityksen henkilöstöä tullaan kouluttamaan joko ulkoisella tai sisäisellä koulutuksella esimerkiksi työssä oppimisella. On mahdollista, että yrityksen omasta henkilöstöstä ei löydy sellaista osaamista, jolla osaamistarve saataisiin katettua nopealla aikataululla. Näissä tapauksissa osaamistarpeen täyttämiseksi laaditaan hankintasuunnitelma. Uusien työntekijöiden rekrytointi suoritetaan siten, että uudet resurssit vastaavat osaamistarpeeseen. Rekrytoinnin kautta osaamisen hankinta mahdollistaa tiedon jakamisen muille yrityksen työntekijöille ja näin saadaan vahvistettua osaamista entistään. (Mayor, 2010).

Oli valintana rekrytointi tai olemassa olevan henkilöstön kehittäminen, suoritetaan henkilöstön osaamistason mittaaminen tavoite- ja kehityskeskusteluissa. Lisäksi tietoa kerätään kehityskeskustelujen välillä päivittäisen työn seurannan

yhteydessä. Osaamisen kehityksen perusteella päivitetään osaamiset osaamiskarttaan, sekä tarkastellaan yrityksen ydinosamisia, osaamisalueita ja osaamisia. Jatkuva kehittyminen edellyttää, että yrityksessä kyetään kilpailukyvyyn säilyttämiseksi arvioimaan yrityksen osaamistarpeita henkilöstöltä, asiakkailta ja muilta sidosryhmiltä saadun palautteen perusteella. Yrityksen on nopealla muutosvalmiudella kyettävä päivittämään osaamiset siten, että kilpailukyky kyetään varmistamaan edelleen. Päivittyneiden osaamisten perusteella yhdistetään henkilöstön kehittyminen ja osaamiset osaamiskarttaan. Tarvittaessa tehdään uudet arviot kehittämis- ja hankintasuunnitelmia varten. Ajoittain saatavan palautteen perusteella voidaan saada kehittämistarvetta myös visioon tai strategiaan. Se puolestaan vaatii tarkempaa tarkastelua yrityksen suuntaviivojen osalta. (Mayor, 2010).

Johtamisen näkökulmasta osaamisen johtaminen on jatkuva kehä, jossa liiketoiminnan tavoitteita pyritään ohjaamaan jatkuvalla johtamisella, kehittämisellä, arvioinnilla ja tarpeiden täyttämällä hankinnan kautta. Näiden pohjalta syntyvät liiketoiminnan tulokset. Onnistunut johtaminen edellyttää niin johdon kuin henkilöstön sitoutumista. Käytetty kieli ja termistö tulee olla yksiselitteistä ja osaamisen painopiste tulisi siirtää yksilöstä tiimin osaamisen ohjaamiseen. Toiminnan tulee olla myös systemaattista ja kokonaisuuden hallinta edellyttää toimivia tietojärjestelmiä. Osaamisprosessia läpikäydessä on kyettävä arvioimaan lyhyen aikavälin kehittämistarpeet ja -mahdollisuudet. Ne on eroteltava pidemmän aikavälin tulevaisuuden tavoitteista. Näiden arviointi pitäisi kyetä tekemään jo useita vuosia etukäteen. Yrityksen johdolla tulee olla visioita ja strategiaa laadittaessa kykyä ennakoita tulevaa mahdollisimman hyvin. Mitä parempi ennakointikyky johdolla on, sitä varmemmin yritys säilyy kilpailukykyisenä ja pystyy vaurutamaan tuleviin muutoksiin markkinoilla. Onnistuessaan tavoitteissaan yritys kykenee osaamisen johtamisella saavuttamaan tarvittavan osaamispääoman niin henkilö-, rakenne- kuin suhdepääoman suhteen, seuraavan kuvan mukaisesti (Kuva 7). (Mayor, 2010).



Kuva 7 Osaamispääoma (Mayor, 2010).

Osaamisen tasapainoisuus mahdollistaa osaamispääoman, jolla yritykselle saavutetaan korvaamaton kilpailukyky. Yhteiset näkemykset edellyttävät, että jokainen toimija yrityksessä kykenee täyttämään samat vaatimukset. Henkilöstöltä ja johdolta tarvitaan toimialalla tarvittavaa ammattiosaamista, joka on monipuolista. Lisäksi tarvitaan laatuosaamista, nopeaa ongelmien ratkaisukykyä, sekä hyvä kyky nähdä asioita asiakkaiden näkökulmasta. Työnantajan on kyettävä tarjoamaan työntekijöille motivoivat työtehtävät siten, että työntekijän osaaminen vastaa tehtäviä. Lisäksi on luotava sellaiset työtavat, oppimisympäristö ja kommunikointikeinot. Työyhteisön ja ulkoisten verkostojen ilmapiiri ja toiminta on saatava riittävän korkeatasoiseksi, laaja-alaisiksi ja yrityksen toimintaa hyödyttäväksi. Toiminnalla voidaan varmistaa, että hiljaista tietoa ei yrityksestä katoa. Toisaalta osaamisen jakaminen muodostuisi osaksi jokaisen työntekijän päivittäistä työtä. (Mayor, 2010).

2.3 Ohjaaminen testauksen näkökulmasta

Testaustiimin ohjaamiseen ja johtamiseen vaikuttavat myös testauksen standardit ja yleisesti hyväksi havaitut käytänteet. Jotta testausta voitaisiin ohjata, on ymmärrettävä, mitä testaus on ja mitä siltä odotetaan. Sen sijaan, että testausta pidettäisiin projektin yhtenä vaiheena, pitäisi sen olla enemmänkin näkökulma. Tämä näkökulma tulisi huomioida projektin jokaisessa vaiheessa ja näkökulman kehittämistä voidaan tehdä kaikista kohdin työskentelyä. Oleellista ohjaamisessa ja testauksen kehittämisessä on se, että työssä säilytetään käytännönläheisyys. (Kivinen, 2006).

Testauksen tarkoitus

Testauksen tarkoitus on moninainen, mutta testattavasta kohteesta riippumatta tarve testaukselle on ilmeinen. Ohjelmiston viat voivat vaikuttaa heikentävästi yrityksen kustannuksiin, ajankäyttöön ja yrityksen imagoon. Pahimmillaan viat voivat aiheuttaa onnettomuuden tai jopa ihmisen kuoleman. Testauksella voidaan vähentää ohjelmiston vikoja siten, että viat tai niiden aiheuttamat häiriöt havainnoidaan ajoissa ennen ohjelmistojen ottamista käyttöön asiakkaiden operatiivisessa toiminnassa. Häiriöiden havainnoinnin lisäksi testauksella voidaan ennaltaehkäistä vikojen syntymistä suorittamalla staattista testausta hyvissä ajoin ennen ohjelmiston koodaamista. Tällä tavoin voidaan huomioida riittävän ajoissa mahdolliset riskikohdat, joiden yhteydessä vikoja saattaa generoitua. Testauksella pyritään myös keräämään tietoa ohjelmiston tilasta, saavuttaa luottamusta ohjelmiston toimivuudesta vaatimusten mukaan. (Istqb Foundation, 2011).

Vaikka testauksella voidaan löytää vikoja ohjelmistosta, testauksella ei voida todentaa, että ohjelmistossa ei olisi lainkaan vikoja. Vaikka vikoja ei löydetä testauksen avulla, saattaa jokin vika jäädä testauksen yhteydessä havaitsematta. Tämä johtuu siitä, että testauksella ei ole mahdollista varmistaa kaikkia mahdollisia yhdistelmiä käyttä ohjelmistoa. Toisaalta asiakasympäristö

poikkeaa useasti testausympäristöstä, jossa testaus pääasiallisesti suoritetaan. Ohjelmistoviat saattavat johtua inhimillisestä virheestä määrittelyssä, suunnittelussa tai toteutuksessa. Toisaalta tuotteelle asetetut vaatimukset saattavat olla puutteelliset. Koodaajan taidot eivät välttämättä vastaa toteutettavan ominaisuuden koodaamiseen tarvittavia taitoja. Teknologiat kehittyvät ja samoin järjestelmien yhteensovittaminen ei ole myöskään helppoa. Ympäristön ja tiukoissa aikatauluissa kilpailevat toimijat kohtaavat usein painetta, joka johtaa testauksen ja kehityksen priorisointiin kilpailutilanteen ollessa kova. (Istqb Foundation, 2011).

Testausprosessin voidaankin sanoa koostuvan ohjelmistotuotteen kaikista elinkaaren toiminnoista, niin staattisista kuin dynaamisista. Prosessissa huomioidaan ohjelmistotuotteen ja siihen liittyvien tuotteiden suunnittelun, valmistelun ja toteutus. Testauksella varmistetaan, että edellä mainitut toiminnot täyttävät niille asetetut vaatimukset. Lisäksi pyritään varmistamaan, että ohjelmisto soveltuu käyttötarkoitukseensa. Ennen kaikkea tavoitteena on löytää vikoja ja niistä aiheutuvia häiriöitä ohjelmiston toiminnassa. (Istqb Intermediate, 2011).

Testaukselle on määritelty seitsemän periaatetta. Ensimmäisen periaatteen mukaan testauksella voidaan osoittaa vikojen olemassa olo, mutta ei sitä, että vikoja ei ole. Testauksen tehtävänä ei ole osoittaa ohjelmiston virheettömyyttä. Täydellinen testaus ei ole yleisesti mahdollista. Testaaminen täydellisesti vaatisi kaikkien erilaisten syötteiden ja alkutilanteiden huomiointin, sekä kaikkien mahdollisten toimintopolkujen suorittaminen. Täydellisen testauksen korvaa testauksen kohdistaminen priorisoimalla ja riskianalyysiä hyödyntäen riskialttiimpiin toimintoihin. Kolmas periaate koskee aikaista testausta. Testaustehtävien suorittaminen tulisi aloittaa mahdollisimman ajoissa tuotteen elinkaarta ajatellen. Samalla on tarpeen huomioida tuotteelle asetetut tavoitteet. On yleistä, että viat kasaantuvat yleensä tiettyihin ominaisuuksiin. Näin ollen testauksen yhteydessä tulisi ennalta kyetä arvioimaan kriittisimmät toiminnot ja seurata vikojen esiintymistiheyttä eri toiminnoissa. Testauksen priorisoinnissa pitäisi huomioida nämä toiminnot. Toisaalta on ymmärrettävä, että kertaalleen laadittu-

jen testien toistaminen kerta toisensa jälkeen uudelleen vähentää vikojen löytämistä sovelluksesta. Tämän vuoksi testitapauksia tulisi muokata säännöllisesti, laatia uusia ja etsiä vikaherkempiä yhdistelmiä toimintojen käytön suhteen. Testitapauksia ei siis laadita kerralla kuntoon, vaan niiden kehittäminen projektin edetessä on tarpeen. Lisäksi on huomioitava, että testausta tulisi suorittaa eri tilanteissa eri tavalla. Testausta suunniteltaessa on huomioitava ohjelmistos- ta ja ympäristöstä riippuvat testaustavat, kuten käytettävyys, suorituskyky tai tietoturva. Kaikissa ympäristöissä ei toimi samat testaustavat. Erittäin tärkeä periaate on huomioida, että ohjelmisto saattaa toimia hyvin, eikä vikoja löydy, mutta tuotteen käyttö asiakkaan ympäristössä ei ole mahdollista. Testauksella on kyettävä varmistamaan, että valittu ratkaisumalli soveltuu käyttöön. Tuotteen on täytettävä asiakkaan lisäksi myös varsinaisten käyttäjien tarpeet ja odotukset tuotteelta. (Istqb Foundation & Intermediate, 2011).

Testausta on eritasoista. Suoritetaan yksikkö-, integrointi-, järjestelmä- ja hyväksyntätestausta. Yleensä yksikkö- ja monesti myös integrointitestauksen suorittaa kehittäjä itse tai toinen kehittäjä samasta tiimistä. Hyväksyntätestaus suoritetaan asiakkaan toimesta tuotteen lopputoimituksen yhteydessä. Keskustelua käydäänkin paljon siitä, että kenen tulisi suorittaa järjestelmättestaus. Löytyy organisaatioita, joissa kehittäjät testaavat omat tuotoksensa tai toinen kehittäjä testaa tuotoksen. On myös organisaatioita, joissa projektiryhmään kuuluu kiinteästi testaaja. Toisaalta löytyy itsenäisiä testaustiimejä tai jopa yrityksen ulkopuolisten testauspalvelujen käyttöä. On tärkeää osata huomioida riippumattomuuden merkitys testaustyön onnistumiselle. Riippumattomuuden kasvattaminen tehostaa vikojen ja häiriöiden löytämistä. Testaajien kriittinen ja käyttäjälähtöinen asennoituminen testattavaan kohteeseen poikkeaa merkittävästi kehittäjän suhtautumisesta omaan koodiinsa. (Istqb Foundation, 2011).

Häiriöiden etsiminen edellyttää uteliasta ja kriittistä tarkastelukulmaa. Lisäksi on omattava ammattimainen pessimismi tuotetta kohtaan, oltava hyvä kyky nähdä yksityiskohtia sekä loistava keskusteluyhteys kehittäjien kanssa. Myös vahva testauskokemus auttaa arvaamaan mahdollisia vikakohtia ohjelmistossa. Riip-

pumattomuutta voidaan käyttää kaikilla testauksen tasoilla suunnittelemaan ja suorittamaan testit. Riippumattomuuden etuina on se, että testaajat näkevät erilaisia vikoja kuin kehittäjät. He voivat myös varmistaa olettamuksia, joita käyttäjät tulevat tekemään ohjeiden tai vain ohjelmistoa käyttäen. Haittana voi olla, että testaajat eivät tunne ohjelmistoa välttämättä yhtä hyvin kuin kehittäjät. Tällöin mahdollisten vikakohtien arviointi ei onnistu yhtä hyvin kuin kehittäjiltä. Toisaalta itsenäiset testaajat saattavat olla pullonkaula, jos heidät otetaan mukaan projektiin vasta viimeisessä vaiheessa. Tiiviisti projektiryhmässä toimiva testaja lieventää huomaamattaan helposti tarkastelutapaansa käydessään muulle tiimille tutuksi kaveriksi. Kriittinen suhtautuminen edellyttää tietynlaisen etäisyyden säilyttämistä oikean tarkastelutavan säilyttämiseksi. Hyvän testaustyön onnistumisen perusta on, että testausta ei koeta uhkana kehitystyölle. Testausta suoritetaan sekä kehittäjien, että itsenäisten testaajien toimesta. (Istqb Foundation & Intermediate, 2011).

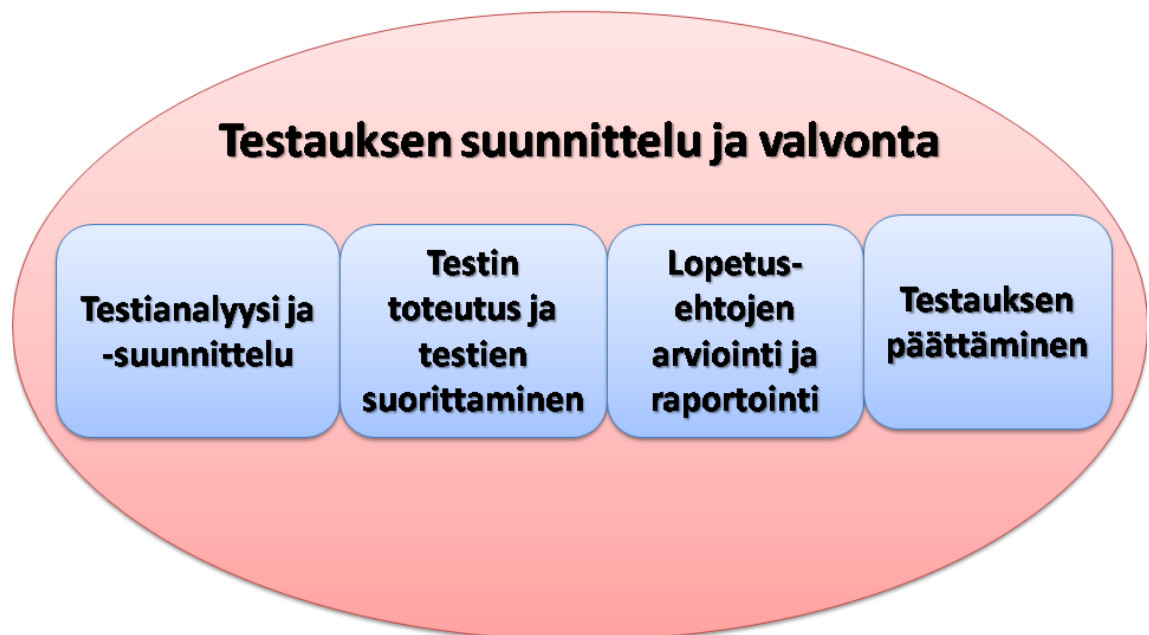
Testaajien palaute voidaan usein kokea kritiikkinä tuotteelle tai jopa kehittäjälle itselleen. Tämän vuoksi on äärimmäisen tärkeää, että testaajat ja testauspäällikkö omaavat hyvät ihmissuhdetaidot. Vioista ja sovelluksen riskeistä on kyettävä puhua avoimesti projektiryhmän kanssa. Lisäksi yhteistyö edellyttää, että projektiryhmä asennoituu avoimesti vastaanottamaan ilmoitukset vioista ja kehitysehdotuksista. Vikojen löytymisen ja kehitysehdotusten antamisen tarkoituksena on säästää aikaa, rahaa ja pienentää riskien realisoitumisen todennäköisyyttä. (Istqb Foundation, 2011).

Testausprosessi ja testaustavat

Ohjelmistokehitysprojekteja viedään läpi noudattaen erilaisia menetelmiä. Perinteisin on vesiputousmalli, jossa kehitys etenee kokonaisuutena kohti toimitusta. Prosessi alkaa määrittelyvaiheella, jonka jälkeen siirrytään suunnitteluvaiheesta toteutuksen kautta testausvaiheeseen. Prosessimalli on usein hankala testauksen kannalta, koska testaus on läsnä vain viimeisessä testausvaiheessa. Menetelmää voidaan kuitenkin pehmentää sillä, että staattista testausta suoritetaan

määrittely ja suunnitteluvaiheessa. Näin tuetaan vikojen aikaista havaitsemista. Useissa yrityksissä hyödynnetään iteratiivis-inkrementaalista kehitystä. Tällöin tuotteen vaatimusmäärittely, suunnittelu, toteutus ja testaus muodostavat prosessin, jota suoritetaan lyhyinä kehityskierrosten sarjoina. Näistä menetelmistä esimerkkejä ovat RAD eli prototyyppien käyttö nopean sovelluskehityksen apuna, RUP ja ketterät kehitysmallit kuten agile ja XP. Projektin käyttämä malli vaikuttaa myös testauksen etenemiseen ja vaiheistamiseen kohti toimitusta. Oli käytössä oleva menetelmä mikä tahansa tai jokin menetelmien yhdistelmä, on kehityksen osalta tarpeen kuitenkin täyttää aina myös vastaavat testaustehtävät ja niiden tavoitteet. (Istqb Foundation & Intermediate, 2011).

Seuraavassa kuvassa, (Kuva 8), on esitelty testauksen kehitysprosessi riippumatta siitä, mitä menetelmiä projekteissa käytetään. (Istqb Foundation, 2011).



Kuva 8 Testausprosessi

Testauksen suunnittelun ja valvonnan yhteydessä varmistetaan testauksen tavoitteet. Testauskohteet määritellään perustuen käytettävään testausstrategiaan. Suoritettavat testauskohteet arvioidaan siten, että testauskohteet ja tes-

tauksen tavoitteet täytetään. Testauksen suunnittelun yhteydessä arvioidaan riskit. Lisäksi määritellään resurssit, joita testauksen yhteydessä tullaan tarvitsemaan. Resurssit kattavat niin henkilö- kuin laite- ja tietojärjestelmäresurssit. Kokonaisuus koostuu myös testauksen valvonnasta. Tämä pitää sisällään suunnitelman toteutumisen seurannan. Käytännössä todellista etenemistä verrataan asetettuihin tavoitteisiin ja näiden perusteella tehdään johtopäätöksiä testausprosessin etenemisen suhteen. Suunnittelua ja valvontaa tehdään koko prosessin ajan. Tarkoitus on, että kaikki havaitut poikkeavuudet huomioidaan mahdollisimman aikaisessa vaiheessa. Näin varmistetaan, että poikkeamiin reagointi voidaan tehdä ajoissa ja suunnata voimavarat testaustyön tehostamiseen. (Istqb Foundation & Intermediate, 2011).

Testausprosessin vaiheittaisesta etenemisestä ensimmäinen on testianalyysi ja testisuunnittelu. Tämä vaihe koostuu yleisten testikohteiden tarkentamisesta testausilanteiksi ja testitapauksiksi. Testauskohteet ovat komponentteja tai järjestelmiä, jotka on tarkoitus testata. Testausilanteet puolestaan käsitetään asioiksi, jotka voidaan varmistaa yhdellä tai useammalla testitapauksella. Vaiheen yhteydessä koostetaan tarvittava tausta-aineisto ja arvioidaan tausta-aineiston ja testauskohteiden testattavuus. Erittäin merkittävässä asemassa on testausilanteiden tunnistaminen ja testattavien asioiden priorisointi siten, että kaikki oleellinen huomioidaan testauksessa. Samassa yhteydessä tunnistetaan testidata, jota tullaan testauksen yhteydessä tarvitsemaan testitapauksia ja testausilanteita tukemaan. Vaihe ei edellytä kyseisen aineiston luomista, vaan varautumista aineiston hankintaan. Vaiheeseen kuuluu myös testausympäristön valmistelu. (Istqb Foundation & Intermediate, 2011).

Testin toteutus ja testien suorittamisvaihe siirtää suunnittelun asteen pidemmälle. Vaiheessa suunnitellaan testitapaukset ja laaditaan testien suorituksen kuvaukset. Oleellinen osa suunnittelua on testitapausten priorisointi siten, että testauksen yhteydessä voidaan keskittää suurimmat voimavarat kriittisimpiin osaluoihin. Lisäksi varmistetaan, että testausympäristö vastaa odotettua. Vaiheen yhteydessä ajetaan testit ja tuotetaan tulokset dynaamisella testauksella. Tes-

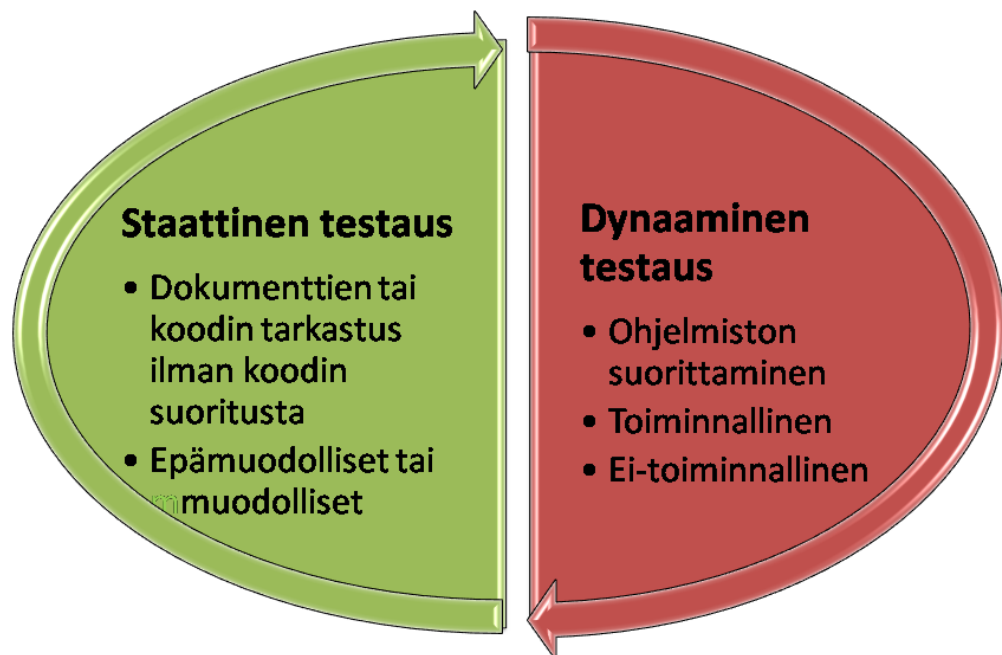
tauksen yhteydessä kirjataan havainnot ohjelmiston puutteellisesta tai poikkeavasta toiminnasta verrattuna testitapauksiin. Lopetusehtojen arviointi ja raportointi on oma kokonaisuutensa. Sen yhteydessä kerätään tiedot testeistä, koetaan kokemukset, tiedot käytetyistä testivälineistä ja tuloksista. Samassa yhteydessä varmistetaan, että kaikki testauksen yhteydessä suoritettavaksi suunnitellut testit on suoritettu ja tulokset näistä on koottu yhteen. Lisäksi varmistetaan, että testauksen yhteydessä tehdyt havainnot on kirjattu samoin kuin kehitysehdotukset. Tuloksista laaditaan yhteenvetoraportti, joka voi olla sähköinen tai paperiversio. (Istqb Foundation & Intermediate, 2011).

Testauksen päättämisen yhteydessä dokumentoidaan järjestelmän hyväksyminen, viimeistellään ja arkistoidaan testiympäristö ja testivälineet. Tarkoitus on, että ympäristö voidaan tarvittaessa koota myöhemmin esimerkiksi asiakkaan tiloissa havaittujen vikojen ilmaantumisen varmistamiseksi testausympäristössä. Tämän perusteella voidaan tehdä vikakorjaukset toimivuuden varmistamiseksi uuden version yhteydessä. Lisäksi todennetaan, että sidosryhmien kesken sovitut yleiset ja erityiset kriteerit täytetään, jotta testaus voidaan päättää. Testauksen edistymistä raportoidaan lopetusehtoja vasten. Niiden perusteella voidaan tehdä lopullinen päätös siitä, onko suoritettu testaus ollut riittävää. Lopetusehtojen tarkastelussa käytetään usein mittareita. Näitä voivat olla muun muassa ajettujen testien määrä kaikista testitapauksista tai vikojen määrä kriittisten toimintojen yhteydessä. Testausprosessi on kuvattu edellä vesiputousmallin mukaisesti, mutta se soveltuu hyvin myös iteratiivisiin projekteihin. Käytännössä suunnittelun ja suorittamisen vaiheita toistetaan toimintokokonaisuus tai ominaisuus kerrallaan. Varsinainen testauksen päättäminen suoritetaan tuotteen lopputoimituksen yhteydessä. (Istqb Foundation & Intermediate, 2011).

Iteratiivinen testaus asettaa haasteet testauksen suunnittelulle, koska iteraatiot ovat lyhyitä. Testauksen tulosten tulisi ohjata ja johtaa kehitystyötä ja iteraation työn pitäisi perustua edellisen iteraation tuloksille. Iteratiivisessa kehityksessä regressiotestauksen osuus tulee merkittäväksi. Iteraatioiden perusteella pitäisi löytyä kykyä löytää esimerkiksi vaikutusanalyysin kautta ne asiat, joihin tuleva

kehitystyö vaikuttaa. Tämän perusteella suoritetaan testauksen suunnittelu ja testattavien tilanteiden valinta. Vaihe on yleensä liian lyhyt ja toisaalta vaatii paljon resursseja. Iteratiivisessa kehityksessä myös ei-toiminnallinen testaus on haasteellista, koska kaikkea kehitystyötä ei ole iteraation päättyessä tehty. (Istqb Intermediate, 2011).

Testausprosessin yhteydessä käytetyt testaustekniikat ovat joko staattisia tai dynaamisia (Kuva 9).



Kuva 9 Testaustekniikkatyypit

Staattisten tekniikoiden yhteydessä ei suoriteta varsinaista ohjelmistoa, mutta dynaamisen testauksen yhteydessä näin tehdään. Staattinen testaus koostuu koodin tai projektidokumentaation manuaalisesta tutkimisesta ja analysoinnista. Tällä menetelmällä voidaan ohjelmistovikojen määrää alentaa jopa kymmenesosaan siitä, mitä se olisi ilman staattisen testauksen suorittamista. Staattista testausta suoritetaan useimmiten ennen dynaamista testausta eli aikaisessa vaiheessa tuotteen elinkaaren alussa. Staattisessa testauksessa kohteena voivat olla vaatimus- ja suunnittelukuvaukset, koodi, testaussuunnitelmat, testita-

paukset ja testien suorituskuvaukset sekä käyttöohjeet. Tavoite on kuitenkin sama kuin dynaamisella testauksella eli vikojen löytäminen. Katselmoinnit voivat olla joko muodollisia, kuten tekninen katselmointi tai tarkastus. Ne voivat olla myös epämuodollisia tai läpikäyntejä riippuen katselmoitavasta dokumentista, katselmoinnin tavoitteesta tai raportointitarpeesta esimerkiksi asiakkaalle. Vastaavasti dynaaminen testaus koostuu useista erilaisista testaustavoista. Niitä ovat toiminnallinen testaus ja ei-toiminnallinen, johon kuuluvat muun muassa

- suorituskyvyn testaus
- kuormitustestaus
- volyymitestaus
- käytettävyys- ja esteettömyystestaus
- skaalautuvuuden testaus
- tietoturvatestaus
- asennus- ja konfigurointitestaus
- regressiotestaus
- uudelleentestaus
- savutestaus
- ylläpitotestaus.

Testaustekniikat edellyttävät testaajilta erityisosaamista suoriutuakseen ohjelmiston vaatimasta testauksesta riittävällä tasolla. (Istqb Foundation, 2011).

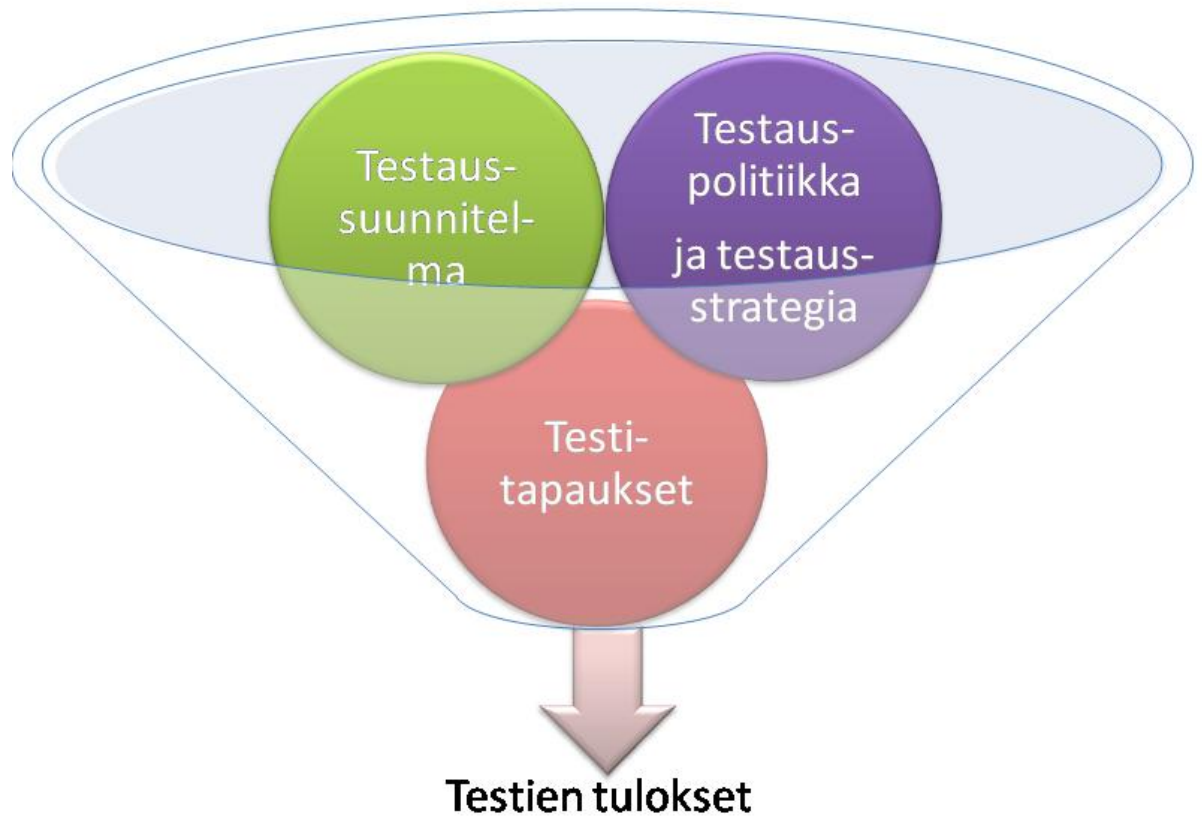
Testauksen hallinta

Hyvä testaus lähtee siitä, että testaaja ymmärtää testauksen kohteen. Ymmärtämisellä on useita tasoja: konseptin, käytön, vaatimusten, toiminnan, havaintojen ja tuotteen liiketoiminnallinen ymmärtäminen. Testaaja tietää, mihin tarkoitukseen tuotetta käytetään, kuka sitä käyttää ja miten sitä tullaan käyttämään. Lisäksi on tarpeen hahmottaa tuotteesta saatava etu, mahdolliset riskikohdat ja mitä tekniikkaa tuotteen kehittämisessä on käytetty. Vaatimusten yhteydessä

kuvataan asiakkaan odotukset tuotteen suhteen. Vaatimukset ovat toiminnallisia ja ei-toiminnallisia. Testausosaamista ja testauksen tarpeita voidaan tarkastella myös organisaation eri näkökulmista. Johdon ja liiketoiminnan kannalta tärkein tarkastelutapa on projektin tuotto ja tuoteriskien kokonaisuuden tila. Tarkastelussa ei keskitytä seuraamaan yksittäisiä laatuominaisuuksia ja tämän vuoksi testaus on prosessi muun tuotekehityksen joukossa. Näissä organisaatioissa testaus toiminnan olemassaolo riippuu täysin kannattavuudesta. On myös niitä yrityksiä, joissa laatu koetaan olevan pohja liiketoiminnan menestykselle. Näissä organisaatioissa testaus on kriittinen toiminto, eikä testauksesta tingitä. Testaus koetaan osaavien ihmisten aivotyöksi. Tuotekehityksen näkökulmasta testauksella todennetaan tuotteita ja on tärkeä tuotekehityksen kaikissa vaiheissa. Heidän näkökulmastaan käytetään tuotteiden vaatimia testaustapoja. (Vuori 2-2011).

Vastaavasti ohjelmistotuotannon näkökulmasta testaus ei tuota mitään, vaan kuluttaa budjettia. Ohjelmistotuotannon toiveena on tämän vuoksi pitää testaus mahdollisimman yksinkertaisena. Testaukseen investoidaan, jos kyseessä on testausprosessia nopeuttava tai tehostava tapa. Tilaaja puolestaan joko luottaa tuotteen olevan testattu toimittajan puolesta tai he osallistuvat esimerkiksi hyväksymistestauksen muodossa testaukseen. Ohjelmistokehittäjät suorittava koodipohjaista testausta ja kokevat usein erillisten testaaajien tekemän työn liioittelevaksi, vikojen etsimiseksi ja negatiiviseksi asenteeltaan. Vastaavasti testausorganisaatio pitää itsestään selvänä testauksen suorittamista ja vikojen etsimistä. Testauksen tarve ja mahdolliset ongelmat ovat selkeitä. Laatuorganisaatio puolestaan kokee testauksen väistämättömäksi, joka halutaan määritellä standardin mukaisesti. Koska näkökulmat poikkeavat näin merkittävästi, on tärkeää ylläpitää aktiivista ja avointa vuorovaikutusta eri osapuolien kesken. (Vuori 2-2011).

Testauksen hallittu ohjaaminen (Kuva 10) edellyttää, että testausta varten on laadittu testauspolitiikka ja testausstrategia, sekä työskentelyohjeet testauksen suorittamiseksi.



Kuva 10 Testauksen hallittu ohjaaminen

Testauspolitiikka vastaa kysymykseen, miksi yrityksessä testataan. Poliittikka määrittelee periaatteet, lähestymistavat ja tavoitteet suoritettavalle testauksella. Testauspolitiikka pitää sisällään testausprosessin, sekä mittarit prosessin tehokkuuden mittaamiseksi ja odotetun laatutason. Lisäksi testauspolitiikkaan eritellään toimintatavat testausprosessin kehittämisen yhteydessä ja mahdolliset kehysmallit, joihin testausprosessin kehittäminen pohjautuu. Testausstrategia puolestaan vastaa kysymykseen, kuinka testaus tehdään. Strategian yhteydessä kuvataan tekniikat, joita testauksessa hyödynnetään. Myös testausresurssien jako, käytettävissä olevat resurssit, aika ja testitettävät on määriteltävä. Lisäksi kuvataan testausympäristö, regressio- ja uudelleentestauksen menetelmät, käytettävät mittarit ja testien automatisointiin liittyvät asiat. Testausstandardien mukaan testaussuunnitelma kokoaa yhteen testauspolitiikan ja testausstrategian kehitettävän tuotteen osalta. Yrityksessä voidaan laatia testaussuunnitelma kuvaamaan yleisesti kaikkien projektien osalta testaus toimintaa. Suun-

nitelman tueksi laaditaan projekteille alitestaussuunnitelmat. Ne kattavat ainoastaan ne osat, joilta kuvatut asiat joko poikkeavat yleisestä suunnitelmasta tai vaihtoehtoisesti täydentävät sitä. (Istqb Intermediate & Advanced, 2011).

Testaussuunnitelmaan kuvataan tavoitteet, kohde, sekä testattavat ja testattamatta jätettävät toiminnot. Lisäksi määritellään testauksen yhteydessä käytettävissä olevat henkilöt, aikataulut, sekä heidän vastuiden jako ja koulutustarpeet. Lähestymistapojen osalta kuvataan testaustekniikat ja kriteerit, joiden perusteella testien tulokset voidaan määritellä (läpi/ei läpi, oletukset). Testausympäristön osalta kuvataan käytettävät laitteet, koneet, ohjelmistot ja niiden versiot sekä koneiden verkkorakenne. Lisäksi määritellään testauksen yhteydessä laadittavat tuotokset, kuten testidokumentaatio. Testaussuunnitelma koostuu myös riskianalyysistä, testauksen tehtävistä ja testauksen yhteydessä tehtyjen havaintojen tunnistamisesta, käsittelystä ja vikakorjausten hyväksyntämenettelyistä. Yleensä kaikkea tietoa ei koota yhteen dokumenttiin, vaan tieto hajautetaan eri dokumenttiin tai tietojärjestelmiin käsittelyn helpottamiseksi. (Istqb Intermediate, 2011).

Testauksen organisointia suunniteltaessa on testauspäällikkö avainasemassa. Hänen tehtäviinsä kuuluvat muun muassa

- testauksen tavoitteiden ja strategian määrittely, sekä suunnittelu
- aikataulut ja tehtävien seuranta
- tehtävien organisointi
- resurssien valinta tehtäviin
- testausstiimin valinta, organisointi ja ohjaaminen.

Testauspäällikön on ymmärrettävä testauksen elinkaari, järjestelmät ja testauksen mittarit. Hänen on kyettävä kuvaamaan testausstrategian näkyvyys testausprosessin eri vaiheissa. Niiden pohjalta hänen tehtävänä on

- arvioida testausta
- aikatauluttaa, valvoa ja ohjata testausprosessin noudattamista
- vikojen hallinnan ja testauksen arvon arviointi
- arviointi liiketoiminnan ja riski-analyysin kannalta.

Lisäksi on kyettävä kehittämään testaustoimintaa ja huomioitava testaustiimin osaamisen kehittyminen. Merkittävässä roolissa on arvioida tiimin jäsenten soveltuvuus motivaation, vuorovaikutuksen sekä osaamisen kautta testaustiimin tarpeisiin. (Istqb Advanced, 2011).

Testauksen hallintaan liittyy tiiviisti pohdinta siitä, kuinka paljon testausta on riittävästi. Yksiselitteistä vastausta ei voida antaa. Keinot arvioida voidaanko testattu tuote julkaista ja toimittaa riippuvat useasta eri seikasta. Testauksen yhteydessä on tehtävä jatkuvaa arviointia tuote- ja projektiriskeistä. Lisäksi on keskusteltava käytettävissä olevasta testausajasta, henkilö- ja laiteresursseista ja kustannuskehityksestä. Toisaalta on pohdittava käytettävissä olevien henkilöiden taitotasoa suhteessa tuotteen kattavan ja hyvän testauksen edellyttämiin testaustekniikoihin. Arviointi ei tarkoita tulosten arviointia, vaan testaustarpeen arviointia projektin alussa ja projektin edetessä. Arvioinnissa hyödynnetään aiemmista testausprojekteissa saatuja kokemuksia ja materiaalitarpeita. Näitä ovat testausautomatisointiin tarvittavat työkalut, kattava ja arvokas testidata-aineisto sekä vaatimuskuvaukset, suunnitelmakuvaukset ja uudelleen käytettävät testimateriaalit. Arviointia voidaan tehdä intuition ja arvauksen perusteella. Käytettävissä on myös erilaisia analyysitapoja, joilla voidaan arvioida testauksen tarvetta tuotteen osalta. (Istqb Advanced, 2011).

Onnistuakseen tehtävässään on testauspäällikön suoritettava jatkuvaa arviointia, valvontaa ja ohjaamista. Esimerkiksi testauksen yhteydessä havaittujen vikojen määriä verrataan valittujen mittareiden mukaisesti sallittujen vikojen määrään. Mittareita voidaan valita eri näkökulmista. Oheisessa taulukossa on esitetty joitakin tarkastelunäkökulmia ja niiden yhteydessä käytettäviä mittareita:

Taulukko 1 Tarkastelunäkökulmia mittareiden määrittämiseksi

Näkökulma	Esimerkki mittarista	Kuvaus
Tehokkuus	Vikojen havaitsemis %	Testaajien löytämät viat suhteessa kaikkiin löydettyihin vikoihin
	Toimituksen jälkeiset viat	Asiakkaan löytämien vikojen määrä toimituksen jälkeen ennalta määriteltynä ajanjaksona
Hyötysuhde	Laadun kustannukset	Kustannukset ilman testausta verrattuna kustannuksiin testauksen kanssa
	Laatukustannusten hyötysuhde	Staattisen testauksen hyödy suhteessa dynaamisen testauksen hyötyyn. Jos luku kasvaa, myös hyöty kasvaa.
	Vikojen havaitsemisen aikaisin	Yksikkö- ja integroititestauksen yhteydessä havaitut puutteet suhteessa hyväksyntätestaukseen
	Testauksen suhteellinen hyöty	Testauksen hyöty suhteessa koko projektiin esim. kustannuksien kautta
	Testauksen hyödyllisyys	Vikojen määrä ja vakavuus verrattuna käytettyyn testausaikaan.
	Automatisoinnin taso	Automaattisten testien osuus ajetuista testeistä
	Testauksen tuottavuus	Testitapausten määrä liittyen testauksen kokonaispanokseen
Läpimenoaika		Testausvaiheen läpimenoaika suhteessa odotettuun aikaan
Ennustettavuus	Suorituksen läpimenoaika	Todellisen ja arvioidun testauksen suorituksen läpimenoajan erotus
	Hyötykustannukset	Todellisten ja arvioitujen kustannusten ero
	Testitapausten ennuste	Toteutuneiden ja arvioitujen testitapausten määrän erotus.

Lisäksi näkökulmana voi olla tuotteen laatu ja kypsyyden. Tuotteen laadun osalta laatuattribuutit ja kattavuusattribuutteja on paljon. Niiden valinta on suoritettava tuote- ja organisaation tavoitteet huomioiden. Kypsyyden osalta mittarit määritellään kypsyydellä, esimerkiksi TPI ja TMMi, mukaisesti. Niitä voivat olla kyky suoriutua annetuista tehtävistä määritellyssä aikataulussa, vaatimusten täyttyminen, työkuorma ja resurssien riittävyys. Lisäksi voidaan tarkastella onnistu-

mista välietapein tai suoriutumista arvioitujen kustannusten suhteen. Myös havaittujen, korjattujen vikojen määrän osalta tai vikojen korjaukseen käytetyn ajan perusteella voidaan arvioida onnistumista. Aikataulutuksen osalta testauspäällikön on tehtävä tiivistä yhteistyötä kehitystiimien kanssa, koska testaus on äärimmäisen riippuvainen kehityksen aikatauluista. Testauksen suunnittelun osalta olisi ideaalisempaa aloittaa suunnittelutyö ajoissa ja koota suunnitelmia paloittain sitä mukaa, kun tietoa saadaan. Käytännössä tämä tarkoittaa sitä, että testauksen ei tulisi odottaa toimintojen vakiintumista. Suunnittelua tulisi rakentaa mahdollisimman aikaisesta materiaalista, jotta testauksen suunnittelu ehdittäisiin saada valmiiksi riittävän ajoissa ennen testausta. (Istqb Advanced, 2011).

Testauksen valvonnan ja ohjaamisen näkökulmasta mittareita voivat olla riskeihin perustuvat mittarit, kuten riskien pienentämiskyky. Lisäksi mittarit voivat pohjautua vikoihin eli raportoitujen vikojen suhde ratkaistujen vikojen määrään, tai vikojen määrä tuoteversiossa. Testitapausten osalta voidaan arvioida suunniteltujen testitapausten suhdetta ajettuihin testeihin tai läpimenneisiin testeihin. Regressiotestien tilaa tai testien suunnitteluun käytettyä aikaa voidaan mitata myös. Testien kattavuuden osalta voidaan arvioida vaatimusten ja suunnitelmien kattavuutta, riskien kattavuutta tai ympäristöjen kattavuutta. Valittujen mittausten antamien tulosten pohjalta voidaan tehdä ohjaavia toimenpiteitä testaus- tiimin toiminnan suhteen. Esimerkkinä voi olla testauksen kohdistaminen tiiviimmin vikaherkkään ominaisuuteen. Jotta työ onnistuisi, tarvitaan riskienhallintaa. Siten tunnistetaan testauksen näkökulmasta tuote- ja projektiriskit, analysoidaan riskien vaikutus, kriittisyys ja todennäköisyys työn etenemiseen ja tuotteen toimittamiseen nähden. Riskein tunnistamiseksi on hyvä käyttää monipuolisia menetelmiä. Näitä ovat esimerkiksi asiantuntijoiden haastattelut, riippumaton arviointi, aiemmista projekteista opitut asiat, kick-off tyyppiset tiimipäivät, aivoriihet tai tarkastuslistat. Riskit voivat liittyä turvallisuuteen, taloudellisuuteen, tietoturvallisuuden, kilpailukykyyn ja toisaalta tekniseen kykyyn. Näitä voivat olla konkreettisesti riskit, jotka liittyvät työkaluihin tai teknologiaan tai tiimien välisiin ristiriitoihin. Riskit voivat liittyä myös sopimuksellisiin rajoitteisiin tai ongelmiin,

käytettävissä olevaan aikaan, resursseihin tai huonoon johtajuuteen. (Istqb Advanced, 2011).

Testauksen osalta yleisimpiä projektiriskejä ovat seikat, jotka liittyvät testausympäristöön ja työkalujen kykyyn suoriutua testauksen tarpeista. Myös testaushenkilöstön taidot ja mahdollisuudet osallistua projektin testaukseen ilman muuta työkuormitusta vaikuttavat osaltaan riskien määrään. Lisäksi liian suuret muutokset projektissa kasvattavat riskejä. Myös standardien, sääntöjen ja tekniikoiden noudattamatta jättäminen kohottaa riskien todennäköisyyttä realisoitua. Tuoteriskit liittyvät puolestaan yleisemmin sopivan testitekniikan valintaan, staattisen testauksen puuttumiseen ja testitapausten katselmoinnin puuttumiseen. Myös testauksen alhainen riippumattomuuden tason sekä regressiotestauksen puuttuminen lisäävät riskejä. Lisäksi testauksen suorittamisen yhteydessä ei ole käytettävissä aina kokeneinta testaustekniikan hallitsevaa testaajaa, vaan jokaisen testaajan oletetaan osaavan testata kaikilla tekniikoilla. Testauksessa pyritään ohjaavin toimin minimoimaan riskien toteutuminen. Hyvin ohjatun testauksen riskianalyysin pohjalta voidaan testausta kohdentaa kriittisiin osa-alueisiin ja huomioida näin testauksen priorisointia suoritettaessa. Vahvalla valvonnalla saavutetaan yhtenäinen ymmärrys testauksen etenemisestä. Valvonnalla saavutetaan myös kyky reagoida mahdollisimman aikaisin ongelma-kohtiin, sekä tehdä ohjaavia liikkeitä mahdollisimman aikaisin varmistukseksi toimitusten onnistumisen. (Istqb Intermediate & Advanced, 2011).

2.4 Kehys- ja kypsyysmallit

Siinä missä testauksella pyritään parantamaan ohjelmistoa, niin ohjelmiston laatuprosesseja käytetään parantamaan ohjelmiston kehitysprosesseja. Tätä tapaa voidaan hyödyntää myös testausprosessin kehittämiseksi. Testaus vie yleensä suuren osan projektin kustannuksista. Tätä osuutta halutaan ymmärrettävästi alentaa testaustuloksen kuitenkin kärsimättä muutoksesta. Testausprosessin kehittämistä varten on luotu erilaisia malleja, kuten

- Test Maturity Model Integration (TMMi)
- Systematic Test and Evaluation Process (STEP)
- Critical Testing Processes (CTP)
- Test Process Improvement (TPI).

Näistä menetelmistä TPI ja TMMi koostuvat mittareista, joita voidaan käyttää esimerkiksi benchmarking-tyyppiseen vertailuun. Mallit soveltuvat erityisesti organisaatioon, jossa testausprosessi on jo määritelty. Vastaavasti CTP ja STEP soveltuvat yrityksiin, joissa testausprosessi pitäisi määrittää tai kustannusten ja riskien huomiointi testausprosessissa on tarpeen. On olemassa myös muita kehitysmalleja. Käytännössä testauksen ammattilaisten tulisi tutustua erilaisiin malleihin ja määritellä sen jälkeen, mikä malleista sopii parhaiten testausorganisaation tilanteeseen. Mallit sallivat yleensä sen, että yritys voi pitää testausprosessin tason haluamallaan tasolla. Myös testausprosessin kehittäminen on tuettu mallien avulla. (Istqb Advanced, 2011).

Testausprosessin kehittäminen ilman malleja

Testausprosessin kehittäminen voidaan suorittaa myös ilman malleja. Tällöin kehittämisen tulisi tehdä huomioimalla erilaisia lähestymistapoja (Kuva 11).



Kuva 11 Lähestymistavat ilman kehittämismallia

Näitä voivat olla henkilöresurssien osaamisen kehittäminen, jolloin keskitytään testaajien, testauspäälliköiden, kehittäjien ja muiden tiimin jäsenten osaamisen ja ymmärtämisen kehittämiseen. Taidot ja osaamisalueet liittyvät yleisesti testauksen ymmärtämiseen, kuten testausperiaatteet, tekniikat ja työkalut. Toinen näkökulma on järjestelmätuntemus, joka pitää sisällään esimerkiksi ohjelmiston vaatimukset ja kehitystyökalut. Toimialatuntemus sisältäen liiketoiminnan kasvattavat tiimin osaamista. Kokemus osallistumisesta useaan eri kehitysprosessin vaiheeseen tukevat osaamista. Näitä vaiheita ovat muun muassa vaatimustenhallinta, suunnittelu, koodaus ja tekninen tuki. Kehittämisen näkökulmasta on huomioitava myös tiimin tarpeet. Uuden henkilön rekrytoinnin tai vanhan työntekijän kouluttamisen yhteydessä on huomioitava tiimin heikkoudet ja pyrittävä

vahvistamaan niitä. Tiimin osaamisen ollessa vahva, kykenee se suoriutumaan useista monimutkaisista projekteista yhtä aikaisesti, vaikka toimintaympäristöt poikkeaisivat toisistaan merkittävästi. Jokaisella tiimin jäsenellä tulisi olla selkeä kuvattu rooli ja rooliin tulisi olla selvä kaikille tiimissä toimiville. Lisäksi niin sanotut pehmeät arvot, kuten vuorovaikutus, työskentelytehokkuus ja raportointitaidot, ovat merkittävässä asemassa. Vuorovaikutus voi olla kolmella eri tasolla, joista ensimmäinen on testausdokumentaatio. Se sisältää testistrategian, testisuunnitelman, testitapaukset, testiraportit ja vikaraportit. Toinen vuorovaikutuksen taso on katselmoinneista saatu palaute. Katselmointia tehdään muun muassa vaatimuksille, toiminnallisille dokumenteille ja käyttötapauksille. Kolmas vuorovaikutuksen taso on kommunikointi kehittäjien, toisten testaajien ja projektijohdon kanssa. (Istqb Advanced & Expert. 2011).

Hyvän testaajan on sanottu olevan analyyttinen, huolellinen, ahkera, käyttäjiä edustava, teknologiaosaaja, testauksen tunteva, aktiivinen, mukautuva, hyvin viestijä, tiimipeluri, oman roolinsa ottava, oma-aloitteinen, ennakoiva ja vastuun ottava toisten auttaja, nopea ja erilaisia toimintatyylejä hallitseva. (Vuori 6-2010).

Toinen lähestymistapa kehittämiseen on työkalujen käyttöönotto. Testauksen hallinnan työkalut helpottavat testauksen suunnittelun ja vikojen hallintaa. Niiden on todettu tuottavan yleensä tehokkuutta, kasvattavan valvontamahdollisuuksia ja tuotosten laatua. Työkalujen käyttöönotto edellyttää kuitenkin sitä, että työkalu hankitaan selkeän tarpeen perusteella. Prosessin parantaminen työkalun avulla ei onnistu, jos vaatimukset työkalulle eivät ole selvillä ennen työkalun hankintaa. Testausympäristössä voidaan ottaa levynkuvia, virtualisoida ympäristöjä tai simuloida ympäristöjä käyttäen erilaisia työkaluja. Toisaalta testauksen automatisointia varten on myös useita erilaisia työkaluja, jotka liittyvät testauskieliin, nauhoita & toista -ohjelmiin, erilaisiin robotteihin tai mallipohjaisiin työkaluihin. (Vuori 6-2010 & Istqb Expert 2011).

Testausta voidaan kehittää myös erilaisista testausnäkökulmista. Voidaan suorittaa staattista tai dynaamista testausta. Staattisen testauksen lisääminen prosessin alkupäähän alentaa yleensä vikojen määrää projektin loppuvaiheesta. Prosessia voidaan vielä läpi vesiputousmallia noudattaen tai käyttäen ketteriä menetelmiä. Toisaalta testaaja voi suorittaa tuotteelle testausta eri tekniikoita käyttäen. Esimerkiksi tietoturvatestaus edellyttää vahvaa osaamista tuotteen käyttötarkoituksen, käyttötapausten ja erilaisten käyttötapojen osalta. Samoin tietovirtojen ja arkkitehtuurin tuntemus on eduksi testauksen suorittamisessa. Vastapainoksi käytettävyytestauksessa ei tarvita osaamista arkkitehtuurin osalta, vaan riittää käyttötarkoituksen, käyttötapausten, sujuvan käytön ja erilaisten käyttötapojen hallitseminen. Testausta voidaan kehittää myös standardien ja säännösten pohjalta. Useilla toimialoilla on ennalta jo asetettu tiettyjä sopimuksellisia velvoitteita, joita toimittajan on noudatettava ohjelmistojen kehittämisen ja testaamisen yhteydessä. Tällöin prosessin kehittämisessä on huomioitava nämä ohjeet.

Testausprosessin kehittäminen edellyttää myös mallittomassa kehittämisessä tiettyjen vaiheiden läpikäynnin. Tätä prosessia voidaan kutsua nimellä IMPROVE tai IDEAL. IMPROVE-prosessi sisältää seuraavat vaiheet:

- **I**nitiate
- **M**easure
- **P**riorize and **P**lan
- **D**efine and **R**edefine
- **O**perate
- **V**alidate
- **E**volve

Aloitusvaiheessa (Initiate) sovitaan testausprosessin kehittämisen kohteet, tavoitteet, näkökulmat ja kattavuus. Tässä vaiheessa valitaan myös prosessimalli, jonka mukaan parannuksia tunnistetaan ja tehdään prosessin aikana. Jos olemassa olevista malleissa ei löydy yrityksen sopivaa mallia, on hyvä laatia yrityk-

sen toiminnalle sopiva malli itsenäisesti. Aloitusvaihetta seuraa mittausvaihe (Measure), jonka yhteydessä laaditaan lista mahdollisista prosessin parannuskohdista. Priorisointi- ja suunnitteluvaiheessa (Priorize and Plan) järjestetään prosessin parannusehdotukset tärkeysjärjestykseen. Tämä vaihe voi perustua riskeihin, mitattavissa oleviin laadullisiin ja määrällisiin etuihin sekä strategiaan näkökulmiin. Kun priorisointi on suoritettu, laaditaan suunnitelma parannuskohdeiden toteutus- ja käyttöönottoajankohdasta. Seuraavassa vaiheessa (Define and Redefine) prosessin parannusehdotusvaatimusten pohjalta tarkastellaan nykyistä prosessia ja etsitään kohtia, joihin vaatimukset edellyttävät muutoksia. Samassa yhteydessä arvioidaan muutosten toimivuutta ja uudelleen arvioidaan prosessin toimivuutta muutosten jälkeen. (Istqb Advanced, 2011).

Kun perusteellinen arviointi ja kohdistaminen on saatu suoritettua, otetaan parannukset käyttöön (Operate). Tämä vaihe sisältää perehdytystä, koulutusta ja valmennusta sekä tarvittaessa pilotointia. Näin voidaan varmistaa muutosten vaikutus kehitykseen. Kun kaikki muutokset on saatu käyttöön, suoritetaan hyötyjen testaus (Validate). Käytännössä edut, joita muutoksesta kehitysprosessin alussa oletettiin saavutettavan, on kyettävä todentamaan kehitysprosessiin valituilla mittareilla. Prosessi päättyy vaiheeseen (Evolve), jossa prosessia valvotaan. Samalla aloitetaan käytettävissä olevasta mallista riippuen seuraavan tason vaatimusten arviointi. Usein kehityskierroksen päättymistä seuraa uusi kierros sisältäen uutta arviointia ja uusia kehityskohteita. Lisäksi tarkastellaan aiemmilla kierroksilla havaittujen puutteiden uudelleenkehittämistä muutoksen jälkeen. Prosessin kehittäminen ei edellytä kehitysmallien käyttöä, vaan kehityskohteita ja prosessin parantamista voidaan tehdä palaverien ja analyttisiä lähestymistapoja käyttäen tiimille itselleen sopivalla tavalla. (Istqb Advanced, 2011).

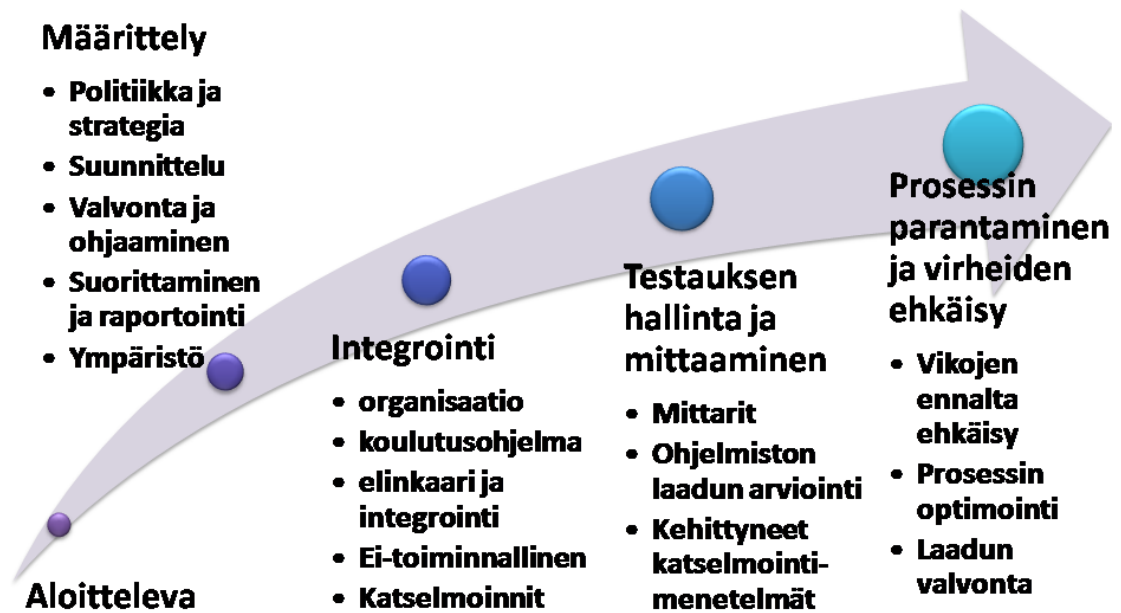
IDEAL-prosessi sisältää vähemmän vaiheita, mutta perusajatus on sama kuin IMPROVE-prosessissa. IDEAL-prosessin vaiheet ovat:

- Initiating
- Diagnosing
- Establishing
- Acting
- Learning

Aloitusvaiheessa (Initiating) määritellään syyt kehittämiselle ja kuvataan kohde ja todetaan ympäristö. Vaihe jatkuu arviointivaiheella (Diagnosing), jonka aikana arvioidaan ja luonnehditaan nykyisiä käytäntöjä. Samalla arvioidaan kehityksen ehdotuksia ja muita saatuja tuloksia. Tämän jälkeen seuraa vaihe (Establish), jonka yhteydessä asetetaan strategia ja priorisoidaan kehittämiskohteet. Samalla luodaan testauksen prosessin ryhmä muutoksen suorittamaan ja suunnitellaan suoritettavat toimenpiteet. Vaihe jatkuu konkreettisella toiminnalla (Acting). Määritellään prosessiin tehtävät muutokset ja mittarit, joilla muutosta tullaan arvioimaan. Tämän jälkeen suunnitellaan ja käynnistetään pilotointi, jonka perusteella kerätään tuloksia. Lisäksi suunnitellaan, suoritetaan ja jäljitetään käyttöönottoa. Viimeinen vaihe (Learning) koostuu dokumentoinnista ja kokemusten analysoinnista, sekä lähestymistavan uudelleen arvioinnista. (Istqb Expert. 2011).

TMMi

Testing Maturity Model Integration (TMMi) mittaa testauksen kypsyyttä organisaation tasolla. Menetelmä koostuu viidestä tasosta, joista tasoilla 2-4 on jokaisella kypsyyteen liittyviä tavoitteita ja alitavoitteita. Lisäksi tasot sisältävät erilaisia aktiviteetteja ja vastuita. Tasot on kuvattu seuraavassa kuvassa (Kuva 12):



Kuva 12 TMMi -mallin tasot (TMMi Foundation, 2010).

Aloitteleva taso kuvaa toimintaa, jossa testausprosessia ei ole muodollisesti kuvattu. Testausta suoritetaan ad-hoc -tyyppisenä koodaamisen jälkeen. Testauksen tavoitteena on saada ymmärrys siitä, että ohjelmisto toimii. Seuraava taso on testauksen määrittelytaso, joka saavutetaan luomalla testauspolitiikka ja tavoitteet. Lisäksi kuvataan testauksen suunnitteluprosessi ja otetaan käyttöön testauksen perustekniikat ja -menetelmät. Kolmas taso on integrointi, jossa testausprosessi on integroitu ohjelmistokehityksen elinkaareen. Testauksen muodolliset toimintatavat, menettelyt ja työskentelyohjeet on kuvattu. On olemassa myös testaus toimintoja, joita voidaan ohjata ja valvoa. Integrointitasolla käsitellään olevan myös koulutusohjelmia testauksen osalta. Neljäs taso liittyy testauksen hallintaan ja mittaamiseen. Käytännössä tason saavuttaminen edellyttää, että testausprosessia mitataan tehokkaasti, hallinnoidaan ja on määritelty ohjelmiston laatuun liittyviä tavoitteita. Organisaatiossa tehdään tarkastuksia ja katselmoiteja. Viimeinen taso on prosessin parantaminen ja virheiden ehkäisy. Testausprosessista tuotettavilla tuloksilla voidaan ehkäistä vikoja, testausprosessia parannetaan ja laatua kontrolloidaan mittarien ja tavoitteiden avulla. (Istqb Advanced, 2011 & Stenberg, 2007).

TMMi-mallin eri tasojen kriteerit on määritelty yksityiskohtaisesti TMMi -mallin kuvauksessa (TMMi Foundation, 2010). Kriteeristö ohjeistaa testausprosessin arvioinnin ja prosessin kehittämisen suhteen tarkasti tavoitteet ja määriteltävät asiat. Tason 2 saavuttamiseksi on testauspolitiikan ja -strategian osalta määriteltävä testauksen tavoitteet ja testauspolitiikka. On myös arvioitava tuotteiden riskit ja näiden kaikkien pohjalta laadittava testausstrategia. Lisäksi on määriteltävä ja otettava käyttöön testauksen suorituksen mittarit. Kaikki nämä on hyväksyttävä yrityksessä ennen käyttöönottoa. Testauksen suunnittelun osalta on suoritettava tuoteriskianalyysi, tunnistettava testausolosuhteet ja ominaisuudet. Niiden pohjalta on määriteltävä testauksen lähestymistavat, aloitus- ja lopetus-kriteerit, sekä keskeytyskriteerit. Testauksen arvioinnin osalta on määriteltävä testauksen elinkaari ja arvioitava testauksen hyödyt ja kustannukset. Testaus-suunnitelman osalta on arvioitava testausaikataulu, suunniteltava testaus-resurssit ja vuorovaikutuskeinot sidosryhmien välillä sekä tunnistettava testaus-projektin riskit. Lopuksi testaussuunnitelmalle on oltava katselmointikäytännöt. (TMMi Foundation, 2010).

Testauksen valvonnan ja ohjauksen suhteen edellytetään testauksen suunnittelun parametrien valvontaa. Samoin oletetaan testausympäristön ja testausprojektiin riskien tilanteen jatkuva arviointi. Lisäksi on arvioitava tuotteen laatua suhteessa suunnitelmiin ja odotuksiin. Tuotteen laatu perustuu aloitus- ja lopetus-kriteereihin, vikoihin, sekä tuotteen riskeihin. Lisäksi on oltava määriteltynä toimintamallit korjaavien toimintojen suorittamiseksi. Taso 2 edellyttää myös testauksen suunnittelun ja suorittamisen osalta selkeät menetelmät. Testausolosuhteiden ja testitapausten tunnistaminen, määrittely ja priorisointi on oltava hallinnassa, sekä arvokkaan testidatan käyttö testauksen yhteydessä. On kyettävä määrittelemään testauksen suorituksen kuvaukset ja luomaan testidata. On raportoitava havainnoista ja luotava testauksen edetessä testilokit kuvaamaan testauksen tilannetta sekä vikakorttien käsittelyä. Myös testausympäristön vaatimusten tunnistaminen ja analysointi edellyttävät kykyä arvioida tuotteen testaukseen koottavaa ympäristöä. Lisäksi on oltava keinot raportoida ja hallita testausympäristön havaintoja. (TMMi Foundation, 2010).

Tasolla 3 keskitytään testausorganisaation määrittelyyn ja kokoamiseen. Tunnistetaan testaustoiminnot, kehitetään työnkuvauksia ja koordinoidaan testaus-tehtäviä testaajille. Taso edellyttää myös testaajien urakehityksen suunnittelua ja toteuttamista, sekä tunnistamaan testausprosessin kehitystarpeita. Käytännössä on kyettävä parantamaan testausprosessia hallinnoidusti ja valvomaan toteutusta. Testauksen elinkaariajattelu perustuu siihen, että tasolla 3 on käytössä standardit testausprosessit ja integroidut elinkaarimallit kehitysmallien kanssa. Käytettävissä on myös päättestaussuunnitelma, joka kattaa kaikkien projektien testauksen osalta yleiset asiat. Niitä ovat riskiarvioinnit, testauksen lähestymistavat ja organisaatio. Testauksen suorittamisen osalta taso edellyttää ei-toiminnallisten tuoteriskien tunnistamista ja analysointia. Lisäksi on kyettävä tunnistamaan testattavat toiminnot. Myös ei-toiminnalliset tilanteet on arvioitava ja priorisoitava osaksi testausta. Ei-toiminnalliselle testaukselle on oltava suorituksen kuvaukset, suorituskeinot ja raportointitavat. Lisäksi organisaatiolla on oltava käytössään testauksen tausta-aineiston staattinen testaus. (TMMi Foundation, 2010).

Taso 4 keskittyy testauksen mittareiden määrittelyyn. Käytännössä tiimillä on kyky valikoida mitattavat kohteet ja keinot määrittellä mittarit, tulosten keräys- ja tallennusjärjestelmät, sekä tulosten arviointikeinot. Lisäksi mittaamisen on oltava säännöllistä ja jatkuvaa. Tuotteen laadun arvioinnin perusteella on tunnistettava tuotteen laadun tarpeet ja määriteltävä tavoitteet laadulle. Lisäksi on kyettävä valikoimaan mittarit, tuotettava mittaustulosta koko tuotteen elinkaaren ajan ja osattava analysoida tuloksia verraten niitä asetettuihin tavoitteisiin. Tasolla edellytetään myös kehittyneitä katselmointimenetelmiä. Käytännössä katselmoinneille määritellään mittarit, joiden perusteella arvioidaan katselmointien vaikutusta tuotteen laatuun, kerätään tuloksia ja analysoidaan niitä. Lisäksi arvioidaan tulosten vaikutusta tuotteiden riskeihin. (TMMi Foundation, 2010).

Taso 5 keskittyy optimointiin, jolloin organisaation on kyettävä ennalta ehkäisemään toiminnallaan vikoja. Tiimin on kyettävä priorisoimaan ja määrittelemään sellaisia toimintoja, joilla systemaattisesti vältetään vikojen syntyminen. Tiimin

ennustamiskyky nousee korkeaan arvoon. Laadunvalvonnan osalta tarvitaan vakaa prosessi, kyetään tunnistamaan prosessin ongelmakohdat ja reagoimaan niihin korjaavin toimenpitein. Testausta suoritetaan hyödyntäen tilastollisia menetelmiä siten, että toiminta perustuu toiminnallisiin ja käytettäviin keinoihin. (TMMi Foundation, 2010).

TPI

Test Process Improvement (TPI) käyttää jatkuvaa esittämistä TMMi-tyyppisen portaittaisen esittämisen sijaan. Malli on tarkoitettu testausprosessin parantamiseen. Sen perusteella jäsennetään missä prosessi on, miten sitä pitäisi kehittää ja miten kehittäminen pitäisi toteuttaa. Malli koostuu neljästä pääalueesta, jotka ovat elinkaari, organisaatio, ympäristö ja työkalut sekä tekniikat. TPI-mallia on kehitetty eteenpäin siten, että nykyään arviointialueita on kaiken kaikkiaan 20. Niiden näkökulmasta arvioidaan organisaation tasoa. Jokainen arviointialue sisältää 1-4 tasoa (A-D), joiden välillä on tarkastuspisteitä ja parannusehdotuksia. Arviointialueiden määritelmässä oletetaan, että taso A on hallinnassa myös organisaation tavoittaessa tason B. Näiden tarkastuspisteiden kautta organisaatio kykenee arvioimaan oman tasonsa. Pääalueet, niihin liittyvät avainalueet ja arviointitasot, on kuvattu seuraavissa taulukoissa (Taulukko 2, Taulukko 3, Taulukko 4 ja Taulukko 5). (Hess 2008 & Stenberg, 2007).

Taulukko 2 Ympäristöön ja työkaluihin liittyvät avainalueet

Avainalue	Taso A	Taso B	Taso C
Testauksen työkalut	Suunnittelun ja valvonnan työkalut	Suorituksen ja analysoinnin työkalut	Laaja testausprosessin automatisointi
Testausympäristö	Hallinnoitu ja valvottu testausympäristö	Testaus mahdollisimman soveltuvimassa ympäristössä	Ympäristö tarpeiden mukaan
Toimistoympäristö	Sopiva ja ajanmukainen toimistoympäristö	-	-

Ympäristöön ja työkaluihin liittyviin avainalueisiin kuuluu testaustyökalut, jotka kattavat automatisoidut testauksen työkalut. Testausympäristö sisältää menetelmät, ohjelmistot, laitteistot ja kommunikointitavat. Ympäristön kannalta on merkittävää määritellä vastuut, hallinta, saavutettavuus ja ympäristön joustavuus, sekä vastaavuus asiakkaan ympäristöjen kanssa. Toimistoympäristön osalta arvioidaan työhuoneiden ja kalusteiden ergonomisuutta, soveltuvuutta työhön, sekä käytettävissä olevia tietokoneita työn suorittamiseksi. (Hess 2008).

Taulukko 3 Elinkaareen liittyvät avainalueet

Avainalue	Taso A	Taso B	Taso C	Taso D
Testausstrategia	Strategia yhdelle korkean tason testaukselle	Yhdistetty strategia korkean tason testaukselle	Yhdistetty strategia korkean testaukselle sekä alemman tason testit tai arviointi	Yhdistetty strategia kaikille testeille ja kaikille arviointitasoille
Elinkaarimalli	Suunnittelu, dokumentointi, suorittaminen	Suunnittelu, valmistelu, dokumentointi, suorittaminen ja lopettaminen	-	-
Testauksen mukaantulo	Testauksen valmistaminen	Testausvaiheen aloittaminen	Vaatimusten määrittelyn aloittaminen	Projektin alustusvaihe

Elinkaareen liittyvät avainalueet koostuvat testausstrategiasta ja testauksen mukaantulosta tuotekehitykseen. Tavoitteena on, että testaus on osa prosessia mahdollisimman alusta lähtien. Lisäksi arvioidaan elinkaarimallin monipuolisuutta. Testausprosessissa voi olla useita eri vaiheita ja elinkaarimallin avulla pyritään parantamaan testausprosessin ennustettavuutta ja hallittavuutta. (Hess 2008).

Taulukko 4 Tekniikoihin liittyvät avainalueet

Avainalue	Taso A	Taso B	Taso C	Taso D
Työn arviointi ja suunnittelu	Tuntuva arviointi ja suunnittelu	Tilastollisesti tuntuva arviointi ja suunnittelu	-	-
Testauksen suunnittelu	Epämuodolliset tekniikat	Muodolliset tekniikat	-	-
Staattiset testausmenetelmät	Taustamateriaalin tarkastukset	Tarkastuslisät	-	-
Metriikka	Projektin tuotemittarit	Projektin prosessin mittarit	Järjestelmän mittarit	Organisaation mittarit (> 1 järjestelmä)

Tekniikoihin perustuvat avainalueet liittyvät työn arviointiin ja suunnitteluun, testauksen suunnitteluun, staattisiin testausmenetelmiin ja mittausmenetelmiin. Arvioinnilla osoitetaan toimintojen suoritusajankohdat ja tarvittavien resurssien määrät. Testitapausten määrittelyn suhteen on suositeltavaa laatia toimintamallit, joita noudatetaan ja joilla varmistetaan testien kattavuus. Staattisten testausmenetelmien käyttöönotto vähentää vikojen määrää varsinaisessa dynaamisessa testauksessa. Lisäksi mittareiden avulla voidaan arvioida tuotteen tai prosessin tilaa ja tulosten perusteella arvioida kehittämiskohteista ja muutostarpeita. (Hess 2008).

Taulukko 5 Organisaatioon liittyvät avainalueet

Avainalue	Taso A	Taso B	Taso C	Taso D
Sitoutuminen ja motivaatio	Budjetin ja ajan hallinta	Testaus on integroitu osaksi projektiorganisaatiota	Testausin-sinöörit	-
Tukiryhmät ja koulutus	Testauspäällikö ja testaajat	Menetelmät, tekniikat ja toiminnallinen tuki, hallinnointi	Muodollinen sisäisen laadun varmistaminen	-

Menetelmien käyttöalue	Projektikohtainen	Organisatiokohtainen	Organisaation optimointi	-
Kommunikointi	Sisäinen kommunikointi	Projektin kommunikointi (viat, muutosten valvonta)	Organisaation vuorovaikutus liittyen testausprosessin laatuun.	-
Raportointi	Viat	Edistyminen (testit, tuotteet, kustannukset, aikataulu, tavoitteet, viat)	Riskit ja suositukset, mittareiden avulla	Suosituksset, joissa on ohjelmistoprosessiin liittyviä kehittämiskohtia
Vianhallinta	Sisäinen vianhallinta	Laaja vianhallinta joustavilla raportointimenetelmillä	Projektin vianhallinta	-
Testimateriaalin hallinta	Sisäinen testimateriaalin hallinta	Ulkoinen taustamateriaalin ja testauskohteiden hallinta	Uudelleenkäytettävä testausmateriaali	Järjestelmävaatimusten jäljitettävyys vaatimuksista testitapauksiin
Testausprosessin hallinta	Suunnittelu ja suorittaminen	Suunnittelu, suorittaminen, valvonta ja sovittaminen	Valvonta ja sovittaminen organisaatiotasolla	-
Tarkastukset ja katselmoinnit	Arviointitekniikat	Tarkastusstrategia	-	-
Yksikkötestaus	Alemman tason testauksen elinkaari (suunnittelu, dokumentointi ja suorittaminen)	Lasilaatikkotekniikat	Alemman tason testi-strategia	-

Tekniikoihin perustuvat avainalueet liittyvät sitoutumiseen ja motivointiin eli kykyyn sitouttaa testaustiimin jäsenet testausprosessin läpiviemiseksi. Lisäksi on saatava sitoutettua projektiryhmä ja johto testaukseen. Testaustiimin kehittämi-

nen ja kyky suoriutua kaikista projektien asettamista vaatimuksista edellyttää kouluttamista ja osaamisen jakamista. Vuorovaikutuksen ja työskentelytapojen osalta toiminnassa on huomioitava rajapinnoissa toimivat osapuolet. Raportoinnin ja katselmointien avulla jaetaan tietoa ja arvioidaan työn tasoa. Materiaalin ja testausprosessin hallinnalla varmistetaan testauksen hallittu eteneminen ja jäljitettävyyden varmistaminen. (Hess 2008).

Avainaluetta tarkastellaan skaalalla 0...13. Skaalan osalta arvioidaan suoriutumisista hallittavuuden, tehokkuuden ja optimoinnin perusteella. Organisaation tasojen arviointia varten on laadittu tarkastuslistoja, joihin vastaamalla voidaan arvioida oman testausprosessin taso. Skaalan ja tasojen limittyminen toisiinsa on esitetty seuraavassa taulukossa, Taulukko 6. Menetelmä ei huomioi ketteriä testausmenetelmiä, vaan antaa yleisiä suuntaviivoja testauksen kehittämiseksi. Menetelmää tuleekin soveltaa organisaatiossa tarpeiden mukaisesti. (Istqb Advanced, 2011 & Hess, 2008).

TMMi ja TPI mallit voidaan yhdistää toisiinsa tasojen 1-3 osalta taulukon, Taulukko 6, mukaisesti. (TMMi Foundation, 2010).

Taulukko 6 TMMi ja TPI -arvioiden yhdistäminen

	Hallittu					Tehokas					Optimoitu			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Testistrategia		A					B				C		D	
Elinkaarimalli		A			B									
Testauksen mu- kaantulo			A				B				C		D	
Työn arviointi ja suunnittelu				A							B			
Testauksen suun- nittelu		A		B										
Staattiset tes- tausmenetelmät					A		B							
Metriikka						A			B			C		D
Testauksen työka- lut				A				B			C			
Testausympäristö				A				B						C
Toimistoympäristö				A										
Sitoutuminen ja motivaatio		A					B					C		
Tukiryhmät ja kou- lutus				A			B			C				
Menetelmien käyt- töalue					A						B			C
Kommunikointi			A		B							C		
Raportointi		A			B		C					D		
Vianhallinta		A				B		C						
Testimateriaalin hallinta			A			B				C				D
Testausprosessin hallinta		A		B								C		
Tarkastukset ja katselmoinnit							A			B				
Yksikkötestaus					A		B		C					

Kuvassa eri TMMi-tasot sisällytetty TPI-taulukkoon. Eri tasot on eroteltu toisistaan eri violetin sävyillä. TMMi -mallin aloittelija taso vastaa TPI-mallin 0-tasoa. Vastaavasti TMMi-mallin toinen taso vastaa kuvassa seuraavaa keskivioletilla olevia arvoja. TMMi-mallin kolmannen tason integrointi malliin kattaa lähes kokonaan TPI-mallin kriteerit (tumma violetti). (TMMi Foundation, 2010).

CTP

Critical Testing Process (CTP) -menetelmän mukaan testausprosessi voidaan jakaa pieniksi prosesseiksi, joista tietyt testausprosessit voidaan asettaa kriittisiksi. Nämä prosessit tukevat hyvin hoidettuina testaustiimin työskentelyä. Vastaavasti, jos prosesseja ei ylläpidetä hyvin, jopa lahjakkaat testaajat ja testauspääälliköt epäonnistuvat työssään. Mallin mukaan on tunnistettavissa 12 kriittistä testausprosessia. Niitä ovat: (Hess, 2008).

- Testaus
- Kohteen arviointi
- Laadun riskianalyysi
- Testauksen arviointi
- Testauksen suunnittelu
- Testaustiimin kehittäminen
- Testausjärjestelmien kehittäminen
- Testausjulkaisujen hallinta
- Testauksen suorittaminen
- Vikojen hallinta
- Tulosten raportointi
- Muutoksen hallinta

Prosessin kehittäminen alkaa olemassa olevan testausprosessin osalta. Aluksi tunnistetaan, mitkä prosessit ovat vahvoja ja mitkä heikkoja. Kehittämisen priorisointi suoritetaan organisaation tarpeiden perusteella. Yleensä CTP-prosessin

yhteydessä hyödynnetään mittareita liittyen vikojen havaitsemisprosenttiin, vaatimusten kattavuuteen ja riskien kattavuuteen. Laadullisia tekijöitä ovat testaustiimin rooli ja tehokkuus, testaus suunnitelma, testaustiimin taidot, toimialan ja teknologian tuntemus. Muita laatutekijöitä ovat vikojen, testitulosten ja muutosten hallinnan raportointimenetelmät ja tasapaino. Kun menetelmässä tunnustetaan heikko alue, suunnitellaan kyseiseen kohtaan parannukset. (Istqb Advanced, 2011).

STEP

Systematic Test and Evaluation Process (STEP) on CTP:een kaltainen, mutta ei TMMi:n ja TPI:n kaltainen malli. Menetelmä ei vaadi, että muutokset tehdään jossakin tietyssä järjestyksessä. Menetelmä sisältää vaatimuspohjaisen testausstrategian. Testaus alkaa elinkaaren alussa, testejä käytetään vaatimusten ja käytön malleina. Testausmateriaalin suunnittelu ohjaa ohjelmiston suunnittelua. Lisäksi vikoja havainnoidaan aikaisemmin ja pyritään estämään. Viat analysoidaan systemaattisesti. Testaajat ja kehittäjät toimivat yhdessä. Käytännössä testaus alkaa vaatimusten määrittelyn yhteydessä ja jatkuu aina siihen asti, kunnes järjestelmän elinkaari päättyy. (Istqb Advanced, 2011).

Malli pohjautuu vaatimuspohjaiseen testausstrategiaan, jolla varmistetaan testitapausten aikainen laadinta. Niiden laadinnan kautta suoritettava vaatimusten todentaminen on ohjeena suunnittelulle ja koodaamiselle. Mallilla keskitytään parantamaan testauksen kolme osa-aluetta: suunnittelua, hankintaa ja mitaamista. Mittareita ovat muun muassa testausvaatimusten ja riskien kattavuus, vikojen korjaustehokkuus, vikatiheys, testauskustannukset ajan ja hyödyn perusteella, sekä asiakastyytyväisyys. Monesti Step-malli on sisällytetty TPI-malliin. (Istqb Advanced, 2011).

2.5 Kanban

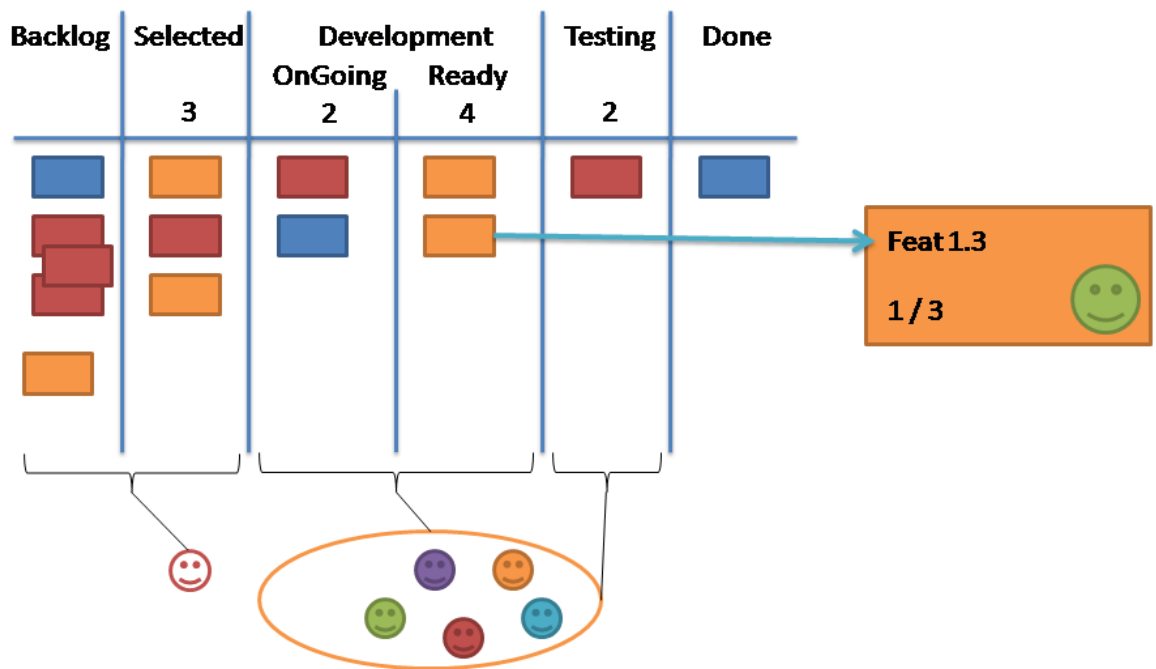
Menetelmästä ja kehittämistavasta riippumatta on hyvä huomioida, että yhtä oikeaa vastausta kehittämiseen ei ole. Ei ole esimerkiksi mahdollista automatisoida testausta 100 %:sti ja jättää muu testaus suorittamatta. Ei voida jäädä odottamaan, että testaus tulee näin suoritetuksi täydellisesti, kevyesti ja sitä voidaan suorittaa aina tarvittaessa. Toisaalta ei voida jättää testausta väliin ajatteleamalla, että asiakas antaa palautteen toimimattomasta sovelluksesta. Käytännössä näillä tavoin vikojen määrä kasvaa ohjelmiston kehittyessä. Ohjelmisto saadaan toimimaan ja riskien realisoitumisen todennäköisyys laskemaan, kun huomioidaan kaikki osapuolet ohjelmistokehityksessä. On tiivistettävä toiminta kohti samaa tavoitetta: saada aikaan toimiva ohjelmisto asiakkaalle ja näin kasvattaa asiakastytyvyyttä. Tähän vaikuttavat testaajan identiteetti ja erikoisosaaminen. Myös jatkuvasti toimintakykyinen ja vakaa ohjelmisto, sekä erilaisia testaustekniikoita osaavat erityistestaajat tukevat kehittymistä. Erilliset testaustiimit, testattavuusvaatimukset ja testattavuuskatselmoinnit varmistavat testauksen onnistumisen yhdessä. Tätä tuetaan tarpeellisten testaustekniikoiden käytöllä ja tiiviillä vuorovaikutuksella. (Vuori 6-2010).

Tiimeissä voidaan hyödyntää eri menetelmiä projektien läpiviemiseksi. Tunnettuja menetelmiä ovat RUP:n, XP:een ja Scrum. Myös testaustiimissä on huomioitava projektiryhmien käyttämät menetelmät. Hankaluuden tuo se, että projektit voivat usein valita itse käyttämänsä menetelmät. Näin ollen testauksen toiminnoista suoriutuminen on haasteellista. Edellä mainittujen menetelmien ongelmat on tunnistettu ja niitä helpottamaan on kehitetty Kanban-menetelmä. Kyseinen menetelmä on Scrum-menetelmän kaltainen, mutta antaa vapaammat kädet ohjata projekteja. Toisaalta menetelmä soveltuu paremmin testaustiimin ohjaamiseen silloin, kun tiimin on suoriuduttava useista projekteista samanaikaisesti ja projektien scrummaus-väli on keskenään erilainen.

Kanban on Scrumin tavoin vain työkalu. Se ei kerro selkeästi miten projektissa pitäisi toimia, vaan antaa apuvälineitä työskentelyn ohjaamiseen. Scrumia ja

Kanbania verrattaessa toisiinsa voidaan havaita, että Kanban sisältää vähemmän rajoitteita työskentelyn ohjaamiseksi kuin Scrum. Toisaalta monet organisaatiot ovat siirtyneet Scrumista Scrumbaniin, joka on Kanbanin ja Scrumin välimuoto. Myös tällainen vaihtoehto on kannattava, jos se yrityksen toimintatapoihin ja työskentelyyn tulevien muiden asetusten sekä ohjeistusten mukaisesti on sopivinta. Idealisinta on ottaa useista menetelmistä sopivimmat käyttöön ja näin muodostaa yritykselle oma menetelmä. (Kniberg, 2009).

Kanban-menetelmässä ei ole rooleja. Käytännössä niitä voi olla, mutta niiden käyttö ei ole pakollista. Menetelmä ei myöskään kiellä roolien käyttöä. Roolien käytöllä on tarkoitus antaa lisäarvoa, ei määrittellä niitä aiheuttamaan konflikteja työn etenemisen suhteen. Scrumista poiketen Kanbanissa ei ole aikataulutettuja iteraatioita. Suunnittelua, prosessin kehittämistä ja julkaisuja tehdään silloin, kun se työskentelyn kannalta on idealisinta ja tarpeen. Menetelmä ei kuitenkaan estä esimerkiksi julkaisujen tuottamista joka toinen viikko, jos tiimi kokee sen tarpeelliseksi. Ainoat rajoitteet Kanbanissa ovat työnkulun ja työtilanteen visualisointi yksiselitteisesti, sekä WIP:n rajoittaminen. WIP tarkoittaa tehtävien määrää, joka on enintään mahdollista ottaa suoritukseen yhdessä työvaiheessa. Käytännössä tämä eroaa Scrumista siten, että Scrumissa WIP määritellään koko sprintin osalta suoritettavana maksimiyö määränä. Kanbanin osalta WIP pilkotaan vaihekohtaisesti esimerkkikuvan mukaisesti, Kuva 13. (Kniberg, 2009).



Kuva 13 Kanban - seinätaulu ja tehtäväkortti

Kanbania otettaessa käyttöön on aluksi hankala tietää, mitkä ovat oikeat WIP-arvot kuhunkin sarakkeeseen. Käytännössä arvoon vaikuttaa vaiheessa työskentelevien henkilöiden määrä ja tehtävien kesto. Lisäksi vaikuttaa vaiheen suorittamiseen vaikuttavat muut tekijät, kuten muiden tiimien vaikutus työn suorittamiseen. Prosessin edetessä ja kokemuksen karttuessa voidaan arvoja päivittää vastaamaan paremmin toiminnan tehokasta läpivientiä. Arvon nostaminen kannattaa, jos on havaittavissa vapaana olevia henkilöresursseja aina tietyssä vaiheessa. Samanaikaisesti on työn alle otettu sarakkeen sallima maksimimäärä tehtäviä. Vastaavasti arvoa pitää pienentää, jos tehtäviä on otettu alle WIP-määrän, mutta tehtävät eivät valmistu. Syynä voi olla joko resurssien puute tai seuraavana olevan vaiheen ongelmat. WIP-arvon tarkoituksena on osoittaa pulonkaulakohdat, jolloin prosessin kehittäminen on tarkoituksenmukaista. Näin autetaan prosessin läpiviennissä paremmin. (Kniberg, 2009).

Kanban-prosessi käynnistyy sillä, että ProductBacklogiin kootaan suoritettavat tehtävät. Tehtävät vietään Scrum-taulun tyyppiselle seinätaululle, johon on

määritelty juuri kanbania hyödyntävälle tiimille oleelliset vaiheet sarakkeina. Yhdellä seinätaululla voidaan esittää joko yhden projektin tai useiden projektien tehtävät. Tehtävistä valitaan kriittisimmät Selected-sarakkeeseen ja niiden suorittamisjärjestyksestä sovitaan yhdessä tiimin kanssa. Tehtävien valinnan ja priorisoinnin suorittamiseen on yleensä valittu tietyt henkilöt, kuten projektipäällikkö tai testauspäällikkö. Tiimin jäsenet osallistuvat joko kaikkiin prosessin vaiheisiin tai tiettyihin ennalta sovittuihin vaiheisiin. Kanban-prosessi sallii myös muutokset tehtävien jaossa, mutta muutokset tulisi huomioida WIP-arvoja määriteltäessä. Käytännössä yksi tiimin jäsen voi toimia yhdessä tai useissa rooleissa prosessin aikana. (Kniberg, 2009).

Tehtävien valinta työn alle voidaan tehdä eri tavoin. Käytännössä valintatavat sovitaan tiimin kanssa ennalta ja kaikki tiimin jäsenet noudattavat sovittuja menetelmiä. Tehtävät voidaan ottaa suoritettavaksi esimerkiksi siten, että aina ensimmäisenä valitaan seinätaululla ylimpänä oleva tehtävä, vanhin tehtävä tai mikä tahansa tehtävä. Lisäksi kriteereinä voidaan käyttää tehtävien tyyppiä. Tällöin sovitaan vanhoihin ominaisuuksiin liittyvien tehtävien suorittamisesta aina ensisijaisesti uusiin ominaisuuksiin verrattuna. Lisäksi voidaan valita, että tietyn väriset tehtävät on suoritettava aina ennen muita. Seinätaululle tehtävät voidaan merkitä projekteittain eri värein, niiden prioriteetin mukaan, tehtävän suorittajan mukaan tai muuten hyväksi havaitulla tavalla. Uusia ja vanhoja käsittelemättömiä tehtäviä priorisoidaan ja otetaan listalle jatkuvana prosessina. Selected-sarakkeessa voi olla enimmillään tehtäviä vain WIP-arvon kuvaama määrä. (Kniberg, 2009).

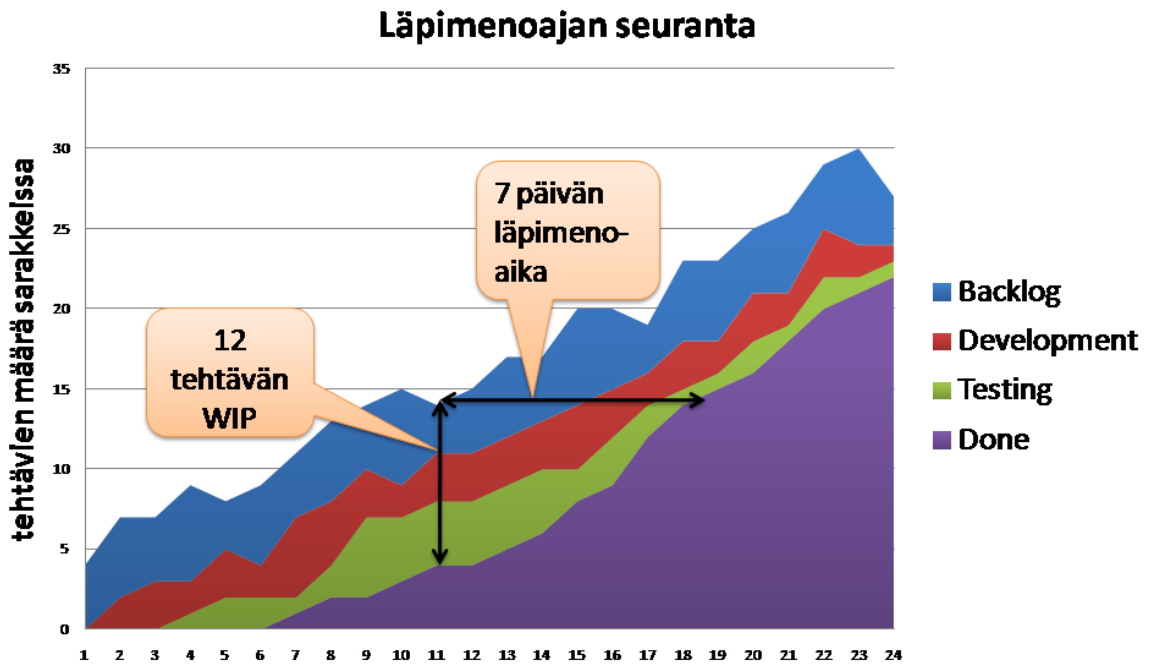
Tehtävät otetaan työn alle joko itsenäisesti tai päivittäisissä palavereissa yhteisesti sopien. Aina kuitenkin noudatetaan yhteisesti sovittuja käytäntöjä. Seinätaululla edetään vaiheittain vasemmalta oikealle tehtävän valmiusasteen mukaisesti. Tehtävässä kuvataan tehtävän tarkoitus, kuinka kauan tehtävän suorittamisen arvioidaan kestävän ja kuka tehtävän on ottanut hoitaakseen. Lisäksi päivitetään aktiivisesti tietoa siitä, paljonko työstä on jo tehty tai paljonko työstä on vielä tehtävänä. Tämäkin on sovittava tiimin kanssa ennen työn aloittamista.

Tehtävien siirrossa huomioidaan aina se, että yhdessä sarakkeessa saa olla enimmillään WIP-arvon kuvaava määrä tehtäviä. Etu mallissa on siinä, että isojen taskien pakollista pilkkomista ei tarvitse suorittaa huomioiden sprinttien kesto kuten Scrum-menetelmän yhteydessä. Seinätaululla on tarkasteltavana kerrollaan koko tilanne, eikä vain yhden sprintin sisällä suoritettavien tehtävien tila. (Kniberg, 2009).

Kanban-menetelmässä ei ole otettu kantaa siihen, että voidaanko tehtäviä siirtää takaisin prosessin alkuun päin. Ei myöskään siihen, että voidaanko joidenkin vaiheiden yli hypätä. Käytännössä tämäkin on sovittavissa tiimin kesken. Läpimenoaikojen seurannan kannalta ja pullonkaulojen välttämiseksi tehtävien palauttaminen seinätaululla alkupään sarakkeisiin haittaa prosessin etenemistä. Tällöin on suositeltavaa laatia tehtävän täydentämistä varten uusi tehtävä tai huomioida sarakkeita määriteltäessä sopivimmat sarakkeet tiimin toiminnan kannalta. Tiimi pitää päivittäisiä palavereja, mutta Scrumista poiketen palaverissa ei niinkään esitetä jokaisen tiimin jäsenen suorittamia tehtäviä. Tämä sen vuoksi, että seinätaululla saattaa olla useiden tiimien tehtävät. Suuren ryhmän palaverien läpikäynti saattaa venyä ja haitata työn tehokasta etenemistä. Palaverissa keskitytään ennen kaikkea koko ryhmälle oleellisten tietojen jakamiseen ja havainnoimaan mahdollisia ongelmakohtia. (Kniberg, 2009).

Ongelmakohtat ratkaistaan siten, että resursseja siirretään ratkaisemaan ongelmatehtävää aina resurssin vapautuessa omasta tehtävästään. Uusia tehtäviä ei oteta, eikä tehtäviä siirretä prosessissa eteenpäin niin kauan kuin ongelma on ratkaisematta. Tarvittaessa ongelmaa ratkaisee koko tiimi, jos muu ei tehtävän ratkaisemiseksi auta. Ongelmakohtat merkitään seinätaululle selkeästi siten, että jokainen tiimissä työskentelevä tietää tehtävien priorisoinnin oman tehtävän valmistuttua. Jokainen auttaa ensisijaisesti ongelman kanssa. Ongelman ratkaisun jälkeen arvioidaan ongelman yleisyys ja syyt mistä ongelma on johtunut. Näiden syiden perusteella tehdään arviot siitä, pitääkö WIP-arvoa muuttaa, tarvitaanko lisäresursseja kyseiseen vaiheeseen tai lisäosaamista. (Kniberg, 2009).

Koko prosessin seurantaan varten tarvitaan muitakin mittareita kuin WIP-arvon seuranta. Kumulatiivinen seurantakaavio antaa hyvin tietoa WIP-arvojen toimitavuuden suhteen ja kertovat tiimille tehtävistä suoriutumisaajoista. Seuraavassa kuvassa on esitetty esimerkki läpimenoajan seurantakaaviosta, Kuva 14. (Kniberg, 2009).



Kuva 14 Kumulatiivinen seurantakaavio

Kaavioon on merkitty tapa arvioida WIP-arvon (12 tehtävää) mukaisten tehtävien läpimenoaikaa (7 päivää). Arvoja voidaan seurata myös yksittäisen sarakkeen läpimenoaikojen osalta. Näin saadaan arvokasta tietoa vaiheen työkuormuksista ja vaikutuksista koko prosessin kehittämisen kannalta. Samalla voidaan seurata resurssien käyttöastetta ja ennakoida. Työskentelyn kehittämisen kannalta on tärkeää etsiä tiimin kehittämisen kannalta merkityksellisimmät mittarit. Kanbanin etuna on se, että ongelmat ja vastaavasti hyvin toimivat prosessivaiheet on helposti nähtävillä kaikille tiimin jäsenille. Tiimin jäsenet näkevät myös hyvin selvästi tulevat työkuormat. Näin tiimi saadaan osallistumaan tiiviimmin tiimin toiminnan suunnitteluun ja osaamisen jakamiseen. (Kniberg, 2009).

3 SOVELTUVUUS TESTAUKSEN OHJAAMISEKSI

Kirjallisuuden ja kokemusten kautta saadun hyödyn perusteella on selvää, että tarkastelunäkökulmia on hyvä hyödyntää myös testaustiimin ohjaamisessa ja osaamisen johtamisessa. Testaustiimiä ei voida rinnastaa projektiryhmään, koska testaustiimin kohde poikkeaa projektiryhmien kohteesta merkittävästi. Projektiryhmällä on tavoitteena projektin tavoitteiden määrittely, suunnittelu, toteutus ja testaus, sekä järjestelmän toimittaminen sovittuna ajankohtana asiakkaalle. Testaustiimin tavoitteet ovat moninaisemmat. Tiimin vastuulla on huolehtia useiden projektiryhmien tuottamien järjestelmien testauksesta huomioiden jokaisen projektin aikataulut ja tarpeet. Testauksen on vaativaa johtuen useista eri testaustavoista ja vaiheista, joissa testauksen pitäisi olla mukana. Työstä suoriutuminen edellyttää tiimiltä erityisosaamista. Voidaan toki ajatella, että testaustiimin sijaan testaajat voitaisiin jakaa projekteihin suorittamaan testaustyötä joko osana projektiryhmää tai itsenäisinä testaajina. Tällöin testaajat jäävät yksin. Tukea osaamiseen tai työn suunnitteluun ja suorittamiseen ei ole saatavissa muilta testaajilta. Testausnäkökulman käyttö projektin edetessä saattaa hämärtä kiinteiden tiimisuhteiden syntyessä projektiryhmän kanssa sen sijaan, että kyettäisiin säilyttämään kriittistä tarkastelukantaa testattavaan tuotteeseen.

Testaustiimin ohjaaminen ja osaamisen johtaminen käsitellään kolmesta eri näkökulmasta. Ensimmäinen näkökulma liittyy testaustiimin ohjaamiseen testausnäkökulmaan, testauksen perustehtävän tarkasteluun ja Kanban-menetelmän soveltuvuuteen testauksen päivittäisen työskentelyn työkaluksi. Lisäksi tarkastellaan perustehtävän suorittamista ja esimiestyöskentelyä osana päivittäistä työtä. Toinen näkökulma käsittää testauksen osaamisen johtamisen, jossa arvioidaan testauksessa hallittavia osaamisalueita huomioiden erilaiset testaustekniikat. Lisäksi tarkastellaan testaajien osaamisen kehittämistä siten, että kehittäminen täyttää yksilön toiveiden lisäksi tiimin tarpeet. Samassa yhteydessä arvioidaan muutoksen valmisteluun ja läpivientiin liittyviä seikkoja testaustiimissä, jossa huolehditaan useista projekteista samanaikaisesti. Kolmas näkökulma

keskittyy testauksen kehittämiseen testaustiimille soveltuvan kehittämismallin avulla ja mittareiden valintaa niin tuotteen kuin koko tiimin tarpeet huomioiden.

3.1 Testaustiimin ohjaaminen

Testausnäkökulman huomiointi

Testaustiimin kohteena olevien projektien tavoitteet ovat pääosiltaan yhtenevät. Jokaisen projektin taustalla on asiakkaan vaatimukset ja yleensä asiakkaana on yksi yksilöity toimeksiantaja. Asiakasvaatimusten tulkitseminen ja saattaminen yksiselitteiseen muotoon asettaa haasteet niin projektiryhmän kuin testaustiimin toiminnalle. Vaihe on kuitenkin kriittinen jo sen vuoksi, että testaustiimin olisi mahdollista hahmottaa järjestelmän toimintatapa ja sen pohjalta alkaa suunnitella järjestelmän testausta. Mitä pidemmälle vaatimusten muuntaminen toiminoiksi vie, sitä vähemmän aikaa testaajilla on varsinaiseen dynaamiseen testaukseen. Toisaalta, jos vaatimusten riittävä pilkkominen jätetään suorittamatta, on väärin tulkintojen todennäköisyys suuri. Tällöin testaustulosten luotettavuus kärsii ja epävarmuus järjestelmän toteutuksen tasosta kasvaa.

Testausstandardit ottavat selkeästi kantaa testauksen tehtäviin ja antavat selvät rajat sille, mitä testaustiimiltä voidaan suoritukseltaan odottaa. Lisäksi avataan tarpeet, joita tiimillä on työstään suoriutuakseen. Niiden mukaisesti tulisi projektiryhmien ohjata toimintaansa testauksen rajapinnassa. Testauksen tarkoituksena on varmistaa järjestelmän toimivuus asetettujen vaatimusten mukaisesti, varmistaa soveltuvuus käyttötarkoitukseensa ja löytää mahdollisimman paljon vikoja ja puutteita järjestelmän käytössä. Paljon keskustelua aiheuttaa se, että riittääkö testauksessa vain asiakasvaatimusten varmistaminen. Vastaukset vaihtelevat verrattaessa eri rooleissa toimivien mielipiteitä. Testausstandardit, seminaareissa ja kokemuksen kautta saadut arviot ovat yhtenevät. Testauskohteet on arvioitava tapauskohtaisesti huomioiden tuotteen ja projektin riskit. Yleensä koetaan, että puhtaasti asiakasvaatimuksista johdettujen toimintojen

testaus ei ainoastaan riitä. Testauksen on otettava kantaa myös muuhun järjestelmän toimintaan. Tällöin testausilanteet voidaan luokitella kriittisyyden mukaisesti. Priorisointia tehden keskitetään suurin voimavara asiakasvaatimusten mukaisuuden testaukseen, mutta samalla myös varmistaa muu toiminnallisuus. Erityisesti, jos toteutuksen yhteydessä on tehty lisätoimintoja käyttöä helpottamaan, on muun toiminnallisuuden varmistus merkityksellistä.

Vaatimuksia voidaan tarkastella esimerkiksi tapauksessa, jossa usean tiedon esittäminen voidaan toteuttaa kukin omassa näkymässään. Toisaalta tieto voidaan esittää myös yhdessä näkymässä siten, että näkymän tietoja voidaan suodattaa eri tavoin. Molemmat toteutukset vastaavat asiakkaan vaatimukseen tiedon esittämisen suhteen, mutta jälkimmäisessä toteutuksessa on käyttöön otettu suodatus-toiminto. Puhtaasti asiakasvaatimuksia tarkastellen voidaan ajatella, että suodatuksen testaus ei ole oleellista, koska juuri sitä toimintoa asiakasvaatimuksissa ei ole edellytetty. Toisaalta voidaan nähdä, että tietojen tarkastelua voidaan edesauttaa suodatus-toimintoa käyttäen. Tällöin on oletettavaa, että käyttäjä myös käyttää toimintoa. Käyttötarkoitusta ja tarkastelun toimivuutta arvioitaessa on aiheellista suorittaa myös suodatus-toiminnon testaaminen, vaikka asiakas ei ominaisuutta ole erityisesti pyytänyt.

Lisäksi asiakkaan esittäessä vaatimuksia on lukuisia olettamuksia, joita järjestelmässä oletetaan olevan ilman erillistä vaatimusta. Näitä voi olla muun muassa toiminnon suorittamisen keskeyttäminen tai pikatoimintojen määrittäminen käyttöä helpottamaan. Lisäksi voidaan olettaa, että järjestelmä ei mahdollista virheellisten syötteiden käyttöä ja toisaalta se kykenee suoriutumaan sillä suoritetuista toiminnoista. On oletettavaa, että kaikki järjestelmään toteutetut toiminnot tai toteuttamatta jätetyt virhetilanteiden käsittelyt kohdataan asiakaskäytössä jossakin vaiheessa. Tällöin käyttäjäystävällinen toiminta ja toipuminen virheistä arvostetaan korkealle. Mahdollisten vioista aiheutuvien häiriöiden ilmaantuminen aiheuttaa ärsyyntymistä käyttäjässä ja antaa huonon kuvan järjestelmän toimittajasta, vaikka viat eivät liittyisikään suoraan asiakasvaatimuksiin.

Testaustiimin tehtävää pohtiessa on hyvä ymmärtää, että testauksen tehtävänä ei ole virheettömän tuotteen todentaminen. Testaajien on täysin mahdotonta testata kaikkia mahdollisia järjestelmän käyttötapoja kaikilla mahdollisilla syönteillä ja kaikilla mahdollisilla käyttöympäristöillä. Testaustiimin tehtävänä on arvioida testattavat asiat ja niiden testaustarpeen laajuus. Myös testauksen sijainti projektissa on merkityksellistä. Testaustyön aloittaminen staattisella testauksella jo projektin alkuvaiheessa on haasteellista. Useimmat projektiryhmät eivät vielä hahmota testaustiimin tehtävänkuvaa, työkuormaa ja toimintatapoja. On erittäin yleistä, että testaustiimin oletetaan sulautuvan jokaisen projektiryhmän toimintamalleihin ja tuottavan projektille ryhmän määrittelemässä ajassa tuotoksen. Onnistunut yhteistyö ja vuorovaikutus edellyttävät koko tiimille määriteltävien yhteisten toimintamallien luomista testauksen rajapinnassa. Sen sijaan, että jokainen testaaja opettelisi lukuisten projektien työskentelytavat, olisi hyvä projektien jakaa keskenään kokemuksia hyväksi havaituista käytänteistä. Olisi tarpeen pyrkiä luomaan yhteistä yrityskulttuuria siten, että testauksen rajapinnoista toimiminen saataisiin yhteneväksi. Tämä tukisi omalta osaltaan myös toimintajärjestelmän noudattamista ja yhteisen toiminnan kehittämistä koko yrityksen tavoitteet huomioiden.

Yhteisten menetelmien kautta saataisiin staattisen testauksen määrää kasvatettua ja vikojen määrää laskettua projektien loppupäästä. Toki toiminta edellyttää sitä, että staattiseen testaukseen annettujen ominaisuuksien vaatimukset ja toteutustavat kyettäisiin hahmottamaan aikaisemmassa vaiheessa. Tällöin staattisen testauksen suorittamisesta olisi hyötyä. Toisaalta staattisen ja dynaamisen testauksen toteutusajankohdat eivät ajoittuisi liian tiiviisti toistensa kanssa. Erittäin merkittävään asemaan nousee se, että projekteissa on tarpeen arvioida dokumentaation merkitystä. Asiakkaan vaatimusten mukaisesti dokumentteja laaditaan paljon, mutta dokumentaation hyödyntäminen kehitystyössä ei ole kaikkien projektien osalta aktiivista. Tämä haittaa merkittävästi testaustiimin kykyä suoriutua heille asetetuista vaatimuksista oman työnsä suorittamisen suhteen. Lisäksi heidän keinonsa oppia tuote ja sen käyttötarkoitus hämärtyy, kun työstä suoriutumiseen tarkoitettu dokumentaatio valmistuu liian myöhään.

Voidaan myös ajatella, että mikä on syynä dokumentaation myöhäiseen syntyyn. Usein koetaan, että asiat on keskusteltu projektipalavereissa tai tieto tarkentuu vähitellen tuotteen kehittämisen edetessä. Tällöin voidaan ajatella, että onko dokumentaation laadintaan käytetty aika tarpeellista. Jos tiedot käydään läpi palavereissa, olisi selkeämpää kirjata sovitut asiat epämuodollisempaan muotoon myöhempää käyttöä varten ja näin saattaa vähitellen sovitut asiat testaajien tietoisuuteen. Näin testaustiimi voisi omalta osaltaan koota testausmateriaalia vähitellen tietojen karttuessa.

Yksittäisen testaajan osallistumisessa kiinteästi projektiryhmän toimintaan on havaittu paljon ongelmia. Testaajat kokevat joutuvansa työskentelemään yksin omaa näkökulmaansa noudattaessaan. Ne testaajat, jotka kokevat työskentelyn hyväksi ja tiiviiksi projektiryhmän kanssa, ovat usein unohtaneet testausnäkökulman ja siirtyneet kehittäjä-testaaja väliseen tilaan. Tällöin järjestelmän tarkastelu on hämärtynyt. Työskentely on usein rutinoitunut siten, että järjestelmästä esiin tuleviin häiriöihin ei reagoida samalla voimakkuudella kuin projektiryhmästä riippumattomat toimijat. Testaajat ovat antaneet myös palautetta siitä, että järjestelmästä tai sen kehittämisestä annettuun palautteeseen ei reagoida riittävällä vakavuudella. Tällöin kehitysehdotuksien huomiotta jättäminen vaikuttaa helposti siihen, että myöhemmin käyttäjien saama kokemus tuotteesta osoittautuu testaajien kokemuksen kanssa samankaltaiseksi. Voimakas projektorientoitunut näkökulma keskittyy usein sopimusehtoihin ja asiakasvaatimukseen. Järjestelmän käytettävyysoongelmiin ei välttämättä puututa riittävästi. Näissä tilanteissa saattaa perustoiminnoiltaan hyvä järjestelmä saada käyttäjiltä palautteen tuotteen toimimattomuudesta, kun käyttäjä kohtaa useita käytettävyyso ongelmia lyhyenkin käyttökokemuksen yhteydessä.

Testaajilta odotetaan yksilöinä laajaa osaamista erilaisista testaustekniikoista. Yksin testaajan on hankala kehittää osaamistaan tai edes hahmottaa omia kehitystarpeitaan. Testaus koetaan usein helpoksi työksi, jonka suorittamisesta selviytyisi kuka tahansa. Testausstandardit kuvasivat hyvin kattavasti testauksen moninaisuutta ja sitä, että testausosaaminen kasvaa vuosikymmenten aikana

suoritetun työn kautta, sekä erilaisilla erityisosaamisiin kohdistetuilla testauskoulutuksilla. Lisäksi testaajien verkostoituminen keskenään parantaa mahdollisuuksia kehittyä työssään. Testaustiimityöskentelystä saatujen kokemusten perusteella tunne omasta työstä suoriutumiseen on kasvanut. Testaajat kokevat saavansa tukea toisista testaajista, oppivat toisilta testaajilta erilaisia testaus-tekniikoita ja kehittyvät omissa erityisosaamisalueissaan seuratessaan toisten testaajien työskentelyä. Testaustiimin kautta voidaan osaamisen jakamisen lisäksi ohjata resursseja kriittisimpiin projekteihin ja saada tukea oman testausnäkemyksen suorittamisessa projektitiimin kanssa. Riippumattomuuden kautta voidaan taata kriittinen tarkastelunäkökulma ja kohdistaa testaajia erikoistumaan erilaisiin testaus-tekniikoihin ja näin laajentaa testausosaamista. Mahdollisuudet testata tuotteita monipuolisemmin kasvavat. Testaajien yksilölliset eroavaisuudet testaustavoissa eivät vaikuta merkittävästi projektissa suoritetun testauksen laatuun, kun testaajia voidaan kohdistaa eri testausosaamisalueille heidän vahvimpien osaamistensa mukaisesti. Toki testaustiimissä ja useiden testaajien käyttö projekteissa aiheuttaa järjestelmätuntemuksen jakamisen muodostuvan haasteelliseksi. Se osoittaa, että kehittäjien ja testaajien vuorovaikutustaidot nousevat merkittävään rooliin työstä suoriutumiseksi.

Oli käytössä yksittäinen testaaja tai testaustiimi, on testauksessa hyvä noudattaa selkeää testausprosessia: määrittele, suunnittele, testaa, raportoi ja arvioi prosessia ja suoriutumista koko ajan. Työtä voidaan tehdä toimintokohtaisissa sykleissä, jolloin tuetaan iteratiivista kehitystä. Testaustiimissä testauspäällikön tehtävänä on päivittäin arvioida testauksen etenemistä, ennakoida tulevaa ja mitata testauksen kykyä suoriutua sille asetetuista haasteista. Testausprosessin kannalta on merkityksellistä saada suunniteltua mahdollisimman hyvin asiakkaan ympäristöä vastaava testausympäristö, pohtia kaikki käyttöjärjestelmät ja sovellukset, joita vasten testaus tulisi suorittaa. Yhteinen prosessi ja sitä vasten määritellyt yhteiset työskentelymenetelmät ja toimintatavat tukevat työtä. Näin voidaan varmistua myös siitä, että testaajasta riippumatta työ suoritetaan aina samalla tavoin. Projektiryhmät voivat olla varmoja saamistaan tuotoksista. Prosessin läpivienti vaatii suurta vuorovaikutusta projektiryhmän ja testaustiimin

välillä. Tämän vuorovaikutuksen onnistuminen edellyttää molemmilta osapuolilta aktiivista roolia. Usein nähdään projektiryhmiä, joissa oletetaan testaajan etsivän ja poimivan tietoja kaikesta puhutusta. Vastaavasti testaaja olettaa kehittäjien toimittavan kaiken tarvittavan materiaalin esimerkiksi testitapausten laatimiseksi.

Näin ei kuitenkaan voi olla. Todellisuudessa molempien tulisi työtään tehdessä huomioida toinen osapuoli. Kehittäjien toimittaa testaajille materiaalia sitä mukaa, kun tieto toteutustavoista saadaan suunniteltua. Vastaavasti testaajan on tarpeen käydä keskustelua suunnitelmasta kehittäjien kanssa ennen testitapausten valmiiksi saattamista. Vain vuorovaikutuksen keinoin voidaan varmistua, että molemmilla osapuolilla on yhteinen käsitys toteutettavasta toiminnosta. Tätä helpotetaan suunnitelmiin kohdistuvalla staattisella testauksella. Monissa yrityksissä on testauksen dokumentaatiosta joko luovuttu tai sitä on kevennetty. Usein nämä yritykset ovat kuitenkin sellaisia, joille ei ole vahvoja asiakasvaatimuksia ja selkeitä vaatimuksia työskentelylle tai toteutetuille ratkaisutavoille. Dokumentaation merkitys on ilmeinen, kun testauksen jäljitettävyyden on varmistettava aina asiakasvaatimuksista testituloksiin asti. Myös testitapausten uudelleen käytön kannalta on merkityksellistä kirjata testattavat asiat. Usein projektit ovat toiminnoiltaan vaativia ja haasteellisia. Tällöin testitapausten tarkka suunnittelu auttaa testaajaa hahmottamaan toimintoihin liittyvät poikkeustapaukset. Lisäksi ymmärrys siitä, että mihin kaikkeen toiminto vaikuttaa, on tarpeellista testaussuunnittelua tehtäessä.

Testaus suorituksen yhteydessä testaajaa helpottaa, kun hänen käytettävissään on selkeä tieto suoritettavista testeistä. Testausta hidastaa merkittävästi se, että testaaja alkaisi testaustilanteessa palauttamaan mieleensä toiminnon asiakasvaatimusten moninaisuutta ja testaustarpeita. Selkeällä testaussuunnittelulla voidaan myös mahdollistaa useiden testaajien käyttö projektin testauksessa, sekä tarvittaessa korvata testaaja esimerkiksi sairasloman ajaksi. Näin voidaan varmistua siitä, että projektin eteneminen ei esty testaajan poissaolon vuoksi. Kokenut testaaja kykenee testauksen suorituksen yhteydessä luovaan testaus-

työhön suunnitellun testauksen lisäksi. Tällöin voidaan löytää uusia keinoja testata järjestelmää, kehittää testitapauksia ja toisaalta löytää uusia vikoja järjestelmästä. Testitapausten kuvaaminen siten, että testitapauksissa ei määritellä esimerkiksi tarkkoja syötteitä, mahdollistaa testien suorittamisen laajemmilla arvoalueilla. Näin voidaan muuntaa testauksen suoritusta eri testikierröksillä testattavien asioiden pysyessä kuitenkin vakiona luovan osuuden täydentäessä testausta.

Testauksen hallinta edellyttää työskentelyn jatkuvaa seuranta ja arviointia niin testaustiimissä, kuin projektirajapinnoissa. On suoritettava jatkuvaa riskiarviointia, jotta voidaan priorisoida projektit keskenään. Tehtävät työt on priorisoitava projekteissa siten, että kriittisin projekti asetetaan aina etusijalle testauksessa. Tämä priorisointi vaatii ymmärrystä projektiryhmiltä siihen, että testausresurssien jako on suoritettava koko tiimiä ja tiimin tulevaisuutta silmällä pitäen. Projektiryhmien ymmärrystä voidaan lisätä sillä, että heille kuvataan selkeästi testauksen tehtävät ja testaajien jakoperusteet projekteihin. Lisäksi heillä on oltava nähtävillään ajantasainen kuva testauksen kuormituksesta, sekä realistinen ymmärrys testaustyöhön tarvittavasta ajasta. Tämän tarkastelutiedon antaminen on testauspäällikön vastuulla, jonka tehtävänä on testaustyön organisointi ja priorisointi.

Testauksen perustehtävä ja päivittäinen työskentely

Testauksen tilannekuvan rakentamisesta ja kuvaamisesta suoriutuakseen on testauspäälliköllä, testaajilla, projektiryhmien jäsenillä ja muilla tiimin jäsenillä oltava yhtenevä kuva testauksen perustehtävästä, sekä kunkin osapuolen omasta perustehtävästä. Lisäksi nämä tehtävät on limitettävä keskenään siten, että kaikki testaukseen liittyvä työ on osoitettu kuuluvaksi joko testaustiimille tai projektiryhmälle. Käytännössä tällainen jako voidaan yleisellä tasolla kuvata siten, että yksikkö- ja integrointitestaus ovat projektiryhmien vastuulla ja järjestelmättestaus puolestaan testaustiimin vastuulla. Lisäksi on sovittava yhteisesti työskentelystä integrointi- ja järjestelmättestauksen rajapinnassa siten, että mo-

lemmilla on selkeä kuva toisen ryhmän suorittaman työn laajuudesta ja syvyydestä.

Testauksen perustehtävänä on näin ollen järjestelmätestaus, johon kuuluu testausprosessissa määritellyt toiminnot. Testausprosessin osalta on tärkeää varmistaa, että prosessi noudattaa yrityksen toimintajärjestelmää. Samalla on kaettava testausstandardien mukaisen testausprosessin osa-alueet. Jokaiselle tiimissä toimijalle on tärkeää kuvata toiminnot ja niiden suorittamiseen liittyvät tarpeet. On oltava yksiselitteisesti esitellyt tiedot, joita testaajat tarvitsevat projektiryhmiltä suunnitelleessaan testausta. Lisäksi testaajien on ymmärrettävä kohdat, joissa heiltä on löydyttävä ammattitaitoa antaa palautetta kehittäjien työhön staattisen tai dynaamisen testauksen muodossa. Testaustiimin sulavan toiminnan kannalta ja ongelmien ehkäisyn kannalta on oleellista kuvata menettelytavat esimerkiksi wiki-tyyppisesti. Näin jokainen tiimissä toimiva voi palauttaa asiat mieleen tarkastelemalla menettely- ja työskentelyohjeistusta. Kun menettelyt ovat projektista riippumatta samat, on testaajien helppo suorittaa työtä kaikissa projekteissa. Tällöin suurin panostus testaajan osalta tehdään järjestelmätuntemuksen kasvattamiseen työskentelytapojen opettelemisen sijaan. Toisaalta projektit saavat tasapuolisen kohtelun testaustiimistä. Tuntiessaan testaajien käyttämät toimintamallit, on projektiryhmien helppo tietää testaajien odotukset heidän suuntaan. Samoin ymmärrys testaajien tuottamasta materiaalista helpottuu.

Onnistuakseen perustehtävän suorittamisessa on testauspäällikön ajoittain varmistettava niin testaustiimin kuin projektiryhmien ymmärrys testaajien perustehtävästä ja sen suorittamistavoista. Tiimin toiminnan kannalta on erittäin merkityksellistä, että tiimin jäsenten välille muodostuu avoin ja luottavainen vuorovaikutus. Jokainen tiimin jäsen ymmärtää palautteen positiivisena ja tiimin yhteisen tehtävän suorittamisen kannalta oleellisena. Yhteisten pelisääntöjen muodostamisen ja avoimen töiden järjestelyn lisäksi testauspäällikön on kyettävä toimimaan esimerkkinä testaajille ja opastamaan heitä työssään omalla toiminnallaan. Myös läsnäolon ja vuorovaikutuksen korostamisen merkitys on suu-

ri perustehtävässä onnistumisen suhteen. Testauspäälliköllä ei ole kuitenkaan mahdollisuutta onnistua työssään yksin, vaan hänen on saatava sitoutunut tuki työlleen tiimin esimiehiltä, sekä projektipäälliköiltä. Selkeä tuki, roolit ja vastuut on kuvattava niin ryhmien kesken, kuin testaustiimin sisällä, jotta mahdollistetaan työskentelyn ilman epäselvyyksiä.

Testaustiimissä saattaa olla testaaajia, jotka on resursoitu tiimin käyttöön vain osaksi aikaa. Aika saattaa olla rajattu esimerkiksi vuodeksi tai siten, että henkilö on käytettävissä esimerkiksi 70 % ajastaan testaustiimissä ja muun ajan hän työskentelee jossakin muussa tiimissä. Tällaiset järjestelyt pitäisi määritellä selkeästi siten, että niin testauspäällikkö, itse testaaaja kuin muut testaaajat ja projektiryhmät tietävät asiasta. Selkeästi kuvatut roolit tiimin sisällä ja vastuiden tarkka kuvaaminen helpottavat tiimin jäsenten työskentelyä keskenään. Näin saadaan myös vältettyä väärinymmärryksiä. Myös omien johtopäätösten luominen esimerkiksi siitä, että miksi joillekin työntekijöille on tiimissä sallittu tiimin ulkopuolisten töiden vastaanotto, kun toisten on noudatettava testauspäällikön jakamia tehtäviä. Avoimuuden kautta luodaan tiimiin rauhallisuutta, mahdollistetaan työskentelyn eteneminen ja tiimin sisäinen sitoutuminen kasvaa.

Testauksen perustehtävän suorittaminen vaatii testaaajalta paljon. Hänen on ymmärrettävä testaus kokonaisuutena järjestelmän laatuominaisuuksien joukossa. Testaaajan on kyettävä pohtimaan, miten koko järjestelmä toimii. Näin hän kykenee löytämään toimintojen suorittamistavat ja virhetilanteet. Keskittyminen yhteen asiaan kerrallaan siten, että löytää mahdollisimman paljon vikoja, on vaativaa. Hänen on myös omalla toiminnallaan kyettävä löytämään viestintätavat ja erilaiset variaatiot ohjelmistosta. Lisäksi on osattava huomioida esitystapojen kautta käytettävyydeltään toimivimmat ratkaisut juuri tiettyjen käyttäjien yhteydessä. Usein ohjelmistoissa on oma kielimaailmansa, jolloin ohjelmiston on noudatettava käyttäjän käsitteistöä. Tämän lisäksi on pohdittava, onko jo jossakin muussa tuotteessa mahdollisesti testattu samaa asiaa. Toisaalta on arvioitava voisiko näistä testitapauksista luoda uudelleen käytettävät tai jopa yleiskäyttöiset testitapaukset.

Tuotenäkökulman lisäksi on testaajan kyettävä toimimaan riippumattomana siten, että testauksen ulkoiset näkökulmat eivät vaikuta järjestelmästä tehtyjen vikahavaintojen määrään. Toki koko tiimin tulee ymmärtää se, että vain testajat eivät testaa tuotetta. Myös kehittäjien on suoritettava yksikkö- ja integrointitestausta, jotta järjestelmän laatua voitaisiin parantaa. Testaajat tarvitsevat testitapausten laadinnan tueksi kehittäjiä, jotta voivat varmistaa oman ymmärryksensä testattavista asioista. Testaajan on oltava myös tietoinen oman osaamisensa rajoista ja ymmärrettävä pyytää apua toisilta testaajilta sekä tietenkin kehittäjiltä. Kokemuksen karttumisen myötä myös osaaminen kasvaa. Lisäksi on oltava ymmärrys siitä, että kaikkea ei voi testata ja toisaalta on oltava valmis auttamaan toisia testaajia. Ongelmista on kyettävä avoimesti keskustelemaan ja olemaan avuksi niitä ratkaistaessa. Testaajien on opittava myös jakamaan omaa osaamistaan samoin kuin kehitettävä vuorovaikutustaitojaan aktiivisesti. Näillä keinoin perustehtävän suorittaminen on helpompaa ja testaaja kykenee selkeämmin pysymään perustehtävän suorittamisessa.

Uusilta työntekijöiltä oletetaan usein vankkaa ja vahvaa kokemusta, kun heidät palkataan testaustiimiin. Kun puhutaan keskiverto työntekijän palkkaamisesta, on tilanne usein se, että työntekijäksi riittää opiskelija tai juuri koulusta valmistunut. Työkokemusta on vähän ja tuoreet opinnot taustalla. Työntekijä itse kokee usein hallitsevansa tehtävät johtuen korkeasta koulutuksen tasostaan tai muutamana vuoden työkokemuksen perusteella. Uusi työntekijä ei rekrytointiprosessin jälkeen kuitenkaan aina ymmärrä testauksen perustehtävän laajuutta koko testausprosessin näkökulmasta. Tällöin on myös mahdotonta ymmärtää työn merkitystä ja työnantajan odotuksia. Palkattavien testaajien osalta on hyvin tärkeää, että rekrytointiprosessiin osallistuu myös testauspäällikkö, jolla on selkein kuva tiimin resurssitarpeista ja osaamistarpeista. Näin uudelle työntekijälle voidaan kuvata tuleva työ jo rekrytointivaiheessa selkeästi ja hänen roolinsa testaustiimissä on yksiselitteinen kaikkien rekrytointiprosessiin osallistuvien kesken. Myös vanhojen työntekijöiden varmuus oman työn koskemattomuudesta vahvistuu, kun heille voidaan riittävän ajoissa esitellä uuden tulevan työntekijän rooli tiimissä ja vaikutukset vanhojen työntekijöiden rooleihin.

Esimiestyö osana päivittäistä työtä

Testaustiimin työskentelyn kannalta ohjaaminen ja osaamisen johtaminen vaatii kehittämistä. Tiimissä on kahdenlaisia esimiehiä: henkilöesimiehiä ja asiaesimiehiä. Testaustiimin testauspäällikkö toimii asiaesimiehen roolissa, jolloin hänellä on vastuu testaustiimin työskentelyn läpiviennistä. Käytettävissä olevilla testaajilla on keskenään eri henkilöesimiehet. Käytännössä työnjako henkilö- ja asiaesimiesten välillä on se, että henkilöesimies huolehtii työntekijän henkilökohtaisesta kehittämisestä, tulevaisuuden suuntautumisesta ja toiveista tulevaisuudessa työn sisällön suhteen. Asiaesimies vastaa päivittäisen työn jakamisesta käytettävissään olevien työntekijöiden kesken. Testaustiimissä testauspäällikkö kartoittaa testaustoimintojen tarpeen (määrittely, suunnittelu, toteutus, raportointi, arviointi, kehittäminen) ja jakaa tehtävät käytettävissä olevan tiimin kesken. Henkilöesimiehet eivät tehtävien jakamiseen osallistu.

Tämä asettaa kehityskeskusteluiden yhteydessä haastetta siten, että henkilöesimiehet huomioivat yksilön kehityssuunnitelmaa laatiessaan testaustiimin kehitystarpeet ja muille testaajille laadittavat kehityssuunnitelmat. Kokonaisuuden havainnointi on tärkeää, jotta kehittäminen tapahtuisi tasapuolisesti yksilötasolla ja testaustyön kehittämistä tukien tiimi- ja organisaatiotasolla. Henkilö- ja asiaesimiesten kesken tulisi luoda kanavat, joiden kautta ennen kehityskeskusteluja luotaisiin yhteisymmärrys organisaation ja tiimin tarpeista kehittämisen suhteen. Tämän jälkeen käytävissä kehityskeskusteluissa koottaisiin yksilöiden toiveet ja näiden perusteella laadittaisiin seuraavan vuoden tavoitteet kehitymiselle. Testauspäällikön kannalta yhteistyö on tärkeää, jotta yksilön ja tiimin näkemykset tulevaisuudesta olisivat yhtenevät. Tällä tavoin välttyttäisiin pettymyksiltä tai ristiriidoilta päivittäisen työn suorittamisen suhteen ja ylläpidettäisiin motivoitunutta tiimiä.

Esimiehillä itsellään ei ole useinkaan riittävästi aikaa levätä tai palautua työn kuormituksesta. Yleensä työpaikoilla on vältettävä tilanteita, joissa esimiestason henkilö toisi tunteensa esille. Hänen tulee olla rauhallinen, kaiken kestävä ja

käytännössä kylmä muuri. Ihmisten tulisi oppia tuntemaan oma stressinsietokykynsä ja opetella hallitsemaan sitä. Samalla on kuitenkin tärkeää muistaa, että haastava työ jatkuvassa kiireessä on raskasta ja tämän vuoksi esimiehelle pitäisi sallia mahdollisuus näyttää omia tunteitaan tietyissä rajoissa. Haastetta antaa se, että esimiehen on kyettävä suoriutumaan esimiestehtävien lisäksi varsinaisesta perustyöstään. Esimiestyötä ei pidetä perustehtävänä. Tehtäviin saatetaan ottaa myös sellaisia henkilöitä, jotka ovat korkeasti koulutettuja, mutta heillä ei välttämättä ole johtamistason koulutusta. Vastaavasti sellainen ilman korkeakoulututkintoa oleva työntekijä, joka soveltuisi tehtävään, saattaa jäädä ilman esimiestyötä. Toisaalta tavallisena testaajana saattaa toimia korkeasti koulutettu työntekijä, jonka siirtäminen tehtävästä vaativampaan koetaan tarpeellisena sen vuoksi, että hänen koulutustasonsa on korkea.

Vieläkin todella harvoin kohdataan työntekijöitä, joilla olisi testaukseen suuntautunut koulutus. Pääsääntöisesti koulutusrakenteessa on vain muutama kurssi liittyen testaukseen, jolloin varsinainen testausosaaminen ei ole mahdollistaakaan hankkia muuta kuin työtä tehden. Testausta pidetään helposti jokaisen osaamana tehtävänä, jonka laaja-alaisuutta ja osaamisten laajuutta ei tiedosteta. Sen sijaan, että yrityksissä keskityttäisiin korkeasti koulutettujen osaajien ylentämiseen esimerkiksi esimieheksi, tulisi sen sijaan kehittää heidän osaamistaan erityisasiantuntijoina. Heitä tulisi palkita heidän osaamisensa kautta, vaikka työntekijän tittelinä edelleen olisikin testaaja. Työntekijän omanarvon tunnetta kohottaakseen voidaan toki testaaja nimikkeen yhteyteen liittää myös erikoisosaamiseen viittaavia nimiä, kuten testaaja, erikoistunut käytettävyyteen.

Testauspäällikön tulee olla asiaesimiehenä tietoisesti läsnä tiimilleen, samoin kuin testaajien henkilöesimiesten tulee olla testaajille ja testauspäällikölle. Tietoisien läsnäolon vaikutus johtajuuteen ja työntekijöiden ohjaamiseen on todettu vaikuttavan merkittävästi koko tiimin työskentelyyn. Jos esimies tuntee olonsa energiseksi, asenne on kohdallaan ja hän hyväksyy omat tunteensa esimerkiksi ilon, pelon ja harmistuksen, on hänen helpompi ohjata muita ihmisiä ja olla esimerkkinä muille. Tällainen eläminen hetkessä ja samalla tuntosarvet koholla

aistimassa oman tiimin tilaa, antaa vahvuuden ja vakaan asiantuntemuksen tunteen muille ympärillä oleville. Käytännössä oma rauhallisuus on aistittavissa ja levitettävissä muillekin tiimin jäsenille.

Kun itseltä löytyy kykyä johtaa itseään, voi johtaa myös muita. Kun testauspäällikkö uskoo omaan kehittämiseensä, voi hän nähdä kehittymismahdollisuuksia myös muissa. Työssä onnistuminen edellyttää, että testauspäällikkö saa tukea työhönsä omilta esimiehiltään ja tiimin vetäjiltä. Useissa lähteissä puhutaan uudistavasta johtamistyylistä, jolla pitäisi pyrkiä vapauttamaan organisaation resurssit luovuuteen ja hyvään tehokkuuteen. Tämä on toivottava kehityssuunta myös testaustiimin osalta. Kun tiimin jäsenet kykenevät tehostetusti suoriutumaan perustehtävästään, saadaan tiimissä irrotettua testaaajien työaikaa erityisesti luovuuteen. Testaustyö on erittäin luovaa työtä, kun testaaajalle annetaan vapaat kädet toimia testattavan sovelluksen kanssa. Vuorovaikutustaitojen parantuminen entisestään mahdollistaa testausmateriaalin uudelleenkäytön ja näin ollen vapauttaa aikaa testaustyön laajentamiseen ja monipuolistamiseen.

Kanban-menetelmän hyödynnettävyys

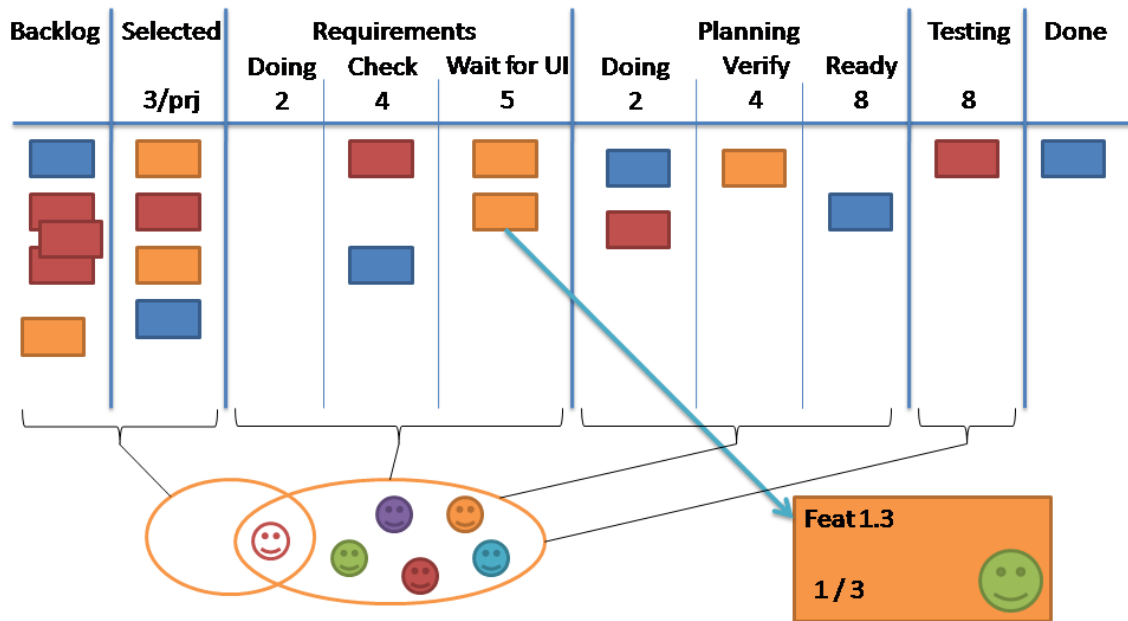
Testaustiimissä on hyödynnetty eri menetelmiä projektien läpiviemiseksi. Projektiryhmät hyödyntävät RUP:n, XP:een ja Scrum-menetelmien hyväksi havaittuja käytäntöjä. Myös testaustiimi on sitoutunut noudattamaan RUP:n menetelmiä, mutta muiden menetelmien käytön suhteen on ollut havaittavissa ongelmia. Testaustiimin toiminta eroaa merkittävästi projektiryhmien toiminnasta johdun useiden projektien hallinnasta ja näkökulmien eroavaisuudesta. Tämän vuoksi tiimin toiminnan kannalta on etsittävä soveltuvimmat menetelmät testaustiimin sujuvan ja avoimen vuorovaikutuksellisen toiminnan kehittämiseksi ja ylläpitämiseksi.

Kanban-menetelmä on menetelmä, joka on joustava ja mukautettavissa testauksen toimintamalliin. Menetelmään voidaan ottaa palasia muista menetelmistä. Palaset voidaan kohdistaa niille osa-alueille, joihin tukea kaivataan. Toisaalta

voidaan säilyttää vapaus toimia testaustiimin tarvitseminen menetelmin. Erityisesti päivittäisen työskentelyn kannalta Kanban lisää avoimuutta, vuorovaikutusta ja paljastaa ongelmakohdat toiminnassa. Testaustiimin sisällä voidaan helposti tuoda esiin hankaluudet testaustehtävien suorittamisessa ja helpommin pyytää toisilta testaajilta apua tehtävän suorittamiseen. Lisäksi hankaluuksien analysoinnin kautta on mahdollista kuvata tarpeita paremmin ja kehittää vuorovaikutusta testaustiimin sisällä ja projektiryhmien kanssa.

Kanban-taulun käyttö on mahdollista muuttaa perinteisen seinätaulun sijaan sähköiseksi. Tällaisen version käytöllä voidaan toiminta testaustiimissä tehdä avoimeksi projektiryhmille. Samalla mahdollistetaan koko testauskuorman näkeminen ja avoimuus. Ymmärrys toisten projektien tarpeista saadaan myös lähemmäksi projekteja. Tilanne on ajan tasalla ja kaikkien osapuolien saatavilla aina tarvittaessa. Sen sijaan, että projektiryhmät sprintin suunnittelupalaverissa arvailisivat testausresurssien käyttöä yksin, yhdessä testaajan tai testauspäällikön kanssa, on heidän mahdollista tarkastella koko testauskuorman vaikutusta seuraavan sprintin aikana.

Tämä tieto auttaa sprintin suunnittelussa ja testausajanjaksojen suunnittelussa. Sen sijaan, että järjestelmätestaus olisi yksinomaan projektilähtöistä, saataisiin suunnattua projektityötä testauslähtöisempään suuntaan. Testauslähtöinen tarkastelutapa on tarpeen kaikilla osa-alueilla, niin yksikkö-, integrointi- kuin järjestelmätestauksen osalta. Myös staattisen testauksen aikataulutaminen ja testaustiimin suoriutuminen tiimille asetetuista tehtävistä helpottuu, kun koko tiimi kykenee huomioimaan työskentelyssään myös testaustiimin mahdollisuudet. Oheisessa kuvassa, Kuva 15, on esitetty versio Kanban-seinätaulusta, jota testaustiimissä ehdotetaan käytettäväksi.



Kuva 15 Kanban-ehdotus

Testaustiimin Kanban-taulussa on kuusi osa-aluetta. Backlog koostuu kaikista testaukselle osoitetuista tehtävistä. Niitä lisätään tauluun sitä mukaa, kun tehtäviä tulee testauspäällikön tietoon. Testauspäällikön suorittaman priorisoinnin kautta valikoidaan Selected-sarakkeeseen tärkeimmät tehtävät. Niistä testaajat yhteisymmärryksessä testauspäällikön kanssa valikoivat ominaisuuden, jolle pitäisi suorittaa testaus suunnittelu ja testaus. Aluksi määritellään hyväksymiskriteerit siten, että testaaja Doing-sarakkeessa arvioi vaatimukset ja laatii todennettaville asioille hyväksyntäkriteerit. Seuraavana on vuorossa kriteerien tarkastus (Check) ja tämän jälkeen jäädään tehtävän osalta odottamaan käyttöliittymäsuunnitelmaa (Waiting for UI). Staattisen testauksen ja käyttöliittymän vahvistumisen perusteella siirretään tehtävä testitapausten suunnitteluvaiheeseen (Planning/Doing), tarkastukseen (Verify) ja odottamaan dynaamista testausta (Ready). Testauksen suorittamisen (Testing) jälkeen toiminnon käsittely on valmis (Done).

Kanban-taululla seurataan tarkkaan ominaisuuksien testaukseen liittyviä vaiheita johtuen siitä, että testauksen perustehtävänä on toimintojen testauksen

suunnittelu ja testaaminen. Testaustiimillä on toki myös tehtäviä, jotka poikkeavat ominaisuuksien läpikäymästä prosessista. Esimerkiksi testausympäristöjen ylläpitoon liittyvien tehtävien yhteydessä taulun käyttö ei onnistu samalla tavoin. Testaustiimin toiminnassa sallitaankin taulun mukautettu käyttö tällaisten tehtävien osalta. Ominaisuuksien testauksesta poikkeavien tehtävien siirto on sallittua sarakkeiden yli siirryttäessä taululla vasemmalta oikealle. Tehtävien palauttaminen takaisinpäin ei ole mahdollista. Jos havaitaan jollekin tehtävälle lisämäärittelyn tarve, laaditaan lisämäärittelyä varten uusi lappu. Tehtävä priorisoidaan suoritettavaksi testauspäällikön toimesta.

Aluksi työskentelyssä edetään siten, että hahmotetaan ensimmäisten viikkojen ajan taulun eri sarakkeiden käyttöaste. Tämän perusteella etsitään vähitellen sopivat WIP-arvot eri vaiheisiin, jotta työskentelyn arviointi olisi mahdollista. Samalla arvioidaan onko tiimin toiminnan kannalta tarpeen ottaa uusia sarakkeita käyttöön tai poistaa joitakin ehdotettuja sarakkeita. WIP-arvoja määriteltäessä on tärkeää huomioida määritelläänkö arvo testaustiimikohtaisesti vai projekti-kohtaisesti. Vaihtoehtojen testaaminen käytännössä on tarpeen, jotta testauksen tehokkaampi läpivienti mahdollistuisi. Tehtäväkortteihin merkitään tehtävää kuvaava tieto, arvioitu kesto, jäljellä oleva kesto ja tehtävää suorittava testaaja. Taululla Backlog, Selected, Waiting for UI ja Ready -sarakkeet ovat sellaisia, joissa tehtävä ei ole osoitettu kenellekään testaajalle. Näin ollen priorisointia tehtäessä on huomioitava projektien toimitusaikataulut, kuormitukset eri sarakkeissa ja vapaana olevien testaajien osaamisalueet tehtäviä otettaessa jatkokeskittelyyn. Oleellista on, että yhdellä resurssilla on työn alla kerrallaan vain yksi tehtävä.

Testaustiimissä on tällä hetkellä jäseniä kolmessa roolissa: testauspäällikkö, testaaja ja laboratorioinsinööri, jonka tehtäviin kuuluu testausympäristöjen koaminen ja ylläpito. Kanban-tilun osalta testauspäällikön tehtäviin kuuluu backlogin ja selected-sarakkeiden ylläpito ja priorisointi. Lisäksi testauspäällikön tehtäviin kuuluu luoda, valvoa ja ylläpitää mittareita läpimenoaikojen seuranta varten. Hänen tulee myös huomioida tulokset tehtävien priorisoinnissa ja testa-

uksen kehittämisesä. Kaikkien tiimin jäsenten tehtäviin kuuluu muiden sarakkeiden käyttö omien tehtäviensä osalta. Lisäksi tulee tarjota avoimesti apuaan ongelmakohtien ratkaisuun oman tehtävän valmistuttua ennen kuin testaaja valitsee uuden tehtävän suoritukseen. Toiminnalla pyritään mahdollistamaan se, että testaajat avoimemmin liikkuisivat eri projekteissa, kehittyisivät testauksen eri osa-alueilla ja toisaalta oppisivat jakamaan osaamistaan keskenään. Vuorovaikutuksen ja itsenäisemmän valinnan kautta pyritään kohottamaan motivaatiota ja sitoutumista työhön yhtenäisenä tiiminä.

3.2 Testauksen osaamisen johtaminen

Testaustiimissä on käytössä selkeä testausprosessi ja työskentelymenetelmät, mutta ydinosaamisia, osaamisalueita ja osaamisia ei testauksen osalta ole yksilöity. Kun tiedetään kriittiset osaamistarpeet, voidaan arvioida tiimin kehitystarpeet. Sitä kautta voidaan muodostaa testaukselle dokumentoitu kehityssuunnitelma. Testauksen osaamiseen liittyvien asioiden keskustelu ainoastaan henkilökohtaisissa kehityskeskusteluissa henkilöesimiehen kanssa ei riitä. Näiden keskustelujen yhteydessä täytetyt lomakkeet pitävät sisällään osaamisia, jotka henkilöesimies kokoaa itsenäisesti ja tarkistaa testaajan kanssa. Tarkasteltaessa vastaavaa asiaa koko organisaation kannalta on toimintajärjestelmässä testaus vahvasti esillä ja korostuu prosesseja kuvattaessa. Testaustiimin osalta on arvioitava osaamisen nykytila. Tarvitaan tiimin oma kartoitus osaamisalueista ja osaamisista. Vasta näiden toimenpiteiden jälkeen voidaan arvioida, löytyykö testaustiimin nykyisistä testaajista kykyä täyttää puuttuvat osaamiset esimerkiksi lisäkoulutuksen avulla vai onko tarvetta uusille rekrytoinneille.

Testauspäällikkö seuraa tilannetta koko testaustiimin kannalta nähden kaikki testaajat työssään ja kaikki projektit rinnakkain. Vastaavasti henkilöesimies saa hyvin arvokasta palautetta yksittäiseltä testaajalta. Hän ei kuitenkaan ole kaikkien testaajien henkilöesimies, eikä näe koko testauksen tilannetta kaikkien projektien näkökulmasta. On erittäin haasteellista, että kaikki tiimin testaajat saavat

tasapuolisen kohtelun esimerkiksi koulutuksiin pääsemisen suhteen. Kommunikaatio vaatii kaikilta esimiesasemassa työskenteleviltä kehittymistä ja toiminnan linjaamista siten, että testauksen tilanne ja tarpeet tiedostetaan kehityskeskusteluissa. Lisäksi on huomioitava kokonaisuus ja testaustiimin mahdollisuudet joustaa yksittäisen testaajan toiveita täyttäen. Oikeudenmukainen kohtelu edellyttää sitä, että kartoitetaan osaamiset. Niiden pohjalta voidaan laatia selkeän testauksen kehityssuunnitelman ja avata keskustelu. Aineiston kokoaminen on haasteellista, koska kartoituksessa ei voida hyödyntää toisten testaustiimien kartoituksia. Työn suorittamiseksi on löydettävä tiimistä kootun tiedon lisäksi myös muita keinoja.

Tiedon tasolla testaustiimissä on panostettava voimakkaasti tiedon jakamiseen testaajien kesken. Tietoa on varastoitava sellaisessa muodossa ja sellaisiin paikkoihin, joissa tieto on kaikkien sitä tarvitsevien saatavilla helposti. Tiedon luominen vaatii ideointia ryhmänä ja tehtävänkierron hyödyntäminenkin voi tiettyillä testauksen osa-alueilla olla hyödyllistä. Tiedon hankinta on tiimissä pääosin projektien varassa. On olemassa projektiryhmiä, jotka ottavat testaajat tiiminä vastaan avoimesti ja sitä myöden jakavat tietoa toteutettavista tuotteista ja niiden käyttötarkoituksesta hyvin. Vastaavasti löytyy projekteja, jotka kokevat testaajat edelleen ulkopuolisina henkilöinä, jolloin tiedon jakaminen unohtuu. Tiedonhankintaan vaikuttaa hyvin paljon henkilöiden asenne eri asiantuntijaryhmiä kohtaan.

Uusien ja tehokkaampien keinojen löytäminen on tarpeen tiedonjakoon ja siihen, että jokainen varastoisii tietoa samalla tavoin keskenään. Tällä keinoin voidaan varmistaa, että jokainen voi helposti saada käsiinsä tietoa. Eikä kaikkien testaajien tarvitse itse etsiä tietoa tai opetella asiaa alusta asti itsenäisesti. Usein testaajien taustat ovat erilaisia. Se voidaan kokea haasteena, mutta myös rikkautena. Testaajien erilaisuus mahdollistaa erilaisten testaustapojen sisällyttämisen osaksi testaajien osaamisen kehittämistä. Lisäksi erilaisten persoonallisuuksien tunnistaminen opettaa kohtaamaan testaajat yksilöinä ja luomaan luotavaisemmat suhteen testaajien kanssa. Tämän ymmärryksen kautta heitä voi

ohjata varastoimaan ja jakamaan tietoa muidenkin testaajien hyödynnettäväksi, sekä hyödynnettäväksi koko tiimin toiminnassa.

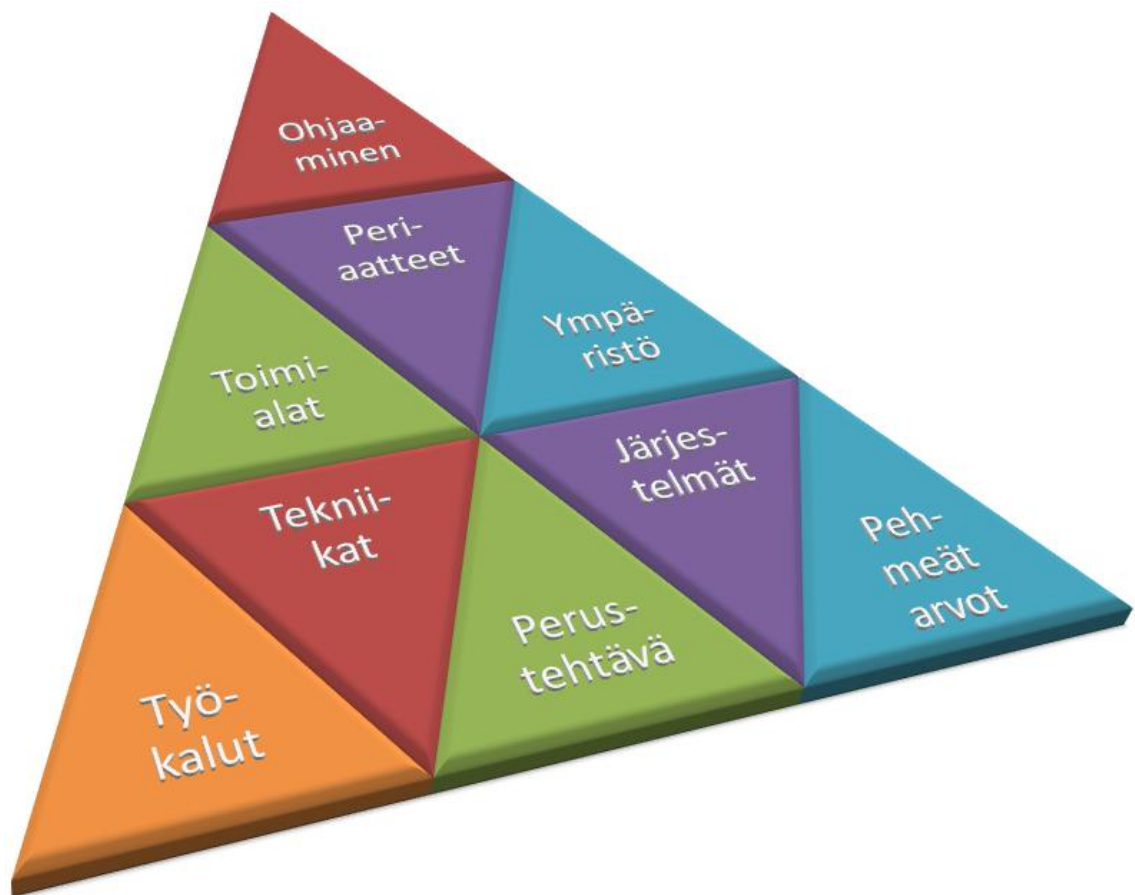
Niin uuden kuin vanhan työntekijän asenne ja tyyppi oppijana ratkaisee, kuinka hyvin tiedon jakaminen, varastointi, perustehtävän suorittaminen ja osaamisen kehittäminen onnistuu. Usein vanhempien työntekijöiden osalta on vanhojen kokemusten perusteella juurtunut tapoja ajatella tiimissä tapahtuvia toimintoja tietyllä tavalla. Usein jopa uusimmat työntekijät, jotka vain esimerkiksi vuosi sitten palkattiin yritykseen innokkaina ja muutoshakuisina, kokevat turhana muutoksen. On selvää, että ryhmään sopeutuminen tapahtuu usein jopa huomaamatta ja monesti koetaan suurten muutosten tapahtuvan yrityksessä liian hitaasti. Tällöin työskentely heikoilla tai epämuukavilla tietojärjestelmillä tai työskentelytavoilla pitkään aiheuttaa turhautumista ja alkuinnostuksen loppaamista.

Testaustiimissä tapahtuvan muutoksen ja kehittämisen yhteydessä on muistutettava muutoksen kohtaavia muutoksen syistä, tarpeista ja nykyisistä hankalista tavoista työskennellä, sekä tulevan muutoksen hyödyistä. Uusimmat muutokset, jotka koskevat koko organisaatiota on pohdittava tarkkaan ja muutosten läpivientiä sekä esittäminen henkilökunnalle perustelujen kera on oltava suunniteltu jo alkuvaiheesta lähtien. Näkökulma muutokseen on käytännön helpottaminen ja vakuuttavasti harkittu suunta. Muutoksen yhteydessä on pyrittävä toteuttamaan muutokset siten, että huolet ja pelot liittyen muutokseen eivät realisoiduisi. Muutoksista informointi on huolehdittava sekä testaajien että projektiryhmän kanssa, jotta ymmärrys säilyisi kaikilla osapuolilla yhtenevänä. Vastaavaa toimintaa edellytetään projektiryhmiltä testaustiimin suuntaan muutostilanteissa. Ennen suunnitelman täytäntöönpanoa on varmistettava, että muutos soveltuu myös testaustiimin toimintaan, eikä haittaa muiden projektien etenemistä.

Yksilöiden muutoshalukkuuden ja tiimin yhtenäisen toiminnan lisäksi on ymmärrettävä eri testaustekniikat, niiden vaatimukset testaajien osaamiselle, sekä testauksen moninaisuuden haasteet testaajille. Tekniikoita on lukuisia ja on täysin mahdotonta löytää testaajia, jotka hallitsisivat kaikki testaustekniikat perusteelli-

sesti. Tekniikoiden opiskelun ja osaamislaajuuden kasvattamisen osalta on testaustiimin kannalta merkityksellistä jakaa erityyppisiä tekniikoita eri testaajille. Jaon perusteena voidaan käyttää projektien tarpeita ja yksilön kiinnostuneisuutta kyseistä tekniikkaa kohtaan. Tällä tavoin voidaan vahvistaa tiimin osaamista ja yksilön keinoja oppia jakamaan hyvin hallitsemaansa osaamista toisten testaajien kanssa tai tukemaan heitä työstä suoriutuakseen.

Seuraavaan kuvaan (Kuva 16) on koottu testauksen osaamisen osa-alueet perustuen testauksen ydintehtävään varmistaa tuotteelle asetettujen vaatimusten täyttäminen, käyttötarkoituksen varmistaminen ja vikojen löytäminen, sekä vuorovaikutustaitoihin projektiryhmien kanssa.



Kuva 16 Testauksen osaamisalueet

Testauksen osalta osaamisalueita on muodostettu yhdeksän. Testauksen periaatteet pitävät sisällään erilaisia osaamisia, joita testaajalla tai testauspäälliköllä tulisi olla riippuen hänen roolistaan testaustiimissä. Osaamisia ovat muun muassa testauksen tarkoituksen ja testauksen seitsemän periaatteen ymmärtämisen testausstandardien mukaisesti. Ajatuksena on, että jokaisen testaustyötä suorittavan pitäisi ymmärtää työn tarkoitus, perusteet työn teolle ja tavoitteet testauksen tuloksille. Erittäin tärkeänä kuuluvat testausstandardien kuvaaman ja yrityksen käytössä olevan testausprosessin ymmärtäminen, sekä menetelmien ja työskentelyohjeiden noudattaminen sovitulla tavalla. Testauspäällikön osalta on tarpeen testauspolitiikan ja -strategian merkityksen ymmärtäminen, laatiminen ja ylläpito, sekä ymmärryksen jakaminen testaustiimille.

Testauksen perustehtävä pitää sisällään testaajan kyvyn arvioida testattavaa kohdetta, testisuunnitelmien merkityksen ymmärtäminen suhteessa testattavaan tuotteeseen, sekä testauspäällikön osalta testianalysointi ja testisuunnitelmien laadinnan. Lisäksi vaiheeseen kuuluvat testitapausten laadinta, testauksen suorittaminen ja tulosten raportointi yrityksen käytössä olevan prosessin ja menetelmien mukaisesti. Testauksen yhteydessä huomioidaan myös vikojen kirjaaminen järjestelmään, vikojen kuvaaminen hallitusti, sekä vikojen käsittelyprosessin merkityksen ymmärtäminen ja noudattaminen. Työnsä tueksi testaajien on hallittava useita erilaisia testaustekniikoita. Niitä ovat staattiset testaus tavat, joilla katselmoidaan ja tarkastetaan tai läpikäydään dokumentaatiota, suunnitelmia ja koodia. Dynaamiset testaus tavat pitävät sisällään puolestaan toiminnallisen ja ei-toiminnallisen testauksen, joita ovat esimerkiksi suorituskyky-, kuormitus-, käytettävyys-, esteettömyys-, savu-, tietoturva-, skaalautuvuus-, regressio-, uudelleentestaus-, ylläpito-, valmistelematon, negatiivinen, volyyymi-, tutkiva ja asennus- ja konfigurointitestaus.

Testausympäristöjen osaaminen on äärimmäisen tärkeää testisuunnitelmien laatimiseksi ja testauksen suorittamiseksi. Ympäristöjen osalta on testaajalla oltava käytössään riittävät kalusteet, ohjelmistot ja laitteet työn suorittamiseksi. Lisäksi osaaminen koneiden virtualisoinnin, levykuvien ja erilaisten käyttöjärjes-

telmien osalta auttaa testauksesta suoriutumista. Laboratorioympäristössä on merkityksellistä kyetä kokoamaan laitteista lähi- ja laajakaistaverkkoja hyödyntäen erilaisia kytkimiä ja reitittäjiä. Näiden osaamisten tueksi on hyvä kyetä asentamaan, konfiguroimaan ja ylläpitämään esimerkiksi nimipalvelimia, sähköpostipalvelimia ja toimialueita. Laite- ja verkko-osaaminen on iso kokonaisuus, josta voidaan pilkkoa lukuisia osaamisia riippuen yrityksen ympäristötarpeista.

Testauksen perustehtävän suorittamista tukemaan on käytettävissä yleensä testaustyökalut. Näiden hallinta on merkittävä osa osaamista. Testauksen suunnittelun, suorittamisen, analysoinnin ja materiaalinhallinnan lisäksi testaus-työtä tukee vikojen ja muutosten hallintaa käytettävät työkalut, sekä testauksen automatisointia varten tarkoitettut työkalut. Automatisoinnin osalta koodausosaaminen nostaa testaajan osaamistasoa merkittävästi. Tällöin testauksessa ei välttämättä tarvitse käyttää hyödyksi kaupallisia automatisointityökaluja tai kehittäjiä työkaluja luomaan, vaan testaustiimi kykenee itsenäisesti kehittämään omaa toimintaansa automatisoimalla esimerkiksi testauksen savutestit.

Testauksen ohjaaminen on testauspäällikön erityisalueita, jonka yhteydessä hänellä on oltava kyky organisoida ja ohjata tiimin työskentelyä. Testauspäällikön on hallittava esimiestaidot ja kyettävä käyttämään sitä tiimin ohjaamisessa. Lisäksi hänen tulee kyetä laatimaan mittareita testausjärjestelmälle ja sen tuotoksille, arvioida ja analysoida mittaustuloksia aktiivisesti projektien edetessä. Testauksen valvonnan lisäksi on hallittava testausprosessin ja menetelmien kehittämiseen liittyvät kehys- ja kehitysmallit, sekä kyettävä tunnistamaan kehityskohteita. Testauspäälliköllä on löydettävä myös kykyä arvioida testausjärjestelmien kehitystarpeita, samoin kuin testaustiimin kehittämistä. Lisäksi osaamista riskianalyyseihin ja tehtävien priorisoinnin suhteen riskianalyysin pohjalta on tarpeen. Myös koulutustarpeiden arviointi testaustiimin osaamista tukemaan sekä osaamistarpeiden arvioinnin suhteen on tärkeää osata.

Testausosaamisen lisäksi merkittäviä osaamisalueita ovat järjestelmä- ja toimialatuntemus. Järjestelmätuntemus koostuu testattavan ohjelmiston vaatimusten

tuntemisesta, tuotteen kulttuurista ja kontekstista, sekä käyttötarkoituksesta. Ymmärrys asiakkaan liiketoiminnasta auttaa ymmärtämään testattavan tuotteen tulevaan käyttötarkoitusta. Kokemus asiakasrajapinnassa toimimisesta lisää osaamista, samoin kuin tuotteen kehittämiseen käytettävän teknologian ymmärtäminen ja osaaminen. Toimialatuntemuksen osalta on hyvä tuntee projektiryhmät ja heidän toimintatavat, sekä erilaiset prosessit tuotteen valmistuksen osalta. Niitä ovat vaatimustenhallinta, suunnittelu ja koodaus. Lisäksi tuntemus teknisen tuen osalta ja yleisesti liiketoiminnasta auttaa ymmärtämään kehitettävää tuotetta ja testauksen kohdetta.

Viimeinen testauksen osaamisalue on pehmeät arvot. Näihin lasketaan testajan tai testauspäällikön kyky toimia vuorovaikutuksessa. Eri vuorovaikutustapoja ovat testausdokumentaatio, katselmointikäytännöt ja niistä saatu palaute, sekä kommunikointikyky. Yhteistyö kehittäjien, toisten testaajien, esimiesten ja projektijohdon kanssa on merkittävä osa suoritusta. Lisäksi pehmeisiin arvoihin voidaan laskea testajan tyyli olla oppijana ja muutoksen sietäjänä. Lisäksi hänen työskentelytehokkuutensa, motivaationsa, tahtonsa, innovatiivisuutensa, arvonsa ja kyky jakaa hiljaista tietoa muulle organisaatiolle voidaan laskea osaamisiksi, joita tarvitaan testauksesta suoriutumiseen menestyksekkäästi.

3.3 Testauksen kehittäminen

Testauksen kehittämiseksi on testaustiimissä otettava käyttöön mittarit, joilla arvioidaan testauksen tehokkuutta, hyötysuhdetta, ennustettavuutta ja läpimenoaikaa. Lisäksi on suunniteltava yhdessä projektiryhmien kanssa mittareita tuotteiden yhteyteen siten, että testauksen edetessä voidaan tehdä arvioita työn etenemisen ja tuotteen laadun suhteen. Mittarien tulee olla yhteneviä projektien kesken ja lisäksi voidaan ottaa käyttöön tuotekohtaisia mittareita, jotta tuotteen laadun arviointi helpottuisi. Mittareiden käyttöä tulee testata ja etsiä juuri tälle testaustiimille sopivimmat mittarit. Myös mittaustulosten tulkinta on oltava yksiselitteistä ja kaikkia osapuolia palveleva. Mittareiden valinta kannattaakin tehdä

vähitellen ottamalla käyttöön mittari kerrallaan ja etsiä tarpeita uusille mittareille tukemaan ja lisäämään käytössä olevien mittareiden kattavuutta. Edellytyksenä mittareiden onnistuneelle käyttöönotolle on, että testauksen tilannekuva on saatu muodostettua. Päivittäisen tilannekuvan avoin jakaminen ja rutinoitunut käyttö mahdollistaa läpimenoajan ja WIP-arvon käytön lisäksi muiden mittareiden valinnan niin testaustiimin etenemistä kuin projektiryhmien työskentelyä tukemaan.

Testauksen kehittämismallit ovat erittäin käyttökelpoisia yrityksessä, johon on jo laadittu testausprosessi. Tällöin valmiit kehittämismallit tukevat testausprosessin ja testauksen eri osa-alueiden kehittämistä edelleen. Näistä erityisesti TMMi ja TPI ovat toimivia (vrt. Taulukko 6) ja niitä voidaan täydentää CTP-menetelmällä. Valmiit mallit sisältävät edellisessä kappaleessa kuvatut testauksen eri osaamisalueet ja antavat konkreettiset ja selkeät ohjeet testauksen kehittämiseksi. Näin ollen testaustiimin osalta on erittäin tärkeäksi verrata nykyistä testausprosessia, menetelmiä ja työskentelyohjeistusta näihin malleihin. Vertailun perusteella testauksen kehittäminen saadaan käynnistettyä perustasolta ja vähitellen voidaan edetä vaativimmille tasoille esimerkiksi TMMi-mallin toisen tason yhteydessä havaittujen puutteiden osalta huomioiden TPI-mallissa vastaavalle tasolle merkityt parannusehdotukset. CTP-mallista testausta käsitellään pieninä prosesseina, jolloin testausprosessin kehittämisen yhteydessä voidaan varmistaa kriittisten prosessien muodostaminen tai jopa säilyvyys kriittisissä prosesseissa kehittämisestä huolimatta.

Testauksen kehittämisen tulisi olla hallittua, joten testauksen arviointia ja siihen liittyviä parannuksia varten on tehtävä selkeät kehityssuunnitelmat ennen muutosten toimeenpanoa. Muutostarpeiden arviointi tulisi vaiheistaa siten, että jokainen testauksen osa-alue tulisi huomioitua ja arvioitua samalla tehokkuudella ja yhtenäisellä tasapainolla. Samassa yhteydessä tulisi arvioida testauksen osaamistarpeita ja niiden täydentämismahdollisuuksia olemassa olevia testausresursseja käyttäen tai uutta henkilökuntaa palkaten. Kehittäminen pitäisi olla myös testauslähtöistä, jolloin tehtävien muutosten pitäisi tukea testaustiimin

työtä, eikä ainoastaan karsia työhön käytettäviä työtunteja. Jokaisen tulevan muutoksen yhteydessä on valmisteltava sekä testaustiimi, että projektiryhmät vastaanottamaan tulevaa muutosta. Lisäksi kehittämisen yhteydessä on huomioitava, että tehtävät muutokset on oltava linjassa yrityksen tavoitteiden ja toimintatapojen kanssa.

On erityisen tärkeää, että kehitystyötä tehdään keskitetysti. Testaustiimin työskentelyyn vaikuttavat projektien menetelmämuutokset vaikuttavat aina myös testaustiimin kykyyn hahmottaa toimintamallit ja sisäistää ne. Jos eri projektit kehittävät toimintaansa itsenäisesti, on selvää, että yhteistä toimintatapaa ei tiimien kesken saada. Erityisesti työskentelyn rajapinnoissa tulisi tapahtua hallinnoidusti siten, että muutoksista sovitaan yhdessä eri projektien ja ennen kaikkea testauspäällikön kanssa. Muussa tapauksessa vuorovaikutuksen ylläpito kärsii ja sitä kautta yhteistyö testaustiimin ja projektiryhmien kanssa heikkenee.

On selvää, että testaustiimin kanssa toimiminen vaatii opettelua ja muutoksen läpivienti tiimissä ja erityisesti organisaation tasolla vaatii paljon työtä onnistukseen. Testauksen kehittämisen ja testaustiimissä toimivien testaajien työskentelyn onnistumisen ja mielekkään työympäristön kannalta on oleellista, että testaustiimi huomioidaan yhtenä tiimeistä suorittamassa merkityksellistä työtä tuotteiden laadun parantamiseksi. Työ ei onnistu ilman projektiryhmien ja tiimin esimiesten vahvaa tukea ja ymmärrystä.

4 POHDINTA

Kehittämistehtävän tavoitteena oli löytää ratkaisu testaustiimin ohjaamiseen ja osaamisen johtamiseen. Työssä lähdettiin tarkastelemaan perustehtäväkäsitetä ja tiimin johtamista perustehtävän ja muutoksen kautta. Tuloksena löydettiin, että toimiva työyhteisö edellyttää kunkin toimijan noudattavan omaa perustehtäväänsä. Toimiakseen hyvin työyhteisö tarvitsee tuekseen vuorovaikutustaitoja, organisaation ja työyhteisön tuen ja ennen kaikkea yhteisesti laaditut pelisäännöt. Tarkastelua vietiin pidemmälle siten, että kuvattiin osaamisen johtamisprosessi. Samassa yhteydessä arvioitiin osaamisen, tiedon ja suorituksen johtamista sekä esimiehelle ominaisia piirteitä suoriutuakseen työstään. Teoria tuki aiemmin käsiteltyä osuutta.

Tarkastelua laajennettiin testauksen näkökulmaan siten, että tutustuttiin eritasoisten testausstandardien kuvauksiin testauksen merkityksestä, tavoitteista ja periaatteista. Lisäksi hahmotettiin erilaisten kehys- ja kypsyysmallien käyttöä testauksen prosessin toimivuuden arvioinnissa ja kehittämisessä. Päivittäisen johtamisen tueksi ja testauksen kehittämisen mittaamiseksi esiteltiin Kanban-menetelmä. Käsitellyt asiat täydensivät yleisen johtamisen yhteydessä tehtyjä havaintoja testauksen ohjaamiseksi. Testaustiimillä on jo nyt käytössään paljon hyviä menetelmiä, mutta niiden tehokas käyttö vaatii muutosvastarinnan loppuun käsittelyä projektiryhmien osalta. Lisäksi on tehostettava testauksen kuorman ja päivittäisen työskentelyn etenemisen näkyvyyttä projektiryhmille. Toki testaustiimin omassa toiminnassa ja työskentelytavoissa on vielä kehitettävää ja sen tueksi on käytettävissä hyviä testauksen kehysmalleja, kuten TMMi ja TPI.

Kehittämistehtävän keskeisimmät kysymykset koskivat testausosaamista. Oli tarpeen selvittää millaista osaamista tiimissä tarvitaan ja miten kehittymistarpeita arvioidaan. Työn yhteyteen määriteltiin testauksen osaamisalueet ja niihin liittyviä osaamisia. Lisäksi löydettiin kehysmallit, joiden tarkkoja kuvauksia hyödyntämällä voidaan kehittää testausprosesseja ja testaustiimin toimintaa. Tes-

taajien vahvuuksien ja heikkouksien huomiointi resurssijaossa saadaan käsiteltyä laatimalla testaustiimistä osaamiskartta sisältäen jokaisen testaajan osalta tehdyn arvion suhteessa testauksen osaamistarpeisiin. Lisäksi testauspäällikön aktiivinen läsnäolo ja osallistuminen testaustiimin päivittäiseen työhön tukee testaajien osaamisten arviointia ja sijoittamista projekteissa siten, että resurssien jako tukee projektien etenemistä mahdollisimman hyvin. Samalla keinoin voidaan arvioida testaajien kehittymistarpeita yksilön itsensä ja testaustiimin näkökulmasta.

Keskinäisen vuorovaikutuksen osalta ja projektien onnistuneen testauksen varmistamiseksi tarvitaan tiivistä päivittäistä työskentelyä testaustiiminä. Tuetaan aktiivista vuorovaikutusta, osaamisen jakamista ja pyritään luomaan testaustiimissä ilmapiiri, jossa kommunikointia ja tiedonkulkua tuetaan. Tuki voi olla eri ohjelmistoja käyttäen, kuten sähköposti tai chat-työkalut tai tietojärjestelmiä käyttäen. Lisäksi kasvotusten keskustelu on vuorovaikutuskeino, johon kannustetaan päivittäisen työskentelyn yhteydessä. Vuorovaikutusta tukemaan otetaan käyttöön Kanban-menetelmä, joka avaa testauksen päivittäisen etenemisen ja työtilanteen testaustiimin lisäksi myös projektiryhmille vuorovaikutuksen tueksi.

Kehittämistehtävässä saadut tulokset ovat lupaavat. Uudet opit erityisesti testauksen kehysmallien osalta antaa tukea testaustiimin ohjaamiseksi ja osaamisen johtamiseksi. Näiden tulosten pohjalta on hyvä suorittaa testaustiimin osaamistarpeen arviointi, testausprosessin ja menetelmien kypsyyden arviointi sekä tulosten pohjalta laatia testaustiimille kehittämissuunnitelma.

LÄHTEET

- Grönroos Mauri G, Mahdollisuuden aika kohti virtuaalista organisaatiota, 2006, Tammer-Paino Oy, Tampere
- Hass, Anne Mette Jonassen, Guide to Advanced Software Testing, 2008, Artech House, Norwood, MA, USA
- Istqb 2010. Tietotekniikan liitto. ISTQB:n testaussanasto suomennettu versio 2.0. Luettu 3.3.2010. http://ttlry-fi-bin.directo.fi/@Bin/2177ffe00529be0c2ed73d44d41555fd/1267608504/application/pdf/14155799/istqb_sanasto.pdf.
- Istqb Advanced 2011. Tietotekniikan liitto. Advanced Level Syllabus, Versio 2007, Luettu 26.4.2011
<http://istqb.org/download/attachments/2326555/Advanced+Level+Syllabus+%282007%29.pdf>
- Istqb Expert 2011. Tietotekniikan liitto. Expert Level Syllabus, Versio 2010, Luettu 26.4.2011
http://istqb.org/download/attachments/2326555/Expert-Level-Syllabus-ImprovingTheTestProcess-Release-V1_0_2.pdf
- Istqb Foundation 2011. Tietotekniikan liitto. Sertifioitu testaaja Certified Tester, Perustason sertifiikaattisisältö, Foundation Level Syllabus, Versio 2010, Luettu 26.4.2011 <http://ttlry-fi-bin.directo.fi/@Bin/52a2bcb5a6251a73815534bb5c53a0c7/1303802225/application/pdf/162071547/FL%20Syllabus%2020101123.pdf>
- Istqb Intermediate 2011. Tietotekniikan liitto. Intermediate Certificate in Software Testing Syllabus, Versio 2010, Luettu 26.4.2011
<http://www.bcs.org/upload/pdf/pcstintsyll.pdf>
- Kannisto Päivi & Salenius Bo-Magnus & Sigfirds Camilla, Johtamisen pakolliset kuviot, 2004, Talenum, Helsinki
- Kasslin-Pottier Hannele, Reilu ja rohkea esimies, 2009, Yrityskirjat Oy, Helsinki
- Kniberg Henrik, Kanban vs Scrum, version 1.1, 29.6.2009,
<http://www.crisp.se/henrik.kniberg/Kanban-vs-Scrum.pdf>, luettu 3.5.2011.

- Järvinen Pekka, Ammattina esimies, 2009, WSOYPro, Helsinki ja Esimiestyön haasteet, 18.11.2010, Seminaari Ilves-hotellissa, Tampere
- Mattila Pekka, Otollinen tilaisuus – Miten tarttua muutokseen, 2008, Talentum, Helsinki
- Kivinen Mitro, Testausprosessin kehittäminen käytännössä, 2006, http://ttlry-fi-bin.directo.fi/@Bin/05214b3fb46b62c20102191f2c0f2c35/1287811628/application/pdf/1524657/Testausprosessien_kehitt%C3%A4minen_k%C3%A4yt%C3%A4nn%C3%B6ss%C3%A4.PDF, luettu 23.10.2010
- Mayor Päivi, Osaamisen johtaminen, syksy 2010 – luentomateriaali, luettu 25.4.2011
- Stenberg Arto, Ohjelmiston testaus 2007 - testauksen kehittäminen, 2007, http://www.pori.tut.fi/~stenberg/index_files/kehittaminen.pdf, luettu 20.4.2011
- Sydänmaalakka Pentti, Älykäs organisaatio, 2007, Talentum, Gummerus Kirjapaino Oy, Helsinki
- TMMi Foundation, Test Maturity model Integration (TMMi), Version 3.1, 2010. <http://www.tmmifoundation.org/downloads/tmmi/TMMi%20Framework.pdf>, luettu 3.5.2011
- Toroi Tanja, Testausprosessin kehittäminen prosessimallien avulla, Kuopion yliopisto, 2006, <http://his.uku.fi/avointa/julkaisut/ProsessinKehitys.pdf>, luettu 23.10.2010
- Vuori 2-2011, Vuori Matti, Testaus organisaatiossa - eri osapuolten näkökulmia laadunvarmistukseen ja testaamiseen, 1.7.2010, http://mattivuori.net/julkaisuluettelo/liitteet/nakokulmia_testaamiseen.pdf, luettu 20.4.2011
- Vuori 6-2011, Vuori Matti, Testauksen piirteiden kehityskaaria, 12.4.2010, http://mattivuori.net/julkaisuluettelo/liitteet/testauksen_kehityskaaria.pdf, luettu 20.4.2011

Käsitteet ja termit (Taulukko 7) on koottu pääosin ISTQB:n testaussanastosta (Istqb 2010). Toimeksiantajan testaustoimintojen yhteydessä on käytetty osittain erilaisia termejä. Näiden termien yhteydessä käsitteenä on kuvattu toimeksiantajan käyttämä termi ja kuvauksen yhteydessä tarkennettu testaussanaston vastaava käsite.

Taulukko 7 Käsitteet ja termit

Käsite	Kuvaus
Apinatestaus	(Monkey Testing) Testausta, jossa syötteet valitaan satunnaisesti suuresta joukosta ja jossa satunnaisesti painetaan näppäimiä, välittämättä lainkaan siitä, miten tuotetta on tarkoitus käyttää. (Istqb 2010).
Asennusohje	(Installation Guide) Millä tahansa sopivalla medially toimittut ohjeet, jotka ohjaavat asentajan asennusprosessin läpi. Ohje voi olla käsikirja, vaihe-vaiheelta ohje, asennusvelho tai mikä tahansa muu samankaltainen prosessikuvaus. (Istqb 2010.)
CTP	Critical Testing Process (CTP) -menetelmän mukaan testausprosessi voidaan jakaa pieniksi prosesseiksi, joista tietyt testausprosessit voidaan asettaa kriittisiksi.
Dokumentoinnin testaus	(Documentation testing) Dokumentaation esim. käyttö- tai asennusohjeen, laadun testaus. (Istqb 2010).
Dynaaminen testaus	(Dynamic testing) Komponentin tai järjestelmän testaus suorittamalla ohjelmisto. (Istqb 2010).
Ei-toiminnallinen vaatimus	(Non-functional requirement) Vaatimus, joka ei kohdistu toiminnallisuuteen, vaan ominaisuuksiin kuten luotettavuus, tehokkuus, käytettävyys ja ylläpidettävyys ja siirrettävyys.

	(Istqb 2010).
Havainto	(Incident) Mikä tahansa tapahtuma, joka aiheuttaa tutkimista. (Istqb 2010).
Havaintojen hallinta	(Incident management) Prosessi, jossa havaintoja tunnustetaan, tutkitaan, toimitaan ja poistetaan. Tämä käsittää havaintojen kirjaamisen, luokittelun ja vaikutuksen tunnistamisen. (Istqb 2010.)
Havaintoraportti	(Incident report) Dokumentti, jossa kuvataan mikä tahansa esimerkiksi testauksen aikana sattunut tapahtuma, joka vaatii tutkimista. (Istqb 2010).
Hyväksymistestaus	(Acceptance Testing) Muodollinen testaus joka suoritetaan käyttäjän, asiakkaan tai muun valtuutetun tahon järjestelmän tai komponentin hyväksymisen määrittelemiseksi. Testaus tehdään vasten käyttäjätarpeita, vaatimuksia ja liiketoimintaprosesseja ja sen perusteella määritetään, täyttääkö järjestelmä hyväksymiskriteerit. (Istqb 2010.)
IDEAL	Testauksen kehittämismenetelmä, joka sisältää seuraavat vaiheet: Initiating, Diagnosing, Establishing, Acting ja Learning.
IMPROVE	Testauksen kehittämismenetelmä, joka sisältää seuraavat vaiheet: Initiate, Measure, Priorize and Plan, Define and Redefine, Operate, Validate ja Evolve
Inkrementaalinen	Eri iteraatioissa yhdestä toiminnosta toteutetaan vain osa, seuraavassa iteraatiossa ominaisuutta toteutetaan pidemmälle ja seuraavassa taas pidemmälle.

Integrointitestaustaus	(Integration testing) Testaus, jolla pyritään paljastamaan vikoja liittymissä ja integroitujen komponenttien tai järjestelmien välisessä vuorovaikutuksessa. (Istqb 2010).
Iteraatio	Iteraatio on kokonainen kehitysjakso, joka päättyy lopullisen tuotteen yksittäisen suoritettavan osan (sisäiseen tai ulkoiseen) julkaisuun (Istqb 2010). Iteraatio on kestoltaan 1-3 kuukautta.
Iteratiivinen	Projekti on jaoteltu useisiin iteraatioihin, joissa jokaisessa iteraatiossa toteutetaan osa tuotteen ominaisuuksista
Iteratiivinen kehitysmalli	(Iterative development method) Kehityksen elinkaarimalli, jossa projekti jaetaan (yleensä isoihin) iteraatioihin. Iteraatio on kokonainen kehitysjakso, joka päättyy lopullisen tuotteen yksittäisen suoritettavan osan (sisäiseen tai ulkoiseen) julkaisuun. Kehitettävä tuote kasvaa iteraatiosta toiseen lopulliseksi tuotteeksi. (Istqb 2010.)
Jäljitettävyys	(Traceability) Mahdollisuus tunnistaa yksiselitteisesti testaukseen liittyvät sovellukset ja dokumentaatio, kuten esimerkiksi vaatimukset. (Istqb 2010).
Järjestelmätestaustaus	(System testing) Testaus, jolla varmistetaan, että integroitu järjestelmä täyttää sille asetetut vaatimukset. (Istqb 2010).
Kanban	Projektinhallintamenetelmä, jota käytetään yleisesti ketterässä ohjelmistokehityksessä.
Kattavuusanalyysi	(Coverage analysis) Testien suorituksen aikana saavutetun kattavuuden mittaaminen määritellyä kattavuusyksikköä kohti. Tuloksia verrataan ennalta määriteltyihin kriteereihin, jolloin voidaan päätellä, tarvitaanko lisää testausta, ja jos tarvitaan, niin minkälaisia testitapauksia. (Istqb 2010.)

Ketterä testaus	(Agile Testing) Ketteriä ohjelmointimenetelmiä, kuten eXtreme programming (XP), käyttävien projektien testauskäytäntö, jossa kehittäjät ovat testauksen asiakkaita, ja jossa korostetaan testitapausten kautta tapahtuvaa suunnitteluparadigmaa. (Istqb 2010.)
Kokeileva testaus	Valmistelemattoman testauksen ja tutkivan testauksen yhdistelmä. Testaus, jossa alustavia versioita ominaisuudesta testataan staattisen testauksen perusteella tehtyjen testitapausten runkoja vasten. Samalla tutkivan testauksen tapaan etsitään keinoja täydentää testitapauksia ja valmistelemattoman testauksen osalta testataan ilman suunnitelmaa. Kaikkea testattua ei välttämättä kuvata testitapauksiin.
Kuormitustestaus	(Load testing) Testityyppi mittaamaan komponentin tai järjestelmän käyttäytymistä kasvavan kuormituksen alla. Esim. samanaikaisten käyttäjien ja/tai tapahtumien määrä, jonka komponentti tai järjestelmä pystyy käsittelemään. (Istqb 2010.)
Käytettävyys	(Usability) Kuinka ymmärrettävä, opittava, käytettävä ja houkutteleva ohjelmisto on käyttäjän kannalta määrätyissä olosuhteissa käytettäessä. (Istqb 2010).
Käytettävyystestaus	(Usability Testing) Testaus, jolla määritetään, missä määrin ohjelma on ymmärrettävä, helppo oppia ja käyttää sekä houkutteleva käyttäjälle, kun sitä käytetään tietyissä olosuhteissa. (Istqb 2010).
Käyttötapaus	(Use case) Konkreettisen tuloksen tuottava tapahtumasarja käyttäjän ja järjestelmän välisessä vuorovaikutuksessa. (Istqb 2010).

Laatu	(Quality) Kuinka hyvin komponentti, järjestelmä tai prosessi vastaa tiettyjä vaatimuksia ja/tai asiakkaan tarpeita tai odotuksia. (Istqb 2010).
Lean johtaminen	Arvoa tuottamattomien prosessien poistaminen seuraavilla osa-alueilla: kuljetukset, varastot, liike, odotusaika, ylituotanto, yliprosessointi, viallinen tuote-
Lopetusehdot	(Exit Criteria) Joukko sidosryhmien sopimia yleisiä ja erityisiä ehtoja, joiden täyttyessä sallitaan prosessin virallinen lopetus. Tarkoituksena on estää se, että tehtävä ajatellaan lopetetuksi, kun siinä on vielä käynnissä olevia osia. Testauksen edistymistä raportoidaan lopetusehtoja vastaan ja ehtojen perusteella päätetään, milloin testaus lopetetaan. (Istqb 2010.)
Loppukäyttäjättestaus	(User Test) Testaus, jossa todelliset käyttäjät ovat mukana arvioimassa komponentin tai järjestelmän käytettävyyttä (Huom: loppukäyttäjättestaus voi koskea myös muita tavoitteita kuin käytettävyyttä). (Istqb 2010).
Muistivuoto	(Memory leak) Ohjelmiston dynaamisen muistinvarauslogiikan vika, joka estää muistin vapautumista käytön jälkeen ja aiheuttaa ohjelman suorituksen epäonnistumisen muistin puutteen vuoksi. (Istqb 2010).
Nauhoitus/Toistotyökalu	(Capture/Playback tool) Testauksen suoritusta tukeva työkalu. Testisyötteen nauhoitetaan manuaalisen testauksen aikana ja niistä generoidaan automaattisesti testiskriptejä, jotka voidaan toistaa myöhemmin uudelleen. Näitä työkaluja käytetään tukemaan automatisoitua regressiotestausta. (Istqb 2010.)

Negatiivinen testaus	(Invalid testing, Negative testing) Testaus käyttäen syöteinä arvoja, jotka komponentin tai järjestelmän pitäisi hylätä. Testaus, jonka pyrkimyksenä on osoittaa, että komponentti tai järjestelmä ei toimi. Negatiivisen testauksen pohjana ovat enemmänkin testaajien asenteet kuin määrätyt testauksen lähestymistavat tai suunnittelutekniikat, esim. testaus epäkelvoilla arvoilla tai poikkeustilanteiden testaus. (Istqb 2010.)
Odotettu tulos	(Expected Result) Määrittelyyn tai muuhun lähteeseen perustuva komponentin tai järjestelmän ennustettu käyttäytyminen määritetyissä olosuhteissa. (Istqb 2010).
Ominaisuus tai toiminnallisuus	(Feature) Komponentin tai järjestelmän ominaisuus, joka on määritelty tai johon viitataan vaatimusmäärittelydokumentaatioissa. Esim. luotettavuus, käytettävyys tai suunnittelurajoite. (Istqb 2010.)
Paras käytäntö	(Best practice) Ylivoimainen menetelmä tai innovatiivinen käytäntö, joka myötävaikuttaa organisaation parantuneeseen suorituskykyyn tietyssä viitekehyksessä. Vastaavat organisaatiot yleensä tunnustavat sen ”parhaaksi” (Istqb 2010).
Paritestausta	(Pair testing) Kaksi henkilöä työskentelee yhdessä virheiden löytämiseksi. Nämä kaksi henkilöä voivat olla molemmat testaajia, tai testaaja ja ohjelmoija, tai testaaja ja loppukäyttäjä. Yleensä virheiden etsimiseen käytetään yhtä tietokoneita yhdessä. (Istqb 2010.)
Poikkeama	Vaatimusten täyttymättä jääminen. (Istqb 2010)
Poikkeusten käsittely	(Exception Handling) Järjestelmän tai komponentin käyttäytyminen vastauksena virheelliseen syötteeseen (ihmisen tai

	toisen komponentin) tai sisäiseen häiriöön. (Istqb 2010).
Projektiriski	Riski, joka liittyy (testaus)projektin hallintaan ja kontrollointiin esim. henkilöresurssien puute, liian tiukka aikataulu, muuttuvat vaatimukset. (Istqb 2010).
Rasitustestaus	(Stress testing) Suorituskykytestauksen tyyppi, jolla arvioidaan järjestelmän tai komponentin toimintaa, kun sitä kuormitetaan yli sen odotetun tai määritetyn kuormituksen tai vähentyneillä resursseilla (esim. käytettävissä oleva keskusmuisti tai palvelimet). (Istqb 2010).
Regressiotestaus	(Regression testing) Aiemmin testatun ohjelman testaus siihen tehtyjen muutosten jälkeen. Testauksella varmistetaan, että muutokset eivät ole tuottaneet tai paljastaneet uusia vikoja ohjelmiston alueilla, joita ei ole muutettu. Testaus suoritetaan, kun ohjelmisto tai sen ympäristö muuttuu. (Istqb 2010.)
Rinnakkaisuustestaus	(Concurrency testing) Testaus jolla varmistetaan miten komponentti tai järjestelmä hallitsee kaksi tai useampaa samaan aikaan tapahtuvaa toimintoa. (Istqb 2010).
Riski	Tekijä, joka voisi aiheuttaa tulevaisuudessa negatiivisia seurauksia. ne ilmaistaan usein vakavuuden ja todennäköisyyden avulla. (Istqb 2010).
Riskianalyysi	Prosessi, jolla arvioidaan tunnistettuja riskejä ja niiden mahdollisten seurausten vakavuutta ja todennäköisyyttä. (Istqb 2010).
Riskien kontrollointi, riskien valvonta, riski-	Prosessi, jolla tehdään päätöksiä ja toteutetaan toimenpiteitä, joilla vähennetään riskejä tai pidetään riskit tietyllä tasolla. (Istqb 2010).

en hallinta	
Riskipohjainen testaus	(Risk-based testing) Lähestymistapa, jolla pyritään tuoteriskien vähentämiseen ja informoimaan sidosryhmiä riskien tilasta, projektin alusta alkaen. Siihen liittyy tuoteriskien tunnistaminen ja niiden käyttö testausprosessin ohjauksessa. (Istqb 2010.)
RUP	Iteratiivinen ohjelmiston kehitysprosessi.
Savutesti	(Intake test, Smoke-test) Aloitustestin muoto, jolla päätellään, onko komponentti tai järjestelmä valmis tarkempaan testaukseen. Suoritetaan yleensä testivaiheen alussa. Järjestelmän päätoiminnallisuuden kattava kaikista suunnitelluista testitapauksista valittu osajoukko, jolla varmistetaan että kriittisimmät toiminnot toimivat, mutta pienempiin yksityiskohtiin ei kiinnitetä huomiota. (Istqb 2010.)
Scrum	Ketterä ohjelmistokehityksen menetelmä, jossa iteraatiot on jaettu pienempiin jaksoihin. Jakson aikana toteutetaan osa iteraation toiminnoista.
Sprint	Sprint on 2-4 viikon pituinen kehitysjakso iteraation sisällä. Iteraatio on jaettu tasaisesti sprintteihin. Yhden sprintin aikana toteutetaan osa iteraation yhteydessä toteutettavista toiminnallisuuksista.
Staattinen testaus	(Static testing) Komponentin tai järjestelmän testaus määrittely- tai toteutustasolla suorittamatta ohjelmistoa, esim. katselmoinnit. (Istqb 2010).
STEP	Systematic Test and Evaluation Process (STEP) - testauksen kehittämismalli

Strateginen johtaminen	Määritellään toimintalinjat taloudelle, henkilöstölle ja johtamiselle. Rakennetaan visiota ja asetetaan tavoitteita, sekä kehitetään mittareita.
Suorituksen johtaminen	Suorituksen johtamisella ohjataan päivittäistä toimintaa.
Suorituskyky	(Performance) Miten hyvin järjestelmä tai komponentti saavuttaa suunnitellut toimintonsa tietyillä reunaehdoilla, jotka liittyvät prosessointiaikaan ja läpäisykapasiteettiin. (Istqb 2010).
Test Maturity Model Integrated (TMMi)	Viisiportainen tasomalli testausprosessin kehittämiseen, vahvaa sukua CMMI:lle. Kuvaa tehokkaan testausprosessin avainelementit. (Istqb 2010).
Test process improvement (TPI)	Jatkuva testausprosessin kehittämisen kehysmalli, joka kuvaa tehokkaan testausprosessin avainelementit ja on erityisesti suunnattu järjestelmä- ja hyväksymistestaukseen. (Istqb 2010).
Testaaja	(Tester) Osaava ammattilainen, joka on mukana yksittäisen komponentin tai järjestelmän testauksessa. (Istqb 2010).
Testauksen hallinta	(Test Management) Prosessi, jolla käytännön testaustyötä hallitaan, seurataan, mitataan jne., tavoitteena tuotteen ja testausprosessin mittaaminen projektin johdon päätöksenteon ja laadunohjauksen tueksi. Kuuluu normaalisti testauspäällikön tehtäviin. (Istqb 2010.)
Testauksen materiaalit	(Testware) Testausprosessissa syntyvät tuotteet, joita tarvitaan testien suunnittelussa, määrittelyssä, ja suorittamisessa, kuten dokumentaatio, skriptit, syötearvot, odotetut lopputulokset, käynnistys- ja lopetustoimenpiteet, tiedostot, tietokannat, ympäristö ja muut tarvittavat testauksessa käytettävät resurssit.

	tettävät ohjelmistot ja välineet. (Istqb 2010).
Testipisteanalyysi (TPA)	Yhtälöön perustuva testauksen (työmäärän) arviointimenetelmä, jonka pohjautuu toimintopisteanalyysiin (TMap).
Testauksen suoritusvaihe	(Test execution phase) Ohjelmistokehityksen elinkaaren hetki, jolloin ohjelmistotuotteen osia käytetään ja arvioidaan, täyttääkö ohjelmistotuote sille asetetut vaatimukset vai ei. (Istqb 2010).
Testauksen yhteenvetoreportti	(Test summary raport) Asiakirja, johon on koottu testausaktiviteetit ja testauksen tulokset. Siinä myös arvioidaan kaikki testauskohteet ja verrataan tuloksia hyväksymiskriteereihin. (Istqb 2010.)
Testaus	(Testing) Ohjelmistotuotteiden ja niiden liitännäistuotteiden suunnitteluun, toteutukseen ja arviointiin liittyvistä sekä staattisista että dynaamisista elinkaaren toiminnoista muodostuva prosessi. Prosessin tarkoituksena on arvioida, vastaavatko tuotteet niille asetettuja vaatimuksia, osoittaa, että ne sopivat suunniteltuun käyttöönsä, ja löytää virheitä. (Istqb 2010.)
Testausautomaatio	(Test automation) Ohjelmistojen käyttö testauksessa tai testausta tukeviin toimiin, esimerkiksi testauksen hallinta, testauksen suunnittelu sekä testitapausten suorittaminen ja tulosten analysointi. (Istqb 2010.)
Testausprosessi	(Test process) Perustestausprosessiin kuuluu testien suunnittelu ja valvonta, testauksen määrittely, toteutus ja suoritus, lopetuskriteerien tarkastus ja raportointi sekä päätös-

	toimenpiteet. (Istqb 2010).
Testauspäällikö	(Test Manager) Henkilö, joka vastaa testauksesta ja testattavan kohteen arvioinnista. Tyypillisiä tehtäviä ovat johtaminen, valvonta, vastaaminen suunnitelmista ja testausmenetelmistä. (Istqb 2010.)
Testaussuunnitelma	(Test Plan) Dokumentti, jossa kuvataan aiottujen testaus toimien tarkoitus, lähestymistapa, resurssit ja aikataulu. Rajataan testaus sekä määritellään, mitä ominaisuuksia testataan, testaustehtävät ja kuka vastaa tehtävistä, testauksen itsenäisyyden aste, testausympäristö, testauksen suunnittelutekniikat, testauksen aloitus- ja lopetuskriteerit ja perustelut niiden valinnalle, sekä testaukseen liittyvät riskit, jotka vaativat varasuunnitelmia. (Istqb 2010.)
Testaus toiminnot	Ks. Testausprosessi
Testiloki	(Test Log) Kronologinen tallenneketju relevantteja tietoja testien suorituksesta. (Istqb 2010).
Testin suoritusautomaatio	(Test Execution Automation) Ohjelmistojen, esim. nauhoita/toista-työkalun, käyttö testien suorituksen hallinnoimiseksi, todellisten ja odotettujen tulosten vertailuun, testauksen esiehtojen täyttämiseksi, sekä muuhun testauksen hallintaan ja raportointiin. (Istqb 2010).
Testauksen suunnittelu	(Testin toteutus = Test Implementation) Prosessi, jossa luodaan ja priorisoidaan testiproseduurit, luodaan testiaineisto ja mahdollisesti valmistellaan testikehys sekä kirjoitetaan automatisoidut testiskriptit. (Istqb 2010).
Testiohje	(Test charter) Selvitys testauksen tarkoituksesta ja mahdol-

	lisistä testausideoista. Testausohjeita käytetään usein tutkivan testauksen apuna. (Istqb 2010).
Testisessio	(Test session) Keskeyttämätön ajanjakso, joka käytetään testien suorittamiseen. Tutkivassa testauksessa kukin sessio keskittyy johonkin aiheeseen, mutta testaajat voivat session aikana myös etsiä uusia mahdollisuuksia tai olennaisia asioita. Testaaja luo ja suorittaa testitapaukset lennossa ja kirjaa niiden edistymisen. (Istqb 2010.)
Testiskripti	(Test script) Käytetään tavallisesti viittaamaan testitoimenpiteiden kuvaukseen, varsinkin automatisoituun sellaiseen. (Istqb 2010).
Testitapaus	(Test Case) Syötearvojen, suorituksen esiehtojen, odotettujen tulosten ja suorituksen jälkiehtojen muodostama kokonaisuus, joka on muodostettu tiettyä tavoitetta tai testauksen kohdetta varten tietyn ohjelmapolun testaukseen tai vaatimustenmukaisuuden varmistamiseksi. (Istqb 2010.)
Testitapausten kuvaus	(Testisuunnitelma = Test case specification) Dokumentti, jossa määritellään testattavan kohteen testaamisessa tarvittavat testitapaukset. (Testispeksi, Testisuunnitelma = test design specification) Dokumentti, jossa määritellään testattavan kohteen testausehdot, yksityiskohtainen testauksen lähestymistapa sekä testaukseen liittyvät korkean tason testitapaukset. (Istqb 2010.)
Tiedon johtaminen	Prosessilla luodaan, hankitaan, varastoidaan, jaetaan ja sovelletaan tietoa.

Tietojohtaminen	Tähtää siihen, että organisaatiossa kyetään käsittelemään inhimillistä pääomaa ja sitä kautta tuotetaan lisäarvoa muutuen tietopääomaksi ja kilpailukyvyksi.
Tietoturva	(Security) Ohjelmistotuotteen ominaisuudet, jotka vaikuttavat sen kykyyn estää luvaton joko tahallinen tai tahaton järjestelmään tai tietoihin käsiksi pääsy. (Istqb 2010).
Tiimi	Toimeksiantajan palveluksessa oleva ryhmä kehittäjiä, testaajia, projektipäälliköitä ja tuotepäälliköitä, jotka työskentelevät yhdessä. He työskentelevät projekteissa, joiden testaustoiminnot suorittaa tämän tutkimuksen piiriin kuuluva testaustiimi.
Todentaminen	(Verification) Määrättyjen vaatimusten täyttymisen vahvistaminen kokeellisesti ja objektiivisen todistusaineiston avulla. (Istqb 2010).
Toiminnallinen vaatimus	(Functional Requirement) Vaatimus, joka määrittelee toiminnon, joka komponentin tai järjestelmän pitää suorittaa. (Istqb 2010).
Tuoteriski	Riski, joka liittyy suoraan testauksen kohteeseen. (Istqb 2010).
Tuotevaatimus	(Vaatimus = Requirement) Käyttäjän tai sopimuksen, standardien, määritysten tai muun muodollisesti laaditun asiakirjan järjestelmältä tai järjestelmän komponentilta edellyttämä kyky tai ominaisuus ongelman ratkaisemiseksi tai tavoitteen saavuttamiseksi. (Istqb 2010).
Tuotevaatimusten hyväksyntäkriteerit	(Hyväksymiskriteerit = Acceptance criteria) Testin tai testien lopettamiskriteeri, jonka komponentti tai järjestelmä tulee täyttää, jotta käyttäjä, asiakas tai muu valtuutettu taho voi hyväksyä kokonaisuuden. (Istqb 2010).

Turvallisuus	(Safety) Kuinka hyvin ohjelmistotuote saavuttaa määrittelyssä käytössä ihmisiin, liiketoimintaan, ohjelmistoon, omaisuuteen tai ympäristöön kohdistuvan hyväksyttävät riskitasot. (Istqb 2010).
Tutkiva testaus	(Exploratory Testing) Epämuodollinen testisuunnittelutekniikka, jossa testaaja aktiivisesti valvoo testien suunnittelua samalla kun testejä suoritetaan, ja käyttää saamaansa tietoa uusien ja parempien testien suunnittelemiseen. (Istqb 2010).
Täydellinen testaus	(Exhaustive Testing) Lähestymistapa, jossa testijoukko kattaa kaikki syötteiden ja esiehtojen yhdistelmät. (Istqb 2010).
Vaatuspohjainen testaus	(Requirement-based testing) Testauksen ja testitapausten suunnittelu vaatimuksista johdettujen testauksen tavoitteiden tai ehtojen perusteella. Esimerkiksi testit, jotka suorittavat tiettyjä toimintoja tai testaavat ei-toiminnallisia ominaisuuksia. (Istqb 2010.)
Vaiheittainen testaus = Inkrementaalinen testaus	(Incremental Testing) Testaus, jossa komponentit tai järjestelmät integroidaan ja testataan yksi kerrallaan, kunnes kaikki komponentit tai järjestelmät on integroitu ja testattu. (Istqb 2010).
Valmistelematon testaus	(Ad-hoc testing) Testaus, joka tehdään ilman tunnistettua testitapausten suunnittelumenetelmää. Testaus, joka tehdään epävirallisesti, ilman muodollista testien suunnittelua, ilman tunnistettua testitapausten suunnittelumenetelmää ja ilman odotettuja tuloksia. Testaus etenee mielivaltaisesti ilman odotettuja tuloksia ja ohjeistusta testauksen suorittamiseen. (Istqb 2010.)

Vianhallinta	(Defect Management) Prosessi, jolla vikoja tunnistetaan, tutkitaan, niiden perusteella ryhdytään toimenpiteisiin ja vikoista lopulta päästään eroon. Siihen liittyy vikojen tallentaminen, luokittelu ja vaikutusten tunnistaminen. (Istqb 2010.)
Vika	(Defect) Komponentissa tai järjestelmässä oleva virhe, joka voi aiheuttaa sen, että komponentti tai järjestelmä ei pysty suorittamaan siltä edellytettävää toimintoa; esim. virheellinen lauseke tai muuttujan määrittely. Jos virhe kohdataan suorituksen aikana, se voi aiheuttaa komponentin tai järjestelmän häiriö. (Istqb 2010.)
Visiojohtaminen	Visiojohtamisella luodaan tarkoitus toiminnalle ja strategisella johtamisella suunta toiminnalle.
WIP	WIP tarkoittaa tehtävien määrää, joka on enintään mahdollista ottaa suoritukseen yhdessä työvaiheessa Kanban-menetyksessä.
Vikaraportti	(Defect Raport) Dokumentti, joka raportoi jonkin komponentin tai järjestelmän vian, joka voi aiheuttaa sen, että komponentti tai järjestelmä ei pysty suorittamaan siltä vaadittua toimintoa. (Istqb 2010).
Vikatiheys	(Defect Density) Komponentissa tai järjestelmässä olevien vikojen määrä jaettuna komponentin tai järjestelmän koolla (ilmaistuna jollain standardoidulla tavalla, esim. koodirivien määrä, luokkien tai toimintopisteiden määrä). (Istqb 2010).
Virhe	(Error) Ihmisen toiminta, joka tuottaa väärän tuloksen. (Istqb 2010).
XP	(Extreme Programming) Ketterä menetelmä, jolle tyypillistä testivetoinen kehitys (TDD, ATDD), pariohjelmointi, kiinteä ja avoin tiimityöskentely, jatkuva integraatio ja pienet julkai-

	sut
Yksikkötestaus	(Unit/ Component testing) Yksittäisten ohjelmistokomponenttien testaus. (Istqb 2010).
Ylläpidettävyys	(Maintainability) Kuinka helposti ohjelmistotuotetta voidaan muokata virheen korjaamiseksi, täyttämään uusia vaatimuksia, helpottamaan ylläpitoa tulevaisuudessa tai vastamaan ympäristömuutoksiin. (Istqb 2010).
Ymmärrettävyys	(Understandability) Kuinka hyvin käyttäjä ymmärtää ohjelmiston soveltuvan käyttöön ja miten sitä käytetään tiettyyn tehtävään ja tietyissä käyttöolosuhteissa. Huom: Ymmärrettävyys ei koske pelkästään käyttäjiä, vaan esimerkiksi arkkitehtuurin ymmärrettävyys on tärkeää testauksen näkökulmasta. (Istqb 2010.)