

VAASAN YLIOPISTO
TEKNIIKAN JA INNOVAATIOJOHTAMISEN YKSIKKÖ
TIETOTEKNIikka

Ville Salonen

TESTAUSPROSESSIN ARVIOINTI TESTING PROCESS
IMPROVEMENT -ANALYYSIN AVULLA

Tietotekniikan
pro gradu -tutkielma

Tietotekniikan maisterikoulutusohjelma

VAASA 2018

SISÄLLYSLUETTELO

1	JOHDANTO	5
1.1	Tutkimuksen tavoite ja rajaukset	6
1.2	Tutkimusmenetelmä	7
1.3	Tutkimuksen rakenne	8
2	TESTAUKSEN ROOLI OHJELMISTOKEHITYKSESSÄ	9
2.1	Testauksen merkitys ohjelmistokehityksessä	9
2.2	Testausmenetelmät ja -prosessit	11
2.3	Lähestymistapoja ohjelmistotestaukseen	14
2.4	Mitä laatu on?	16
2.5	Laadun mittaaminen ohjelmistokehityksessä	18
3	TESTING PROCESS IMPROVEMENT -ANALYYSI	22
3.1	Testing Process Improvementin avainalueet	22
3.2	Arvioinnin kypsyystasot	25
3.3	Kypsyystason määrittäminen	27
4	TPI-MALLIN MUOKKAAMINEN YRITYKSEN TARPEISIIN	30
4.1	Yrityksen valitsevat avainalueet	30
4.2	Yrityksen määrittämät tarkistuspisteet	32
4.3	Tarkistuspisteiden tärkeyden määrittäminen	33
5	TUTKIMUKSEN SUUNNITTELU	35
5.1	Haastattelut	35
5.2	Haastatteluihin valmistautuminen	36
5.3	Haastattelujen sudenkuopat	37
5.4	Haastattelujen kulku	39

5.5	Tutkittavat järjestelmät	40
6	TUTKIMUSTULOKSET JA ANALYYSI	41
6.1	Yritystason tutkimustulokset	41
6.2	Järjestelmätason tutkimustulokset	46
6.2.1	Järjestelmä 1	46
6.2.2	Järjestelmä 2	48
6.2.3	Järjestelmä 3	51
6.2.4	Järjestelmä 4	52
7	DISKUSSIO	55
	LÄHDELUETTELO	58
	LIITTEET	60

VAASAN YLIOPISTO**Tekniikan ja innovaatiojohtamisen**

yksikkö

Tekijä:

Ville Salonen

Tutkimuksen nimi:

Testausprosessin arviointi Testing Process Improvement -analyysin avulla

Ohjaajan nimi:

Juho-Pekka Mäkipää

Tutkinto:

Kauppatieteiden maisteri

Ohjelma:

Tietotekniikan maisterikoulutusohjelma

Pääaine:

Tietotekniikka

Opintojen aloitusvuosi:

2012

Tutkimuksen valmistumisvuosi:

2018

Sivumäärä: 85

TIIVISTELMÄ:

Ohjelmistokehitys kehittyy jatkuvasti monimutkaisempaan muotoon. Suurille yrityksille monitoimittajamalli ja monikanavaisuus ovat arkipäivää, kun taas kuluttajat vaativat arkipäivän ohjelmistoilta helppokäyttöisyyttä ja luotettavuutta. Kehityksen myötä myös ohjelmistotestauksen rooli yritysten sisällä korostuu, kun sen pitää pysyä kehityksessä mukana. Tuotantoon pääsevät ohjelmistovirheet voivat aiheuttaa yrityksille suuria kuluja kalliiden korjaustoimenpiteiden sekä kärsivän asiakaskokemuksen myötä. Tämän tutkielman tavoitteena on tutkia suomalaisen teleliikennealan yrityksen ohjelmistotestauksen laatua ja kartoittaa yrityksen sisäisiä ohjelmistotestaukseen liittyviä heikkouksia.

Tutkielman teoreettinen viitekehys pohjautuu alan kirjallisuuteen ja pääasiassa Sogetin kehittämään Testing Process Improvement eli TPI-malliin, joka pyrkii arvioimaan testausprosessin laatua ja etsimään siitä puutteita sekä kehittämiskohteita. TPI-mallissa ohjelmistotestausprosessi jaetaan useampaan osa-alueeseen, jotka pilkotaan yksittäisiin tarkistuspisteisiin. Yksittäisiä tarkistuspisteitä arvioimalla saadaan kullekin osa-alueelle muodostettua TPI-mallin mukainen kypsyyssaste. Tutkielman lopputulos muodostui puolistrukturoiduista haastatteluista sekä niiden tulosten teemoittelusta TPI-mallin tarjoamien avainalueiden mukaan. Ennen haastattelujen teettämistä TPI-mallia muokattiin vastaamaan tutkittavan yrityksen tarpeita sekä yrityksen ohjelmistotestauksessa käyttämää terminologiaa.

Tutkimustulosten pohjalta löydettiin selkeät TPI-mallin mukaiset yrityksen laajuiset ohjelmistotestausprosessin osa-alueiden kehittämiskohteet, kuin myös yksittäisiä järjestelmäkohtaisia heikkouksia. Lisäksi tutkielman yhteydessä muodostettiin yritykselle viitekehys tutkimuksen mahdolliseen uudelleen teettämiseen testausprosessin kehittämisen arviointiin sekä laajentamiseen tutkielman ulkopuolisiin järjestelmiin.

AVAINSANAT: TPI, ohjelmistotestaus, testausprosessi, laatu

UNIVERSITY OF VAASA
Technology and Innovations

Author:	Ville Salonen	
Topic of the Master's thesis:	Evaluating testing process with Testing Process Improvement - analysis	
Instructor:	Juho-Pekka Mäkipää	
Degree:	Master of Science in Economics and Business Administration	
Major:	Computer science	
Year of entering the University:	2012	
Year of completing the Master's thesis	2018	Pages: 85

ABSTRACT

Software development keeps evolving into more and more complex form. Companies often operate in multiple channels and work with multiple vendors, whereas consumers are expecting reliability and demanding more usability. With this development, the role of software testing and quality assurance keeps expanding. Software errors or malfunctions in production can cause major monetary and reputation losses as fixes at that stage can be costly and time consuming. The purpose of this research is to study and evaluate to quality of testing processes of a major Finnish telecommunications company and find weaknesses in the testing process.

The theoretical framework of study is based on the literature of the field and mainly to Soget's Testing Process Improvement model, which aims to evaluate the quality of testing process and to seek either singular points or broader sections of the process that require improvement to become more efficient. In TPI model the software testing process is divided into sections, which again are divided into singular check points. By evaluating these check points, TPI model gives each section of the testing process a degree of maturity. The research material was gathered by having half constructed interviews with each of the four system managers in charge of software's under inspection in the study. Before the interviews took place, the TPI model was modified a bit to meet the needs and terminology of company conducting the study.

In this study the testing process was dived into 14 different sections. Based on the research results, two of these sections were found to be clearly the biggest weaknesses in the testing process throughout the company. Also for each software some individual improvement points were found. In addition to the research results, a framework to reinstate or to expand the study to concern more software was constructed.

AVAINSANAT: TPI, software testing, testing process, quality

1 JOHDANTO

Ohjelmistotestauksen päätavoite on yksinkertaisuudessaan etsiä järjestelmistä virheitä ja auttaa ohjelmistokehittäjiä. Parhaimmillaan ohjelmistotestaus on koko ohjelmistonkehityskaaren läpi jatkuvaa yhteistyötä testaukseen osallistuvien henkilöiden sekä muun projektitiimin kanssa.

Useat samanaikaiset ohjelmistoprojektit ja monitoimittajaympäristössä työskentely on nykypäivää, mikä vaatii koko projektilta hyvää suunnittelua ja jatkuvaa uudelleenjärjestäytymistä. Ohjelmistotestaus on yksi oleellinen osa jokaista ohjelmistoprojektia, ja sen onnistuneella suunnittelulla ja toteutuksella voi olla suuret taloudelliset sekä aikataululliset vaikutukset.

”Mitä laatu on?” ja ”Miten laatua voi mitata?” ovat kysymyksiä, jotka usein näkee ohjelmistotestauksen kirjallisuudessa tai kuulee alan seminaareissa. Usein yritykset mittaavat järjestelmistä löydyttyjen vikojen määrää, virhetilanteita tai keräävät asiakasrajapinnasta avoimia palautteita. Nämä eivät kuitenkaan kerro varsinaisesta ohjelmistotestauksesta tai sen suorittamisen laadusta. Tässä tutkimuksessa esitetään keino ohjelmistotestauksen laadun arviointiin.

Tämä tutkimus pohjautuu alun perin Sogetin muodostamaan Testing Process Improvement eli TPI-analyysiin, joka pyrkii määrittämään ohjelmistotestauksen prosessien laatua. TPI-analyysissä ohjelmistotestaus on jaettu useaan eri avainalueeseen, jotka on pilkottu useaan eri tarkistuspisteeseen. Tarkistuspisteisiin liittyy erilaisia laatuvaatimuksia, joita täyttämällä päästään lähemmäs mahdollisimman tehokasta testausprosessia. Ensimmäinen TPI-kirja julkaistiin vuonna 1998, ja vuosien saatossa menetelmästä on kehittynyt kansainvälinen testaaajien ja laadunvarmistuksen suosima työväline.

Tutkimus tehdään suurelle suomalaiselle tietoliikennealan yritykselle, joka haluaa säilyttää anonymiteettinsä, joten siihen viitataan tässä työssä ainoastaan Yrityksenä. Yrityksessä ohjelmistotestaus on noussut viime vuosien aikana yhä suurempaan rooliin.

Sen merkitys on ymmärretty tärkeäksi jatkuvasti kehittyvällä alalla, sillä onnistuneella ohjelmistotestauksella on suoria vaikutuksia asiakaskokemukseen sekä projektien aikatauluihin. Tässä tutkimuksessa perehdytään neljään Yrityksen järjestelmäkehitysprojektiin kartoittamalla niiden nykytila ja osoittamalla niiden kehityskohteita TPI-analyysin avulla. Tutkimusaihe annettiin tutkimuksen tekijälle työnantona kyseisestä Yrityksestä.

1.1 Tutkimuksen tavoite ja rajaukset

Tämän tutkimuksen tavoitteena on arvioida Yrityksen ohjelmistotestauksen laatua. Tutkimuksessa kartoitetaan neljän Yrityksessä isossa roolissa olevan tietoliikennejärjestelmän testausprosessin nykytilanne TPI-malliin pohjautuvan analyysin avulla. Arvioinnin tulosten perusteella pyritään löytämään selkeitä kehityskohteita järjestelmien testausprosessista ja havaitsemaan selkeitä yrityskohtaisia testausprosessien heikkouksia. Tutkimuksessa muodostetaan myös TPI-mallin mukainen viitekehys Yrityksen järjestelmäkohtaisten testausprosessien kypsyysasteen määrittämiseksi. Halutessaan Yritys voi hyödyntää viitekehystä tutkimuksen teettämisessä myös tutkimuksen ulkopuolella olevien järjestelmien osalta.

Tutkittavat neljä järjestelmää valittiin yhdessä Yrityksen ja tutkimuksen tekijän kanssa sillä periaatteella, että ne palvelevat Yrityksen eri liiketoimintaosastoja ja ovat järjestelmäarkkitehtisuudeltaan mahdollisimman erilaisia. Liiketoiminnallisten osastojen eroavaisuuksilla pyritään tuomaan esille pitkän historian ja siiloutumisen luomia eroja, joiden pitäisi korostaa ovatko mahdolliset testausprosessin heikkoudet järjestelmäkohtaisia vai koko Yrityksen laajuisia. Järjestelmäarkkitehtuurin eroja haettiin erityisesti siitä, onko järjestelmän palvelimet ostettu pilvipalveluina vai sijaitsevatko ne yrityksen omilla palvelimilla.

Tutkimuksen teoreettinen viitekehys on johdettu pääosin Sogetin Business driven testing process improvement -mallista sekä aihetta käsittelevästä ohjelmistotestauksen kirjallisuudesta. Soget kehitti alun perin 1990-luvulla ohjelmistotestauksen laatua

arvioivan tutkimuskehityksen eli TPI-mallin, joka uudelleenarvioitiin 2010-luvun alussa, jolloin syntyi taloudellisen näkökulman paremmin huomioiva BDTPI-malli. Tässä tutkimuksessa käytetään jatkossa arviointikehyksestä nimeä TPI, sillä se on alan kirjallisuudessa yleisemmin esiintyvä termi. Tutkimuksen uudelleen teettäminen ja arvioitavien järjestelmien testausprosessien kehityksen seuranta on rajattu tutkimuksen ulkopuolelle.

1.2 Tutkimusmenetelmä

Tässä tutkielmassa tehdään TPI-mallin teoreettiseen viitekehykseen pohjautuva tapaustutkimus tietoliikennejärjestelmien testausprosessien nykytilasta. Tutkielmassa suoritetaan puolistrukturoituja haastatteluja keskustelemalla järjestelmien asiantuntijoiden kanssa muokatun TPI-mallin asettamista testausprosessien avainalueista, minkä perusteella kunkin järjestelmän testausprosessi pisteytetään TPI-mallin mukaisesti. Ennen haastattelujen pitämistä TPI-mallia muokataan vastaamaan Yrityksen tarpeita, toimintatapoja ja aiheeseen liittyvää termistöä. TPI-mallin muokkaamisesta kerrotaan tarkemmin luvussa 4.

Haastattelujen tulokset analysoidaan teemoittelemalla sekä kvantitatiivisesti. TPI-mallissa teemoittelu tehdään jakamalla testausprosessin eri osa-alueet omaan ylätasoon ryhmään kuuluvaksi avainalueeksi, jolloin esimerkiksi kullekin järjestelmälle saadaan ylätasoon ryhmään *testauksen johtaminen* kuuluvalla avainalueella *testausprosessien johtamiselle* oma kypsyyssaste. Kvantitatiivista analyysia tehdään kullekin järjestelmälle sekä järjestelmien kesken. Puolistrukturoitujen haastattelujen tulosten perusteella kullekin järjestelmälle täytetään taulukossa 5 kuvattu matriisi, jonka avulla testausprosessien nykytilaa pystytään esittämään myös numeroiden ja kuvaajien avulla. Tutkimusmenetelmästä kerrotaan tarkemmin luvussa viisi.

1.3 Tutkimuksen rakenne

Tutkimuksen toisessa luvussa kuvataan ohjelmistotestauksen merkitystä osana ohjelmistokehitysprosessia, tutustutaan erilaisiin ohjelmistotestauksen lähestymistapoihin sekä kuvataan ohjelmistotestauksen merkitystä osana laadukkaan ohjelmiston rakentamista.

Tutkimuksen kolmannessa luvussa perehdytään tarkemmin TPI-malliin. Luvussa kerrotaan, mistä mallissa käytettävät testausprosessin avainalueet ja niiden kypsyystasot muodostuvat sekä miten kunkin avainalueen kypsyystaso määritetään.

Tutkimuksen neljännessä luvussa kuvataan, miten Yritys käyttää luvussa kolme kuvattuja avainalueita ja minkälaisia yrityskohtaisia muokkauksia TPI-malliin tehtiin, jotta tutkimus vastaisi mahdollisimman hyvin Yrityksen tarpeita.

Viidennessä luvussa kuvataan työssä käytettävää tutkimusmenetelmää eli TPI-mallista johdettua puolistrukturoitua syvähaastattelua. Luvussa myös avataan haastatteluihin liittyvää valmistautumista ja niiden mahdollisia haasteita.

Kuudennessa luvussa puretaan syvähaastatteluiden tuloksia sekä havainnollistetaan niitä graafisesti ja numeraalisesti. Tuloksiin pohjautuvat johtopäätökset käydään läpi luvussa seitsemän.

2 TESTAUKSEN ROOLI OHJELMISTOKEHITYKSESSÄ

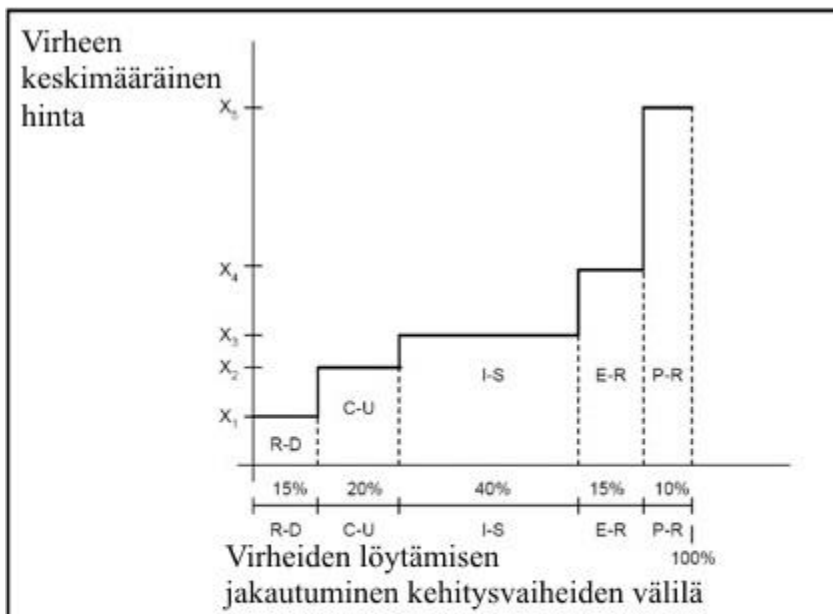
International Software Testing Qualifications Boardin opetusuunnitelma määrittää ohjelmistotestauksen tavoitteiksi ohjelmistovikojen löytämisen, ohjelmistovikojen estämisen, luottamuksen saavuttamisen ohjelmiston laatuun sekä tiedon keräämisen päätöksentekijöille (International Software Testing Qualifications Board 2011). Ohjelmistotestausta voidaan kuvata myös ohjelmistokehityksen osaprosessiksi, joka koostuu syötteistä, joista saadaan eri toimintojen kautta tuloksia (Jonassen Hass 2008: 36). Ohjelmistotestausta voidaan pitää myös osana laajempaa laadunvarmistus-käsitettä, joka sisältää varsinaisten testaustoimintojen lisäksi myös tekniset katselmoinnit, ohjelmistokehityksen kaikki vaiheet kattavan testausstrategian, tehokkaat testausmenetelmät ja -työkalut, ohjelmistokehityksen standardien noudattamisen, hallitun versionhallinnan sekä moniulotteisia kehityksen mittaus- ja raportointityökaluja (Pressman & Maxim 2012: 448).

Tämän kappaleen alaluvuissa käsitellään ohjelmistotestauksen taloudellisia ja aikataulullisia vaikutuksia ohjelmistokehitysprojekteihin sekä erilaisia testaustekniikoita ja lähestymistapoja ohjelmistotestaukseen. Lisäksi luvuissa kuvataan ohjelmistotestauksen vaikutus lopputuotteen tai -palvelun laatuun sekä määritetään erilaisia laadun ulottuvuuksia.

2.1 Testauksen merkitys ohjelmistokehityksessä

Joidenkin arvioiden mukaan ohjelmistotestauksen kulut voivat olla jopa puolet projektin kokonaiskustannuksista (Kasurinen, Taipale, Smolander 2010). On myös arvioitu, että yksin Yhdysvalloissa vuonna 2002 riittämättömän ohjelmistotestauksen aiheuttamat kustannukset nousivat 22,2–59,5 miljardiin dollariin, mistä yli puolen arvioidaan koostuvan käyttäjille aiheutuvasta haitasta. Loppukustannusten arvioidaan syntyvän testausresurssien tehottomuudesta, joka johtuu riittämättömistä testausmenetelmistä ja toimintatavoista. (Tassej, 2002.)

Ohjelmistovirheiden löytämisajankohdasta riippuvasta virheenkorjaushinnasta on esitetty monenlaisia arvioita. Joidenkin arvioiden mukaan tuotannossa löydettävä virhe on monta sataa kertaa kalliimpaa korjata, kuin jos virhe olisi havaittu jo vaatimusten analysointivaiheessa. Kuvassa 1 on esitetty Gregory Tasseyn Yhdysvaltojen National Institute of Standards and Technologylle teettämässä tutkimuksessa muodostettu arvio virheiden löytämisen jakautumisesta eri kehitysvaiheiden välillä sekä niiden korjauskustannuksista. Kuvasta nähdään, että neljännes virheistä löydetään vasta betatestauksessa tai tuotannossa, jolloin virheiden korjauskustannukset ovat moninkertaiset alun kehitysvaiheisiin verrattuna. (Tassey, 2002.)



R-D: Vaatimusten ja arkkitehtuurin suunnittelu
 C-U: Ohjelmointi ja yksikkötestaus
 I-S: Integrointi- ja systeemitestaus
 E-R: Betatestaus ja palaute
 P-R: Julkaisun jälkeinen aika

Kuva 1. Virheiden keskimääräinen löytämisajankohta ja korjaushinta (Tassey, 2002).

Tuotantoon päässeen ohjelmistovirheen aiheuttama todellinen haitta on aina tapauksesta riippuvainen, ja sen rahallisen haitan arvioiminen voi olla haastavaa. Tuotannonvirheistä voi aiheutua rahallisten kulojen lisäksi projektille suuria ajallisia menetyksiä. Tällaisia voivat olla esimerkiksi:

1. virheen löytämisestä aiheutuva ajallinen haitta
2. virheen raportoimiseen kuuluva ajallinen haitta
3. virhetilanteista vastaavalle henkilölle koitua ajallinen haitta virheraportin sisäistämisestä ja ratkaisun etsimisestä
4. tuotannonvirheen aiheuttama suorituskyvyn lasku tai järjestelmän mahdollinen alhaalla oloaika
5. ajallinen haitta, jos virhetilannetta ei saada ratkaistua ilman ulkopuolista tukea
6. virhetilanteen korjaustoimenpiteistä päättämisestä aiheutuva ajallinen haitta
7. virhetilanteen löytämisestä ja kooditasolla korjaamisesta aiheutuva ajallinen haitta
8. korjauksen testaamisesta ja asentamisesta aiheutuva ajallinen haitta. (Jonassen Haas 2008: 137.)

2.2 Testausmenetelmät ja -prosessit

Jos testausta johdetaan tapauskohtaisesti, se tarkoittaa yleensä, että siinä ei käytetä laatukriteereitä tai tekniikoita testitapausten luomiseen eikä määritellä tai priorisoida riskejä. Tapauskohtaiselle testaukselle on tyypillistä vajavainen tai olematon suunnittelu, testikattavuuden puuttuminen ja koordinoinnin puuttuminen testausta suorittavien osapuolien välillä. Tämä johtaa yleensä aikapaineisiin projektin loppupuolella sekä testauksen tehottomuuteen. (van Ewijk, Linker, van Oosterwijk, Visser, de Vries, Wilhelmus & Marselis 2013: 32–33.)

Tapauskohtaiseen testaukseen verrattuna hyvin suunnitellun testauksen etuihin voidaan lukea muun muassa virheiden löytäminen aikaisemmassa vaiheessa kehitystä, parempi riskien ymmärtäminen ja hallinta, testausprosessin parempi ymmärrettävyys ja hallittavuus, testitapausten uudelleenkäytettävyys sekä ajankäytön optimointi. Todellisuudessa ohjelmistotestaus on usein jotain tapauskohtaisen testauksen ja täydellisesti suunnittelun testauksen väliltä. (van Ewijk ym. 2013: 33.)

Testaussuunnitelma antaa ohjelmistotestaukselle suuntaviivat. Siinä kuvataan ohjelmistotestauksen aikana suoritettavat toimenpiteet, kuten esimerkiksi milloin, kenen toimesta ja millä prioriteetilla tehtävät tulee suorittaa. Testaussuunnitelman tulisi ohjata testaajia työtehtävissä ja samalla viestiä projektin etenemisestä. Kirjallisuudessa on useita esimerkkejä erilaisista ohjelmistotestausstrategioista. Usein niissä toistuvat seuraavat asiat:

1. Teknisten katselmointien tärkeyttä korostetaan, sillä ne eliminoivat virheitä aikaisessa vaiheessa.
2. Testaus aloitetaan yksikkötasolta, josta se etenee laajempiin kokonaisuuksiin aina integraatioihin asti.
3. Eri testaustekniikat sopivat eri kehitystapoihin ja -vaiheisiin.
4. Testauksen suorittavat sekä kehittäjät että erillinen testausryhmä.
5. Testaus ja virheiden korjaaminen ovat erillisiä toimintoja, joiden pitää olla osa testausstrategiaa. (Pressman ym. 2012: 466–467.)

Glenford J. Myers listaa teoksessaan *The Art of Software Testing* 12 hyvän testauksen suunnittelun ominaispiirrettä, jotka ovat:

1. Tavoitteet: jokaisen testausvaiheen tavoitteet pitää olla määritelty.
2. Hyväksyntäkriteerit: kriteerien pitää määrittää, milloin mikäkin testausvaihe on saatu valmiiksi.
3. Aikataulut: jokaiselle testausvaiheelle pitää olla aikataulu, joka sisältää testitapausten suunnittelun, kirjoittamisen ja suorittamisen.
4. Vastuut: jokaisen testausvaiheen toimenpiteille tulee olla vastuulliset henkilöt.
5. Testitapausten hallinta ja vakioimuotoisuus: varsinkin suurissa projekteissa systemaattiset menetelmät testitapausten hallintaan ovat tärkeitä.
6. Testaustyökalut: tarvittavat testaustyökalut tulee olla määritelty, ja niille tulee olla määritetty vastuuhenkilöt.
7. Infrastrukturi: vaadittavat palvelimet ja sovellusasennukset tulee olla etukäteen suunniteltu.

8. Laitteiston konfiguraatiot: mikäli laitteistot vaativat erityisiä konfiguraatioita, niiden tulee olla määritelty.
9. Integraatiot: integraatiot, niiden kokoamisjärjestys ja mahdolliset erityisjärjestelyt tulee huomioida.
10. Testauksen seuranta: testauksen aikana tulee voida seurata eri osa-alueiden testauksen edistystä, tunnistaa virheellisiä moduuleita sekä arvioida testauksen loppuun saattamiseen vaadittavat resurssit.
11. Virheiden korjaaminen: virheiden raportoinnin, niiden korjaamisen ja korjausten verifiointin pitää olla hallittuja toimenpiteitä.
12. Regressiotestaus: regressiotestauksessa tulee olla huomioitu kuka testaa, mitä testataan ja milloin testataan. (Myers 2004: 105–106.)

Myers antaa teoksessaan myös 10 tärkeää, mutta usein helposti unohdettavaa ohjetta ohjelmistotestaukseen ja siihen valmistautumiseen, jotka ovat:

1. Testitapauksessa tulee aina olla määritelty odotettu testitulos.
2. Ohjelmoijan tulisi välttää oman ohjelmistonsa testaamista.
3. Ohjelmointiorganisaatioiden ei tulisi testata omia ohjelmistojaan.
4. Jokaisen testin testitulokset pitäisi käydä tarkasti läpi.
5. Testitapauksessa tulisi olla määritelty sekä muodollisesti oikeanlaiset että invalidit syötteen.
6. Sen lisäksi että ohjelmistotestaaja varmistaa, että ohjelmisto toimii kuten pitää, on tärkeää nähdä, ettei ohjelmisto tee, mitä sen ei ole tarkoitus tehdä.
7. Testitapauksia ei tule hävittää, ellei testattava ohjelmisto ole oikeasti pian poistumassa käytöstä.
8. Aikataulua ei tule suunnitella olettamuksella, että ohjelmistosta ei löydy yhtään korjausta vaativaa virhettä.
9. Todennäköisyys virheiden esiintymiseen jossakin ohjelman osassa on verrannollinen kyseisen osion jo havaittujen virheiden lukumäärään.
10. Ohjelmistotestaus vaatii luovuutta ja on älyllisesti haastavaa. (Myers 2004: 16.)

2.3 Lähestymistapoja ohjelmistotestaukseen

Ohjelmistotestaus on tekninen toimenpide, joka kuitenkin sisältää ulottuvuuksia niin taloustieteistä kuin psykologiastakin. Ideaalitulanteessa jokainen ohjelmiston toiminnallisuus ja vaihtelu testattaisiin ennen sen käyttöönottoa, mutta jopa yksinkertaisimmat ohjelmistot voivat sisältää satoja tai tuhansia erilaisia syöte- ja tulostevasteita. Täydellinen testaaminen on usein epäkäytännöllistä ja veisi liikaa aikaa ja resursseja ollakseen taloudellisesti kannattavaa. Hyvän suunnittelun lisäksi ohjelmistotestaajilta vaaditaan oikeanlaista lähestymistä ja suhtautumista testaukseen. Myers kuvaa teoksessaan *The Art of Software Testing* ohjelmistotestausta prosessiksi, jonka tavoitteena on löytää ohjelmistosta virheitä. Vääränlainen suhtautuminen voi johtaa esimerkiksi liian hyvän testidatan käyttämiseen, mikä taas johtaa siihen, että mahdollinen virhe jää löytämättä. (Myers 2004: 10–11.)

Kuten Myers toteaa (Myers 2004: 10–11), ohjelmistotestausta, sen laajuutta ja lähestymistä suunniteltaessa tulee aina huomioida taloudellinen näkökulma, joka vaikuttaa siihen, millä tavalla ja minkälaisella kattavuudella testausta tulee tehdä. Alla tarkastellaan, mitä ovat black box- ja white box -testaus sekä missä tilanteissa yksikkö-, integraatio-, regressio-, alpha-, beta-, tietoturva-, suorituskyky- ja automaatiotestausta pääasiassa hyödynnetään.

Black box -testauksessa ohjelmiston testaaja tietää, mitä ohjelmiston tulisi tehdä, mutta ei tiedä, miten se sen tekee. Testaustavalle on tyypillistä, että testaaja syöttää ohjelmistoon mahdollisimman paljon erilaisia syötteitä ja tulkitsee niiden vasteet joko hyväksytyksi tai hylätyiksi. Black box -testauksella pyritään varmistamaan, toimiiko ohjelmisto määritelmien mukaisesti ja vastaako se käyttäjän odotuksia vai ei. (Myers 2004: 13.)

Toisin kuin black box -testauksessa, *white box* -testauksessa ohjelmistotestaaja tuntee ohjelmiston koodirakenteen. White box -menetelmässä testitapaukset johdetaan järjestelmäanalyysin pohjalta, jonka perusteella pyritään luomaan testitapauksille esimerkiksi mahdollisimman hyvä koodi- tai haarakattavuus. (Myers 2004: 13.)

Yksikkötestaus keskittyy ohjelmiston pienimmän mahdollisen komponentin tai moduulin testaamiseen. Yksikkötestaus keskittyy yksittäisen komponentin tai moduulin käsittelylogiikkaan ja tietorakenteisiin ja pyrkii löytämään niistä virheitä. (Pressman ym. 2012: 473.)

Kun useita ohjelmiston komponentteja ja moduuleita yhdistetään, tulee niille suorittaa *integraatiotestaus*. Vaikka osat olisivatkin läpäisseet yksikkötestauksen, ne voivat yhdistettäessä käyttäytyä odottamattomalla tavalla. Rajapinnoissa voidaan menettää tärkeitä tietoja, yksikkötasolla olemattomat virheet voivat muuttua merkittäviksi tai jokin osa voi tahattomasti vaikuttaa toiseen osaan haitallisesti. Kovan aikataulupaineen alla oleva ohjelmistoprojekti voi joutua yhdistämään monia, jopa kaikki, osat toisiinsa yhtä aikaa, mikä johtaa usein lukuisiin virhetilanteisiin. Inkrementaalissa integraatiossa testaus suoritetaan mahdollisimman pieniä osia yhdistelemällä, mikä usein auttaa löytämään virheet varhaisessa vaiheessa. (Pressman ym. 2012: 475–476.)

Joka kerta kun ohjelmistointegraatioon lisätään uusi moduuli, ohjelmisto muuttuu. Lisätty moduuli voi aiheuttaa muutoksia ohjelmistokoodin osassa, joka toimi ennen muutosta ja johon kehittäjät eivät uskoneet sen vaikuttavan. *Regressiotestauksen* tarkoitus on löytää tämäntyyppiset virheet suorittamalla ennalta määritellyt testitapaukset, jotka käsittävät ohjelmiston funktionaaliset toiminnot sekä erityisesti muutosta koskeneet toiminnot. (Pressman ym. 2012: 478–479.)

Ohjelmistoa suunniteltaessa on usein mahdotonta arvata, miten sitä lopulta käytetään. Käyttöohjeita ei välttämättä noudateta tai ne tulkitaan väärin. Myös ohjelmistoon syötetty data voi poiketa oletetusta. *Alphatestaus* suoritetaan yleensä ohjelmistotoimittajan tiloissa valvotuissa olosuhteissa. Loppuasiakas käyttää ohjelmistoa samalla, kun kehittäjät seuraavat käyttäjän ja ohjelmiston suoriutumista. *Betatestaus* suoritetaan loppukäyttäjän tiloissa ja toisin kuin alphatestauksessa, ohjelmistokehittäjät eivät yleensä ole paikalla seuraamassa ohjelmiston käyttöä. Loppuasiakas käyttää ohjelmistoa tuotannossa ja raportoi havaitut virheet kehitystiimille. (Pressman ym. 2012: 484–485.)

Kaikki arkaluontoisia tietoja sisältävät tietojärjestelmät, kuten esimerkiksi henkilötietoja tai tietoja, joista mahdollinen hyökkääjä voi hyötyä taloudellisesti tai joilla tämä voi aiheuttaa hyökkäyksen kohteena olevalle taloudellista haittaa, ovat potentiaalisia tietoturvahyökkäyksen kohteita. *Tietoturvatestauksen* tarkoituksena on varmistaa, että tietojärjestelmä on riittävän hyvin suojattu mahdollisten tietoturvahyökkäyksien varalta. Tietojärjestelmä kehityksen tulisi ottaa huomioon jo ohjelmiston suunnitteluvaiheessa turvallisuuspolitiikka, jonka tulisi ohjeistaa järjestelmän suunnittelua tärkeimmissä tietoturvavaatimuksissa. (Pressman ym. 2012: 486, 591.)

Reaaliaikaisissa ja sulautetuissa järjestelmissä suorituskyky on hyvin kriittisessä roolissa. Esimerkiksi verkkokauppajätti Amazonin on arvioitu menettävän vuositasolla 1,6 miljardia dollaria, jos verkkosivuston latausnopeudet hidastuisivat keskimäärin yhdellä sekunnilla. (Fast Company 2012.) *Suorituskykytestauksen* tarkoitus on mitata järjestelmän kyvykkyyttä suoriutua tehtävästään asetettujen aikamääreiden puitteissa. Suorituskykytestausta voidaan suorittaa kaikissa ohjelmistotestauksen vaiheissa ja sen yhteydessä suoritetaan usein myös stressitestausta. (Pressman ym. 2012: 487.)

Ohjelmistotestaus voidaan jakaa manuaaliseen ja automatisoituun testaukseen. *Automaatiotestausta* hyödynnetään usein tapauksissa, joihin liittyy paljon saman käyttötapauksen tai toiminnallisuuden toistamista, kuten esimerkiksi yksikkö- tai regressiotestauksessa. Tyypillisesti automatisoituihin tehtäviin liittyy testitapauksen suorittaminen sekä tulosten verifiointi. Automaatiotestausta suunniteltaessa pitää huomioida automatisoinnin kannattavuus sekä ohjelmiston testattavuus. Harvoin toistettavat testitapaukset kannattaa usein suorittaa manuaalisesti. Joskus myös ohjelmiston kompleksisuus tai keho laatu voivat tehdä automatisoinnista kannattamatonta. (Kasurinen ym. 2010.)

2.4 Mitä laatu on?

Laadun määrittäminen on usein vaikeaa monestakin syystä. Laatu on käsitteenä hyvin moniulotteinen, mutta samalla osa ihmisten arkipäivän sanastoa. Puhekielessä laadulla

voidaan tilanteesta ja puhujasta riippuen viitata suuriin kokonaisuuksiin tai pieniin yksityiskohtiin. Jotta ohjelmiston laatua voidaan kehittää, tulee laatu voida määrittää, ja sitä tulee voida mitata. (Kan 2004: 1.)

Yleinen käsitys laadusta on, että se on aineetonta. Siitä voidaan keskustella ja sitä voidaan arvostella, mutta sitä ei voida punnita tai mitata. Hyvästä ja huonosta laadusta puhutaan usein epämääräisesti, ilman aikomustakaan määrittellä, mitä niillä oikeastaan tarkoitetaan. Kaksi suosittua laadun määrittämistapaa ovat vaatimustenmukaisuus sekä käyttöön sopivuus. (Kan 2004: 1–2.)

Vaatimustenmukaisuudella tarkoitetaan yksiselitteisiä kehitysvaatimuksia, joita ei voida ymmärtää väärin. Laatumittaukset tehdään vaatimuksia vasten, ja täyttämättä jääneet vaatimukset luokitellaan virheiksi – eli huonoksi laaduksi. Käyttöön sopivuus on hyvin samankaltainen määrittelytapa, mutta se huomioi enemmän loppuasiakasta. Tällöin laatuvaatimukset johdetaan asiakastarpeesta, ja koska asiakkaat usein käyttävät tuotteita tai palveluita eri tavoin, voi käyttöön sopivuus tarkoittaa eri asiakkaille eri asioita. (Kan 2004: 2–3.)

Vaatimustenmukaisuutta tai käyttöön sopivuutta voidaan käyttää laadun määritelmänä niin yleisesti kuin myös ohjelmistokehityksessä. International Software Testing Qualifications Board määrittää laadukkaaksi ohjelmistoksi sellaisen, jossa on kohtalaisen vähän virheitä, se on toimitettu ajallaan ja määritetyn budjetin rajoissa, vastaa asetettuja vaatimuksia ja/tai odotuksia ja on helposti ylläpidettävä. International Software Testing Qualifications Boardin mukaan asiakkaille tärkeitä laatuominaisuuksia ovat:

1. hyvä suunnittelu
2. tarpeisiin vastaavat toiminnallisuudet
3. luotettavuus (hyväksyttävä määrä virheitä tai käyttökatkoja)
4. yhdenmukaisuus
5. kestävyys
6. hyvä palvelu myös ostopäätöksen jälkeen
7. rahalle saatu arvo. (International Software Testing Qualifications Board 2011.)

2.5 Laadun mittaaminen ohjelmistokehityksessä

Yksinkertaisin ja suppein tapa mitata ohjelmiston laatua on mitata siitä löytyvien virheiden määrää. Mitä vähemmän virheitä ohjelmistossa on, sitä laadukkaampi se on. Yleisimmin virheiden määrää mitataan laskemalla virheet määriteltyä koodirivimäärää tai toiminallisuutta kohden tai laskemalla ohjelmiston häiriötön käyttöaika tuotannossa määriteltyä aikamäärettä kohden. (Kan 2004: 4–5.)

Jim McCall määritteli tutkimuksessaan *Factors in Software Quality* 11 erilaista laatuominaisuutta, joita ohjelmistokehityksessä tulisi mitata. Nämä ovat ohjelmiston oikeellisuus, luotettavuus, tehokkuus, eheys, käytettävyys, ylläpidettävyys, testattavuus, muokattavuus, siirrettävyys, uudelleen käytettävyys ja yhteentoimivuus. (McCall, Richards & Walters. 1977: 3–5.)

McCallin laatuominaisuuksista *oikeellisuudella* tarkoitetaan sitä, miten hyvin ohjelmisto vastaa vaatimukseen ja täyttää käyttäjän tarpeet. *Luotettavuudella* kuvataan, missä määrin ohjelman voidaan odottaa suoriutuvan tehtävästään ilman virhetilanteita. *Tehokkuudella* mitataan laskentatehoa ja toiminnan suorittamiseen vaadittavan koodin määrää ja *eheydellä* järjestelmän ja sen sisältämän datan turvallisuutta. *Käytettävyydellä* tarkoitetaan sitä, kuinka helppokäyttöinen ohjelmisto on käyttäjän kannalta, kun taas *ylläpidettävyydellä* viitataan järjestelmävirheiden korjaamiseen vaadittavaan työmäärään. *Testattavuudella* kuvataan vaadittavaa testaustyömäärää tarvittavan ymmärryksen saavuttamiseksi ohjelmiston tasosta. *Muokattavuudella* tarkoitetaan sitä, kuinka helppoa ohjelmistoon on tehdä muutoksia, ja *siirrettävyydellä* laitteisto- ja ohjelmistopäivitysten tekemisen helppoutta. *Uudelleen käytettävyydellä* kuvataan mahdollisuutta uudelleen käyttää ohjelmiston koodia muissa hankkeissa ja *yhteentoimivuudella* ohjelmiston kykyä integroitua tarvittaessa muihin järjestelmiin. (McCall ym. 1977: 3–5.)

Taulukossa 1 on kuvattu McCallin määrittämät laatuominaisuudet ja niiden mittaustarpeet eri ohjelmistokehityksen vaiheissa. Taulukossa Δ -merkki kuvastaa tarvetta laatuominaisuuden mittaamiseen kyseisessä kehitysvaiheessa, kun taas X-merkki

kuvastaa kehitysvaihetta, jossa kyseisen laatuomaisuuden mahdolliset puutteet tulevat todennäköisimmin nousemaan esiin. (McCall ym. 1977: 3–11.)

Taulukko 1. McCallin laatuominaisuuksien suositellut mittauspisteet (McCall ym. 1977: 3–11).

Elinkaarivaiheet Laatuominaisuudet	Kehitys			Arvionti	Tuotanto		
	Vaatimusten analysointi	Suunnittelu	Ohjelmointi	Testaus	Operointi	Ylläpito	Muutos
Oikeellisuus	△	△	△	X	X	X	
Luotettavuus		△	△		X		
Tehokkuus	△	△	△		X		
Eheys	△	△		X	X	X	
Käytettävyys		△	△			X	
Ylläpidettävyys		△	△	X		X	X
Testattavuus		△	△			X	X
Muokattavuus		△	△				X
Siirrettävyys		△	△				X
Uudelleen käytettävyys		△	△				X
Yhteentoimivuus		△			X		X

Asiakkaiden tyytyväisyyttä ohjelmiston toimintaan ja käytettävyyteen mitataan usein erilaisilla tyytyväisyyskyselyillä. Kyselyillä voidaan esimerkiksi kartoittaa palveluun tyytyväisten asiakkaiden osuutta koko käyttäjäkunnasta. Syvällisemmistä asiakaskyselyistä voidaan kerätä tuloksia yritysten sisäisesti mittaamista ohjelmiston laatuominaisuuksista. Esimerkiksi IBM on mitannut asiakkaiden tyytyväisyyttä ohjelmistoihinsa jakamalla ohjelmiston laadun yhdeksään eri kategoriaan, jotka ovat toiminnallisuudet, käytettävyys, suorituskyky, luotettavuus, asennettavuus, ylläpidettävyys, dokumentointi/tiedotus, palvelu ja kokonaisuus. Hewlet-Packard taas on käyttänyt vastaavalla tavalla viittä erilaista mitattavaa laatuominaisuutta, jotka ovat

toiminnallisuudet, käytettävyys, luotettavuus, suorituskyky ja huollettavuus. (Kan 2004: 4–5.)

Mitattavia ohjelmiston laatuominaisuuksia määriteltäessä tulee huomioida ohjelmiston käyttötarkoitus ja loppuasiakkaat. Esimerkiksi laajoja verkkopalveluja tarjoavat yritykset haluavat todennäköisesti painottaa mittauksissa luotettavuutta ja suorituskykyä, kun taas itsenäisesti toimivaa ohjelmistoa mitattaessa asennettavuus ja dokumentaatio ansaitsevat todennäköisesti suuremman painotusarvon. Laatuominaisuuksia määriteltäessä tulee myös ottaa huomioon, että niillä voi olla vahvoja vaikutuksia toisiinsa. Esimerkiksi monimutkainen ohjelmisto on todennäköisesti vaikeammin ylläpidettävä kuin yksinkertaisempiin tarpeisiin tarkoitettu ohjelmisto. (Kan 2004: 4–5.)

Käytetään esimerkkinä IBM:n käyttämiä laatuominaisuuksia, ja tarkastellaan niiden suhteita toisiinsa. Kuvasta 2 nähdään, että laatumittareista esimerkiksi toiminnallisuudet on ristiriidassa suorituskyvyn, luotettavuuden, ylläpidettävyyden ja dokumentaation kanssa. Tämä tarkoittaa, että mitä monimutkaisempia toiminnallisuuksia ohjelmistoon tehdään, sitä vaikeampaa mainituissa laatuominaisuuksissa on saavuttaa hyviä tuloksia. Sitä vastoin asennettavuus laatuominaisuutena taas tukee käytettävyyttä, suorituskykyä sekä luotettavuutta.

Capability						
Usability						
Performance	●	●				
Reliability	●	○	●			
Installability		○	○	○		
Maintainability	●	○	●	○		
Documentation	●	○			○	

● = ristiriita
○ = toisiaan tukevat

Kuva 2. IBM:n laatuominaisuuksien suhteet toisiinsa (Kan 2004: 5).

Laatuominaisuuksille on tyypillistä, että niiden suora mittaaminen on vaikeaa tai joissain tapauksissa jopa mahdotonta. Esimerkiksi monet McCallin määrittämät laatuominaisuudet ovat mitattavissa vain epäsuorasti tai arvioihin perustuen. Tästä huolimatta McCallin mittareilla pystytään luomaan kuva ohjelmiston laadusta. (Pressman ym. 2012: 417.)

3 TESTING PROCESS IMPROVEMENT -ANALYYSI

TPI-analyysia käytetään testausprosessin nykytilan kartoitukseen osoittamalla sen vahvuudet sekä kehityskohteet mahdollisimman yksityiskohtaisella tasolla. Kuten alaluvuissa 4.1–4.3 havainnollistetaan, tämä tehdään pilkkomalla testausprosessi ensin useaan avainalueeseen, minkä jälkeen jokaiselle avainalueelle määritetään kypsyystaso hyödyntämällä avainaluekohtaisia tarkistuspisteitä.

Seuraavissa alaluvuissa kerrotaan myös avainalueiden kypsyystasoista ja niiden hyödyntämisestä nykytilakartoituksessa. Testausprosessin avainalue saavuttaa kypsyystason, kun se on täyttänyt kaikki vaadittavat tarkistuspisteiden vaatimukset. Lisäksi tarkistuspisteille määritellään tärkeysluokat, joiden avulla kehityskohteita voidaan priorisoida.

3.1 Testing Process Improvementin avainalueet

TPI-mallissa testausprosessi jaetaan 16 eri osa-alueeseen, jotta testausprosessin tehokkuuden mittaaminen ja kehittäminen olisi mahdollisimman helppoa. Kukin osa-alue voidaan lisäksi luokitella johonkin kolmesta ylätason ryhmästä, jotka ovat *sidosryhmäsuhteet, testauksen johtaminen ja testaajien asiantuntevuus*. Ylätason ryhmät mahdollistavat yksityiskohtaisten tarkistuspisteiden lisäksi ryhmien välisten suhteiden vertailun tutkimustulosten analysoinnissa. (van Ewijk ym. 2013: 43.) Kuhunkin TPI-mallin ylätason ryhmään kuuluvat avainalueet on esitetty kuvassa 3.

Sidosryhmäsuhteet	Omistajien sitoutuminen Sitoutuminen Testaust strategia Testausorganisaatio Kommunikaatio Raportointi
Testauksen johtaminen	Testausprosessin johtaminen Arviointi ja suunnittelu Mittaristot Vikojen hallinta Testausmateriaalin hallinta
Testaajien asiantuntevuus	Testausmetodit Testaajien ammattitaito Testitapausten suunnittelu Testaustyökalut Testiympäristöt

Kuva 3. TPI-mallin ylätasoon ryhmät ja niihin lukeutuvat avainalueet.

Sidosryhmäsuhteiden avainalueisiin kuuluvat *omistajien sitoutuminen, sitoutuminen, testaust strategia, testausorganisaatio, kommunikaatio ja raportointi*. *Omistajien sitoutumisella* kuvataan testauksen kanssa paljon tekemisissä olevien keskitason johtajien ja päälliköiden vaikuttamista testausprosessiin. Usein he asettavat testausprosessille budjetin ja aikataulun, mutta testausprosessi saa heiltä myös arvokasta tietoa, kuten testaustavoitteita ja vaatimuksia. (van Ewijk ym. 2013: 55.) *Sitoutuminen* avainalueena taas määrittää projektin sitoutumista testausprosessiin heti projektin alkuvaiheen toimenpiteistä alkaen läpi koko kehityksen (van Ewijk ym. 2013: 60). *Testausstrategialla* kuvataan testausprosessin kykyä keskittää käytettävissä olevat testausresurssit oikeaan paikkaan oikealla hetkellä mahdollisimman kustannustehokkaasti (van Ewijk ym. 2013: 66). *Testausorganisaatio*-avainalueella kartoitetaan testausorganisaation kykyä tarjota tietotaitoa sitä tarvitseville projekteille koko organisaatiossa (van Ewijk ym. 2013: 68). Avainalueessa *kommunikaatio* arvioidaan projektin kykyä tuottaa ja tarjota oikeamuotoista tietoa testausprosessin tueksi sekä testaajien osallistamista tarvittaviin päätöksentekotilanteisiin (van Ewijk ym. 2013: 74). Avainalue *Raportointi* sen sijaan

kuvaa testausprosessin kykyä tuottaa oleellista tietoa sovitussa muodossa testauskohteen omistajille päätöksenteon tueksi (van Ewijk ym. 2013: 78).

Ylätason ryhmään *testauksen johtaminen* kuuluvat avainalueet *testausprosessin johtaminen, arviointi ja suunnittelu, mittaristot, vikojen hallinta* sekä *testausmateriaalin hallinta*. Avainalue *testausprosessin johtaminen* kuvaa kykyä suunnitella ja toteuttaa testausprosessia hallitusti jatkuvasti muuttuvissa tilanteissa. (van Ewijk ym. 2013: 82.) Avainalue *arviointi ja suunnittelu* on vahvasti sidoksissa *testausprosessin johtamiseen*, sillä se arvioi testausprosessin suunnittelua ja edistymistä läpi ohjelmistokehityksen (van Ewijk ym. 2013: 86). Avainalue *mittaristot* kuvaa projektin kykyä hyödyntää ja luoda mittaristoja testausprosessin toimenpiteiden hyödyllisyyden todentamiseksi sekä toimintatapojen kehittämiseksi (van Ewijk ym. 2013: 90). *Vikojen hallinta* -avainalue arvioi testausprosessin aikana löydettyjen vikojen elinkaarenhallintaa sekä raportoitujen vikojen selkeyttä ja näkyvyyttä koko projektiryhmälle (van Ewijk ym. 2013: 96). *Testausmateriaalin hallinta* taas arvioi testausprosessin hyödyntämien lähdedokumenttien käyttöä sekä niiden suhdetta testitapauksiin (van Ewijk ym. 2013: 100).

Ylätason ryhmään *testaajien asiantuntevuus* kuuluvat avainalueet *testausmetodit, testaajien ammattitaito, testitapausten suunnittelu, testaustyökalut* ja *testiympäristöt*. Avainalue *testausmetodit* arvioi tapoja, joilla testausprosessin tehtäviä suoritetaan. (van Ewijk ym. 2013: 104.) *Testaajien ammattitaito* arvioi testustehtäviä suorittavien henkilöiden koulutustasoa, kokemusta ja sopivuutta työtehtävään (van Ewijk ym. 2013: 108). *Testitapausten suunnittelu* -avainalueessa arvioidaan testitapausten selkeyttä, järkevyyttä ja kattavuutta suhteessa lähdedokumenttiin sekä testitapausten tekemiseen käytettyjä tekniikoita ja niiden jatkuvaa kehittämistä (van Ewijk ym. 2013: 112). Avainalueessa *testaustyökalut* arvioidaan hyödynnettyjen testaustyökalujen sopivuutta sekä testausprosessiin että koko projektin tekemiseen (van Ewijk ym. 2013: 115). *Testiympäristöt*-avainalue arvioi niin kehitettävän ohjelmiston testiympäristöjen toimintatasoa kuin myös järjestelmäsidoitettujen ominaisuuksien ymmärtämistä (van Ewijk ym. 2013: 121).

TPI-mallin 16 avainaluetta listattuna:

1. Omistajien sitoutumien
2. Sitoutuminen
3. Testausstrategia
4. Testausorganisaatio
5. Kommunikaatio
6. Raportointi
7. Testausprosessien johtaminen
8. Arviointi ja suunnittelu
9. Mittaristot
10. Vikojen hallinta
11. Testausmateriaalin hallinta
12. Testausmetodit
13. Testaajien ammattitaito
14. Testitapausten suunnittelu
15. Testaustyökalut
16. Testiympäristöt (van Ewijk ym. 2013: 50.)

3.2 Arvioinnin kypsyystasot

Jokaiselle avainalueelle määritellään TPI-mallin mukainen kypsyystaso, joka on täyttyneiden tarkistuspisteiden perusteella joko alkuvaihe, kontrolloitu, tehokas tai optimoitu. Jotta avainalue voidaan määritellä kontrolloiduksi tai sitä ylemmälle tasolle, tulee sen täyttää kaikki kyseisen tason sekä alemman tason vaatimat tarkistuspisteet. Avainalueet arvioidaan aina ensimmäisen täyttämättä jääneen tarkistuspisteen mukaan. (van Ewijk ym. 2013: 46.)

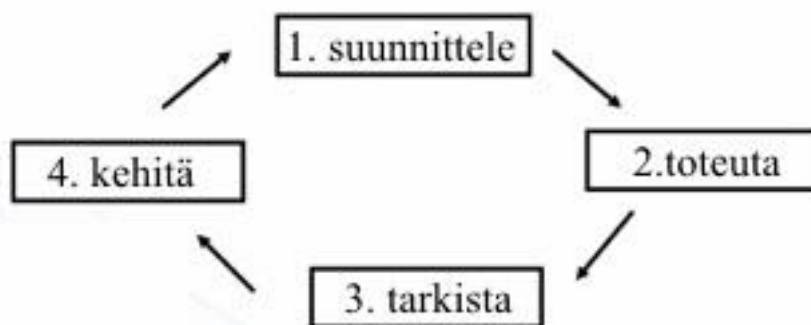
Alkuvaihe-tasolle ei ole määritelty erinäisiä vaatimuksia, vaan sillä kuvataan tilannetta, jossa testausvaihetta ei ole juurikaan suunniteltu ja testaustoiminnot tehdään projektin loppuvaiheessa ad hoc -toimenpiteinä, mikä tarkoittaa, että testauksessa luotetaan testaajien ammattitaitoon ja asiantuntemukseen ilman järjestelmällisyyttä. Alkuvaihe-tasolla testauksessa keskitytään lähinnä siihen, miten järjestelmä toimii, eikä siihen, toimiiko se vaatimusten mukaisesti. Tämän tason testaus aiheuttaa usein viivästyksiä

projekteihin, sillä virheitä löydetään vasta myöhäisessä vaiheessa ja niiden korjaamiseen kuluu aikaa. (van Ewijk ym. 2013: 46.)

Kontrolloidulla tasolla testaus on tunnistettu tärkeäksi prosessiksi ja siihen kiinnitetään huomiota jo projektin alkuvaiheessa. Tällä tasolla virheitä havaitaan aikaisemmassa kehitysvaiheessa, eikä niistä koidu niin suuria kuluja tai aikatauluviivästyksiä. Kontrolloidulla tasolla tehdään oikeita asioita, mutta ei aina välttämättä oikealla tavalla. (van Ewijk ym. 2013: 46.)

Jotta testausprosessi voi olla TPI-mallin mukaan tehokas, täytyy sen ensin olla kontrolloitu. Tehokkaalla tasolla kehitetään kontrolloidulla tasolla tehtäviä testaustoimenpiteitä kustannustehokkaampaan suuntaan. Kriittiset virheet löydetään mahdollisimman nopeasti siten, ettei niistä synny ylimääräisiä kustannuksia kehitysvaiheessa. (van Ewijk ym. 2013: 47.)

Kun testausprosessit on saatu tehokkaalle tasolle, pyritään ne optimoimaan. Tämä tarkoittaa jatkuvaa reagoimista testausprosesseihin vaikuttaviin muutoksiin siten, ettei prosessien taso laske muutostilanteissa. Jatkovaa osa-alueiden itsearviointia toteutetaan niin sanotun Demingin ympyrän mukaisesti, jossa prosesseja suunnitellaan, muokataan tarpeiden mukaan, tarkistetaan muutosten vaikutuksia ja kehitetään säännöllisesti. Demingin ympyrä on kuvattu kuvassa 4. (van Ewijk ym. 2013: 47.)



Kuva 4. Demingin ympyrä (van Ewijk ym. 2013: 48).

3.3 Kypsyystason määrittäminen

Kypsyystason määrittämiseksi jokaiselle avainalueelle tulee määrittää omat tarkistuspisteet. Kontrolloidulla ja tehokkaalla kypsyystasolla tarkistuspisteitä on 3–4 ja optimoidulla kypsyystasolla 2–3. Alla olevassa taulukossa 2 on kuvattu esimerkkinä TPI-mallin mukainen testauksen kypsyysmatriisi tarkistuspisteineen. Taulukosta nähdään muun muassa, että avainalueella *omistajien sitoutuminen* on kontrolloidulla kypsyystasolla neljä tarkistuspistettä ja tehokkaalla sekä optimoidulla kypsyystasolla kolme. Avainalueella *raportointi* vastaavat lukemat ovat kolme, kolme ja kaksi.

Taulukko 2. Esimerkki testauksen kypsyysmatriisista (van Ewijk ym. 2013: 48).

	Kontrolloitu				Tehokas				Optimoitu		
	1	2	3	4	1	2	3	4	1	2	3
Omistajien sitoutuminen	1	2	3	4	1	2	3	4	1	2	3
Sitoutuminen	1	2	3	4	1	2	3	4	1	2	3
Testausstrategia	1	2	3	4	1	2	3	4	1	2	3
Testausorganisaatio	1	2	3	4	1	2	3	4	1	2	3
Kommunikaatio	1	2	3	4	1	2	3	4	1	2	3
Raportointi	1	2	3	4	1	2	3	4	1	2	3
Testausprosessien johtaminen	1	2	3	4	1	2	3	4	1	2	3
Arviointi ja suunnittelu	1	2	3	4	1	2	3	4	1	2	3
Mittaristot	1	2	3	4	1	2	3	4	1	2	3
Vikojen hallinta	1	2	3	4	1	2	3	4	1	2	3
Testausmateriaalin hallinta	1	2	3	4	1	2	3	4	1	2	3
Testausmenetelmät	1	2	3	4	1	2	3	4	1	2	3
Testaajien ammattitaito	1	2	3	4	1	2	3	4	1	2	3
Testitapausten suunnittelu	1	2	3	4	1	2	3	4	1	2	3
Testaustyökalut	1	2	3	4	1	2	3	4	1	2	3
Testiympäristöt	1	2	3	4	1	2	3	4	1	2	3

Taulukossa 3 on kuvattu fiktiivisen yrityksen TPI-analyysin tulokset. Asetetut vaatimukset täyttäneet tarkistuspisteet on kuvattu taulukossa lisäämällä *-merkki tarkistuspisteen soluun. Taulukosta nähdään, että yhteensä seitsemän avainaluetta on saavuttanut kontrolloidun kypsyystason, sillä ne ovat täyttäneet kaikki tason vaatimat tarkistuspisteet. Esimerkkitaulukossa kontrolloidun kypsyystason saavuttaneet avainalueet ovat *testausstrategia*, *testausprosessin johtaminen*, *arviointi ja suunnittelu*, *vikojen hallinta*, *testausmateriaalin hallinta*, *testausmenetelmät* sekä *testaustyökalut*. Tehokkaan kypsyystason on saavuttanut ainoastaan avainalue *mittaristot*. Avainalue *kommunikaatio* on täyttänyt kaikki tehokkaan tason tarkistuspisteet, mutta se luokitellaan yhä alkuvaihe-tasolle, sillä se ei ole täyttänyt kaikkia kontrolloidun tason tarkistuspisteitä. Loput seitsemän taulukossa olevaa avainaluetta luokitellaan yhä alkuvaiheen

kypsyystasolle, koska ne eivät ole täyttäneet kaikkia kontrolloidulle kypsyystasolle vaadittavia tarkistuspisteitä.

Taulukko 3. Esimerkki fiktiivisen yrityksen TPI-analyysistä.

	Kontrolloitu				Tehokas				Optimoitu		
Omistajien sitoutuminen	1*	2*	3*	4	1	2	3		1	2	3
Sitoutuminen	1*	2	3	4	1	2	3		1	2	3
Testausstrategia	1*	2*	3*	4*	1	2	3		1	2	3
Testausorganisaatio	1	2*	3*	4	1	2	3	4	1	2	3
Kommunikaatio	1*	2	3	4	1*	2*	3*		1	2	3
Raportointi	1*	2	3*	4	1	2	3		1	2	3
Testausprosessien johtaminen	1*	2*	3*	4*	1	2	3		1	2	3
Arviointi ja suunnittelu	1*	2*	3*	4*	1	2	3	4	1	2	3
Mittaristot	1*	2*	3*	4*	1*	2*	3*	4*	1	2	3
Vikojen hallinta	1*	2*	3*	4*	1	2	3	4	1	2	3
Testausmateriaalin hallinta	1*	2*	3*	4*	1	2	3		1	2	3
Testausmenetelmät	1*	2*	3*	4	1*	2	3	4	1	2	3
Testaajien ammattitaito	1*	2	3*	4	1	2	3	4	1	2	3
Testitapausten suunnittelu	1*	2*	3	4	1	2	3	4	1	2	3
Testaustyökalut	1*	2*	3*	4	1	2	3	4	1	2	3
Testiympäristöt	1*	2*	3	4	1	2	3	4	1	2	3

Jokainen tarkistuspiste voidaan kohdentaa tärkeysluokkaan, jolla pyritään tarkentamaan tarkistuspisteen merkitystä yritykselle. TPI-mallin mukaisesti tarkistuspisteitä kuvataan kirjaimilla A–M siten, että A-kirjain edustaa tarkistuspisteistä ensisijaisesti täytettävää ja M-kirjain testausprosessin optimoinnin huippua. (van Ewijk ym. 2013: 53.)

TPI-mallin mukaan A-ryhmän tarkistuspisteiden kriteerit on suositeltavaa täyttää ensimmäisenä, sillä niiden vaikutus testausprosessin laatuun on merkittävin. Usein A-ryhmän tarkistuspisteiden vaatimukset ovat yritykselle myös helpoimpia ja yksinkertaisimpia täyttää. A-ryhmän tarkistuspisteiden täyttämisen jälkeen tulisi siirtyä aakkosissa seuraavana tulevaan täyttämättömään tarkistuspisteeseen. (van Ewijk ym. 2013: 53.) Taulukossa 4 esitetään taulukon 2 fiktiivinen tilanne käyttämällä matriisissa TPI-mallin asettamia tärkeysluokkia tarkistuspisteille.

Taulukko 4. Esimerkkimatriisi kirjaimilla.

	Kontrolloitu				Tehokas				Optimoitu		
Omistajien sitoutuminen	A*	B*	B*	C	F	H	H		K	M	M
Sitoutuminen	A*	B	C	E	H		J		L		L
Testausstrategia	A*	A*	B*	E*	F		H		K		L
Testausorganisaatio	A	D*	D*	E	I	I	J	J	K		L
Kommunikaatio	B	C	C	D	F*		J*		M		M
Raportointi	A*		C	C*	F		G		K		K
Testausprosessien johtaminen	A*	A*	B*	B*	G		J		K		M
Arviointi ja suunnittelu	B*	B*	C*	C*	G	H	I	I	K	L	L
Mittaristot	C*		C*	D*	G*	H*	H*	I*	K		K
Vikojen hallinta	A*	A*	B*	D*	F	F	H	J	K	L	L
Testausmateriaalin hallinta	B*	B*	D*	E*	I		J		L	L	L
Testausmenetelmät	C*		D*	E*	F*	H	J	J	M		M
Testaajien ammattitaito	D*	D	E*	E	G	G	I	I	K	K	M
Testitapausten suunnittelu	A*	A*	A*	E	F	I	I	J	K	K	M
Testausvälineet	E*		E*	E	F	G	G	I	L	M	M
Testiympäristöt	C*	D*	D	E	G	H	J	J	L	M	M

Taulukosta neljä nähdään, että yrityksellä on täyttämättä yksi A-tason tarkistuspiste, yksi B-tason tarkistuspiste sekä useampia C-tason tarkistuspisteitä. Kaikki tarkistuspisteiden vaatimukset on kuvattu liitteessä 1, josta nähdään, että *testausorganisaatio*-avainalueen A-ryhmän tarkistuspiste ”Muu organisaatio tietää, miten testausorganisaation palvelut ovat saatavilla” ei ole täyttynyt. Lisäksi nähdään, että avainalueen *sitoutuminen* B-ryhmään kuuluva tarkistuspiste ”Testaukseen liittyvät toimenpiteet aloitetaan hyvissä ajoin ennen varsinaisen testauksen suorittamista” sekä avainalueen *kommunikaatio* B-ryhmän tarkistuspiste ”Jokainen projektin jäsen on tietoinen tehdyistä päätöksistä ja kehityksen vaiheista” ovat myös täyttämättä. Tästä voidaan päätellä, että yrityksen välittömät korjaustoimenpiteet testausprosessin tehostamiseksi tulisi TPI-mallin mukaan kohdistaa kolmeen yllä mainittuun A- ja B-ryhmän tarkistuspisteeseen, sillä ne tulevat aakkosten määrittämässä tärkeysjärjestyksessä ensimmäisenä.

TPI-malli ei kuitenkaan anna tyhjentävää vastausta siihen, miten varsinaiset korjaustoimenpiteet tulisi tehdä, vaan tarjoaa vaihtoehtoisia korjausehdotuksia. *Testausorganisaation* osalta mahdollistajaksi mainitaan hyvä organisaation sisäinen tiedonohjausjärjestelmä ja vaihtoehtoiseksi korjausehdotukseksi esimerkiksi organisaatiossa näkyvässä roolissa oleva testauksen ryhmäpäällikkö (van Ewijk ym. 2013: 73.) Lopulta yrityksen pitää itse päättää sille parhaiten soveltuvasta korjaustoimenpiteestä.

4 TPI-MALLIN MUOKKAAMINEN YRITYKSEN TARPEISIIN

TPI-mallin kehittäjien teoksessa *TPI Next Business Driven Test Process Improvement* määritellyt avainalueet on kuvattu luvussa 4.1 ja tarkistuspisteet liitteessä 1. Tutkittavassa Yrityksessä koettiin tarvetta muokata avainalueita sekä osaa tarkistuspisteistä vastaamaan Yrityksen omaa laadunvarmistusprosessia. TPI-malliin tehdyt muutokset on kuvattu alla olevissa kappaleissa 4.1 ja 4.2.

Tutkittavan Yrityksen laadunvarmistusryhmä koostuu ryhmäpäälliköstä sekä noin 20 testausasiantuntijasta, joista osa on sisäisiä ja osa ulkoisia työntekijöitä. Kullakin asiantuntijalla on vetovastuullaan jatkuvasti 1-3 projektia tai hanketta, joihin sisältyy projektiin osallistuminen sopimusneuvotteluvaiheesta aina tuotantoon vientiin asti sekä mahdollisesti ylläpitovaiheen testauksen organisointia.

Joihinkin projekteihin saattaa osallistua useampi asiantuntija riippuen projektin koosta ja merkityksestä Yritykselle. Yrityksen sisäisillä asiantuntijoilla on kullakin myös erityisosaamista eri osa-alueilta, kuten testausautomaatiosta, suorituskykytestauksesta tai tietoturvatestauksesta, mitä hyödynnetään projektien tarpeiden mukaan.

4.1 Yrityksen valitsemat avainalueet

TPI-mallin avainalueet sekä tarkistuspisteet katselmoitiin Yrityksen laadunvarmistusryhmän kanssa, ja niihin päätettiin tehdä muutamia muokkauksia, jotta avainalueet vastaisivat paremmin Yrityksessä käytössä olevia testausprosesseja. Ryhmäkatselmointiin osallistuivat yrityksen testausasiantuntijat sekä laadunvarmistusryhmän ryhmäpäällikkö. Ryhmäkatselmoinnissa kaikki TPI-mallin avainalueet käytiin yksittäin läpi ja niiden merkittävyttä sekä sopivuutta verrattiin Yrityksen omiin testausprosesseihin. Avainalueisiin tehdyt muokkaukset on kuvattu alla.

Katselmoinnin tuloksena TPI-mallin avainalueet *omistajien sitoutuminen* sekä *sitoutuminen* niputettiin yhdeksi kohdaksi, *Projektin sitoutuminen testaukseen*, sillä se sopi paremmin Yrityksen projektimalliin sekä tutkimuksessa tutkittaviin kohteisiin. Avainalueista *testausmetodit*, *testausorganisaatio* ja *testaustyökalut* jätettiin tutkimuksen ulkopuolelle, sillä niiden tulokset olisivat olleet samat kunkin tutkittavan järjestelmän osalta.

Testausstrategia-avainalueen nimi muutettiin *testauksen suunnitteluksi*, sillä se on Yrityksessä käytettävä termi. Testausmateriaalien hallinta ei ole Yrityksessä keskeisessä roolissa, joten sen TPI-mallin mukaisia tarkistuspisteitä hyödynnettiin kohdassa *testitapausten suunnittelu*. Lisäksi tutkittaviin avainalueisiin lisättiin Yrityksessä keskeisessä roolissa olevat *testausautomaatio*, *suorituskykytestaus* sekä *tietoturvatestaus*.

Lopulta päädyttiin 14 avainalueeseen, jotka on listattu alla:

1. Projektin sitoutuminen testaukseen
2. Kommunikaatio
3. Raportointi
4. Testiympäristöt
5. Testauksen suunnittelu
6. Testausprosessin johtaminen
7. Työmääräarviointi ja suunnittelu
8. Mittaritot
9. Vikojen hallinta
10. Testaajien ammattitaito
11. Testitapausten suunnittelu
12. Tietoturvatestaus
13. Testausautomaatio
14. Suorituskykytestaus

Listatut avainalueet luokiteltiin lisäksi TPI-mallin mukaisiin ylätasen ryhmiin siten, että ylätasen ryhmään *sidosryhmäsuhteet* luokiteltiin avainalueet *projektin sitoutuminen testaukseen*, *kommunikaatio*, *raportointi* ja *testiympäristöt*. TPI-mallin vastaisesti Yrityksessä koettiin, että avainalue *testiympäristöt* sopii paremmin ylätasen ryhmään *sidosryhmäsuhteet* kuin *testaajien asiantuntevuuteen*, sillä Yrityksen testiympäristöissä on useita järjestelmäintegraatioita toisten projektien järjestelmiin. *Testauksen*

johtamiseen luokiteltiin avainalueet *testauksen suunnittelu, testausprosessin johtaminen, työmääräarviointi ja suunnittelu, mittaristot sekä vikojen hallinta*. Ylätason ryhmään *testaajien asiantuntevuus* sisällytettiin avainalueet *testaajien ammattitaito ja testitapausten suunnittelu* sekä Yrityksen malliin lisätyt *tietoturvatestaus, testausautomaatio ja suorituskykytestaus*. Yritykselle muokatun mallin ylätason ryhmät ja niihin lukeutuvat avainalueet on esitetty kuvassa 5.

Sidosryhmäsuhteet	Projektin sitoutuminen testaukseen Kommunikaatio Raportointi Testiympäristöt
Testauksen johtaminen	Testauksen suunnittelu Testausprosessin johtaminen Työmääräarviointi ja suunnittelu Mittaristot Vikojen hallinta
Testaajien asiantuntevuus	Testaajien ammattitaito Testitapausten suunnittelu Tietoturvatestaus Testausautomaatio Suorituskykytestaus

Kuva 5. Ylätason ryhmät ja niihin lukeutuvat avainalueet Yrityksen tarpeisiin muokatussa mallissa.

4.2 Yrityksen määrittämät tarkistuspisteet

Avainalueiden tapaan myös TPI-mallin tarkistuspisteet tarkastettiin Yrityksen testausasiantuntijoiden sekä ryhmäpäällikön kanssa. Suurin osa tarkistuspisteistä hyväksyttiin sellaisenaan, tai niihin tehtiin ainoastaan muodollisia muutoksia, joiden avulla tarkistuspisteiden termistöä muutettiin vastaamaan Yrityksen sanastoa tai käytäntöjä, ilman että asiasisältö muuttui. Esimerkiksi avainalueen *testaajien*

ammattitaito tehokkaan tason ensimmäisessä tarkistuspisteessä ei kysytty TMAP NEXT -sertifikaattia, sillä ISTQB-sertifikaatin suorittaminen on osa Yrityksen normaalia koulutusohjelmaa.

TPI-mallin ulkopuolelta avainalueiksi nostettujen *testausautomaation*, *suorituskykytestauksen* sekä *tietoturvatestauksen* tarkistuspisteet määrittivät kunkin osa-alueen Yrityksen johtavat testausasiantuntijat. Yrityksen määrittämät tarkistuspisteet on kuvattu yksityiskohtaisesti liitteessä 2.

4.3 Tarkistuspisteiden tärkeyden määrittäminen

Yrityksen tarkistuspisteiden määrittämisen jälkeen kullekin tarkistuspisteelle annettiin tärkeysluokka käyttämällä kirjaimia A–M. Yrityksen asiantuntijat antoivat ensin kullekin tarkistuspisteelle tärkeysluokan, jonka jälkeen niitä verrattiin TPI-mallin alkuperäisiin tärkeysluokkiin, jotka on kuvattu taulukossa 3.

Vertailussa selvisi, että asiantuntijoiden ja TPI-mallin määritykset olivat hyvin lähellä toisiaan. Lopullisen ehdotuksen tarkistuspisteiden tärkeysluokista hyväksyi vielä Yrityksen laadunvarmistusryhmän ryhmäpäällikkö. Tarkistuspisteille määritetyt tärkeysluokat on kuvattu alla olevassa taulukossa 5.

Taulukko 5. Yrityksen tärkeysluokat tarkistuspisteille.

	Kontrolloitu				Tehokas				Optimoitu		
Projektin sitoutuminen testaukseen	A	A	B	D	F	F	G		K	K	M
Kommunikaatio	B	C	C	D	F	F	J		L		M
Raportointi	A	C	C	D	F	G	I	J	K		L
Testiympäristöt	C	C	D	E	G	H	J	J	L	M	M
Testauksen suunnittelu	A	B	C	C	F	G	H	J	K		M
Testausprosessin johtaminen	A	A	B	B	F	G	H		K		L
Työmääräarviointi ja suunnittelu	B	C	C	E	G	H	J		M		M
Mittaristot	B	B	C	D	F	G	I	I	K	M	M
Vikojen hallinta	A	A	B	B	F	F	H		L	L	L
Testaajien ammattitaito	C	C	C	C	G	H	I		K	L	L
Testitapausten suunnittelu	A	A	B	E	F	G	J	J	K	M	M
Tietoturvatestaus	A	B	D	D	F	G	G	G	K	K	L
Testausautomaatio	A	B	B	C	F	F	G	J	K	K	L
Suorituskykytestaus	C	C	D	D	F	F	G	J	K		M

TPI-mallin mukaisesti kontrolloidulla tasolla käytetään tarkistuspisteille arvoja A–E, tehokkaalle tasolle arvoja F–J ja optimoidulle tasolle arvoja K–M. A-arvon saaneita tarkistuspisteitä pidetään Yritykselle tärkeimpinä testausprosessin tehokkuuden kannalta, kun taas M-arvon tarkistuspisteiden vaatimuksia pidetään testausprosessin optimoinnin ääripäänä. TPI-mallin mukaan tarkistuspisteiden vaatimuksia tulisi täyttää aakkosjärjestyksessä siirtymällä seuraavaan aakkoseen, kun kaikki edellisen aakkosen tarkistuspisteiden vaatimukset on täytetty. (van Ewijk ym. 2013: 52.)

Taulukosta nähdään, että *Projektin sitoutuminen testaukseen*, *Testausprosessin johtaminen*, *Vikojen hallinta* ja *Testitapausten suunnittelu* sisältävät kukin kaksi A-tärkeysluokan tarkistuspistettä, joten niiden voi olettaa olevan Yritykselle merkittävimpiä osa-alueita. Vastaavasti *Testaajien ammattitaito* ei sisällä kontrolloidulla tasolla yhtäkään A- tai B-tärkeysluokan tarkistuspistettä, joten siihen panostaminen vaikuttaa olevan Yritykselle vähemmän olennaista.

5 TUTKIMUKSEN SUUNNITTELU

Tutkimus toteutetaan tapaustutkimuksena, johon kerätään tutkimusaineistoa puolistrukturoiduilla haastatteluilla kunkin järjestelmän järjestelmävastaavan tai omistajan kanssa. Haastatteluissa esitetään avoimia kysymyksiä tutkittavan järjestelmän testausprosesseista ja -menetelmistä, joilla haetaan vastauksia liitteessä 2 esitettyihin TPI-mallista johdettuihin tarkistuspisteisiin.

Mikäli tarkistuspisteisiin ei saada vastauksia avoimilla kysymyksillä, esitetään TPI-mallin tarkistuspiste kysymysmuodossa suoraan, esimerkiksi: ”Saadaanko toimittajan tekemästä testauksesta raportti sovitussa muodossa?”. Jos vastaus on tarkistuspisteen osalta hyväksytty, pyydetään vastaajaa antamaan konkreettinen esimerkki toimista, joiden perusteella myönteinen vastaus annettiin. Mikäli haastateltava asiantuntija ei osaa antaa kaikkiin kysymyksiin varmaa vastausta, esitetään tarkentavat kysymykset järjestelmän muille asiantuntijoille.

Haastattelumetodina ei haluttu käyttää strukturoitua kyselylomaketta, jotta vastaukset olisivat mahdollisimman totuudenmukaisia. Strukturoitujen kyselylomakkeiden koettiin antavan järjestelmävastaaville mahdollisuuden kaunistella tuloksia oman järjestelmänsä osalta.

5.1 Haastattelut

Tutkimusaineiston hankinnassa käytetään kvalitatiiviseksi tutkimusmenetelmäksi luokiteltua puolistrukturoitua haastattelua. Puolistrukturoiduissa haastatteluissa haastattelijalla on lista läpikäytävistä teemoista ja kysymyksistä, jotka kuitenkin saattavat vaihdella haastattelusta toiseen. Tämä tarkoittaa, että haastattelija saattaa muuttaa kysymyksiä haastateltavasta riippuen tai vaihtaa kysymysten järjestystä. (Robson 2002: 270.)

Puolistrukturoiduissa haastatteluissa tutkimusongelmaa tarkastellaan mahdollisimman objektiivisesti. Tieteellisten tutkimusmenetelmien objektiivisuus voidaan määritellä seuraavasti:

1. Tutkimuskohteen ominaisuudet ovat tutkijan mielipiteestä riippumattomia.
2. Tieteellinen tieto syntyy tutkijan ja tutkimuskohteen välisen vuorovaikutuksen tuloksena.
3. Tiede ei voi perustua dogmien, uskon, ilmestyksen, auktoriteetin tai intuition varaan, vaan tiedon lähteenä ja kriteerinä tieteessä on viime kädessä itse tutkimuskohteesta saatava kokemus.
4. Tutkimuskohteesta on mahdollista saavuttaa totuudellista tietoa, ja tutkijayhteisössä on mahdollista päästä yksimielisyyteen tämän tiedon laadusta. (Aaltola & Valli 2007: 18.)

5.2 Haastatteluihin valmistautuminen

Puolistrukturoiduissa haastatteluissa valmistautumisella on iso merkitys, sillä vaikka läpikäytävät teemat ja kysymykset on osittain ennalta suunniteltu, oman uskottavuuden ja haastateltavan luottamuksen saavuttaminen on tärkeää halutun lopputuloksen saavuttamiseksi. Asiantuntemus aiheesta auttaa luomaan haastattelijasta uskottavan kuvan, mikä rohkaisee haastateltavaa antamaan syvällisempiä vastauksia. (Saunders, Lewis & Thornhill 2007: 318–340.)

Haastatteluihin valmistautuminen voidaan jakaa useaan eri osaan, joita ovat muun muassa asiantuntemus, haastattelun teemojen esittely haastateltavalle, haastattelun sijainti, haastattelijan ulkoasu, haastattelun avaus, kysymysten valmistaminen, kysymysten esittäminen, haastattelukäyttäytyminen, kuuntelutaidot, haastattelun nauhoittaminen sekä kulttuurierot. Tässä työssä haastattelija tarvitsi asiantuntemusta Yrityksen toimintakulttuurista, ohjelmistokehityksestä sekä testausmenetelmistä ja -prosesseista. Haastattelija voi myös parantaa uskottavuuttaan ottamalla yhteyttä haastateltavaan ennen varsinaista haastattelua ja kertoa lyhyesti haastattelussa läpikäytävistä teemoista. Tällä

tavalla haastateltava voi valmistautua haastatteluun ja mahdollisesti kerätä itse teemoja tukevaa tietoa yrityksen tietokannoista. (Saunders, Lewis & Thornhill 2007: 320.)

Haastattelupaikan sijainnin tulee olla haastateltavalle miellyttävä ja rauhallinen, jotta haastattelua ei keskeytetä. Haastattelutilan tulee olla hiljainen, jotta haastattelun nauhoittamisen laatu ei kärsi ulkopuolisesta melusta. Epämiellyttävä ja meluisa haastattelupaikka voi vaikuttaa negatiivisesti haastateltavan antamiin vastauksiin. Myös haastattelijan ulkoasu vaikuttaa haastattelutilanteeseen ja haastattelijan uskottavuuteen. Alipukeutuminen antaa haastattelijasta epäuskottavan mielikuvan, kun taas ylipukeutuminen voi saada haastateltavan kokemaan olonsa epämiellyttäväksi. Hyvänä pukeutumisohjeena haastattelijalle pidetään pukeutumista samalla tavalla kuin haastateltava. (Saunders, Lewis & Thornhill 2007: 321.)

Itse haastattelutilanteessa esille nousee haastattelukäyttäytyminen, haastattelijan kuuntelutaidot, haastattelun nauhoittaminen sekä kulttuurierot. Haastattelijan asento, eleet, puhetyyli ja äänensävy voivat kaikki vaikuttaa haastateltavan puolueellisuuteen. Haastattelijan käyttäytymisen tulee olla mahdollisimman neutraalia. Haastattelijan tulee käyttäytymisellään ja puhellaan innostaa haastateltavaa mutta samalla olla ilmaisematta omia mielipiteitään saaduista vastauksista. Esimerkiksi haastatteluasennon tulisi olla lievästi haastateltavaan päin nojaava, ja käsien tulisi olla avoimesti esillä. Näin haastateltava tuntee mielipiteensä arvostetuiksi. (Saunders, Lewis & Thornhill 2007: 327.) Tutkimuksen onnistumisen kannalta kuuntelutaidot ovat erittäin tärkeitä. Torringtonin, Hallin ja Taylorin (2007: 47) mukaan kuuntelu on signaalien etsimistä ja halua käyttää aikaa haastateltavan kuunteluun ja ymmärtämiseen omista mielipiteistä huolimatta.

5.3 Haastattelujen sudenkuopat

Etenkin kokematon haastattelija voi epäonnistua haastattelutilanteissa kokemuksen puutteen tai huonon valmistautumisen takia. Syvähaastatteluissa tulee ottaa huomioon monia asioita, ettei tutkimuksen luotettavuus vaarannu. Puolistrukturoiduissa

haastatteluissa haastattelun standardisoinnin puute, haastattelijan ja haastateltavan puolueellisuus, haastateltava henkilö sekä saatujen vastauksien yleistettävyyden puute ovat tekijöitä, jotka voivat nostaa esille kysymyksen tutkimuksen luotettavuudesta.

Tutkimustulosten luotettavuuden kannalta on tärkeää, että haastatteluihin saadaan juuri oikeat henkilöt. Usein haastatteluihin kuluva aika vähentää halukkuutta osallistua haastatteluihin, ja tästä syystä haastatteluun ei välttämättä saada sitä henkilöä, jota halutaan haastatella. (Saunders, Lewis & Thornhill 2007: 317–319.)

Haastattelijalla saattaa helposti vaikuttaa haastateltavan mielipiteisiin kysymysten asettelulla tai tuomalla esiin omia mielipiteitään. Tutkimustulosten luotettavuuden takaamiseksi haastattelussa esitettävät kysymykset eivät saa olla millään tavalla johdattelevia, eikä haastattelijalla saa tuoda omia mielipiteitään esille. Haastattelijalla on myös tärkeää kiinnittää huomiota äänensävyyn sekä sanattomaan viestintään, sillä ne ovat tekijöitä, jotka saattavat johdatella haastateltavan vastauksia. (Saunders, Lewis & Thornhill 2007: 317–319.)

Mikäli haastattelijalla ei onnistu pysymään puolueettomana, ei haastattelussa saatuja tuloksia voida käyttää johtopäätösten tekemiseen. Haastateltava voi myös antaa puolueellisia vastauksia tai vastauksia, jotka johtuvat haastattelijasta muodostuneesta mielipiteestä, arkaluontoisesta aiheesta tai oman aseman turvaamisesta. Näissä tapauksissa haastateltava antaa usein lyhyitä, yleispäteviä ja tarkentamattomia vastauksia, jolloin saadut vastaukset eivät ole kelpollisia johtopäätösten tekemiseen. (Saunders, Lewis & Thornhill 2007: 317–319.)

Hyvä keino haastateltavan puolueettomuuden takaamiseksi on pitää haastateltavat anonymineen tutkimustuloksia julkaistaessa (Robson 2002: 274). Hyvin todennäköinen sudenkuoppa on vastausten yleistettävyyden puute. Tästä syystä tutkimusmenetelmiksi valittu puolistrukturoitu haastattelu on hyvä menetelmä, sillä haastattelijalla pystyy tarkentamaan kysymyksiä ja pyytämään tarkennusta saatuihin vastauksiin. Kvalitatiivisen tutkimusmenetelmän edut kvantitatiivisiin tutkimusmenetelmiin nähden on kiteytetty seuraavassa lainauksessa: ”Kvalitatiivisten tutkimusmenetelmien ylivoimaisuus tiedon

keruussa on sen joustavuus ja reagoiva kanssakäyminen, joka mahdollistaa haastattelijan ja haastateltavan välillä käsitteiden tarkentamisen, kysymysten tarkentamisen ja aiheiden laaja-alaisen tarkastelun useista näkökulmista.” (Sykes 1991: 8, lainattu Saunders, Lewis & Thornhill 2007: 319.)

5.4 Haastattelujen kulku

Haastattelutilanteesta pyrittiin luomaan mahdollisimman rento ja haastateltavalle miellyttävä. Haastattelut järjestettiin haastateltaville tutuissa olosuhteissa Yrityksen tiloissa, ja haastatteluihin varattiin kunkin haastateltavan osalta aikaa 1,5 tuntia. Ennen haastattelun alkua haastattelija pyrki keskustelemaan yleisistä asioista haastateltavan rauhoittamiseksi ja korostamaan, ettei kyseessä ollut haastateltavan tai hänen vastuujärjestelmänsä työn arvioiminen, vaan haastattelun tarkoituksena olisi kerätä tietoa koko Yrityksen toiminnan ja testausprosessien kehittämiseksi.

Puolistrukturoidut haastattelut mukailivat TPI-mallin pohjalta Yrityksen tarpeita vastaamaan muokattua haastattelurunkoa, joka on kuvattu liitteessä 2. Haastatteluissa ei esitetty suoria kysymyksiä vaan keskustelua pyrittiin ohjaamaan teemoittain, esimerkiksi esittämällä avoimia kysymyksiä, kuten: ”Voitko kertoa teidän käyttämistänne vikojen hallintaprosesseista?”. Mikäli ensimmäinen avoin kysymys ei johdattanut keskustelua oikeaan suuntaan tai tuonut vastausta kaikkiin haluttuihin tarkistuspisteisiin, esitettiin esimerkiksi jatkokysymys: ”Hyödynnetäänkö vikojenhallintaohjelmaa raportoinnissa?”. Mikäli haluttuja tietoja ei saatu avoimilla kysymyksillä, esitettiin tarkistuspiste suoraan kysymysmuodossa, jolloin haastattelijaa pyydettiin antamaan esimerkki annetun vastauksen tueksi. Mikäli haastateltava ei tiennyt vastausta johonkin kysymykseen, esitettiin kysymys jollekin muulle kyseisen järjestelmän kanssa tiiviisti työskentelevälle henkilölle.

5.5 Tutkittavat järjestelmät

Tutkimuksen teettävä Yritys haluaa säilyttää anonymiteettinsä, joten tässä tutkimuksessa tutkittaviin järjestelmiin viitataan numeroilla 1-4. Kaikkien tutkittavien järjestelmien pääasiallinen tarkoitus on palvella kuluttaja-asiakkaita ja tuottaa heille erilaisia tietoliikennealan palveluita.

Järjestelmä 1 pyörii Yrityksen omilla palvelimilla. Järjestelmä 1 on Yrityksessä suuressa roolissa, koska se toimii asiakastieto-, provisiointi- ja laskutusjärjestelmänä. Lisäksi siellä toteutetaan erilaisia tuoteratkaisuja. Se palvelee Yrityksen kuluttaja-asiakkaita kiinteän ja liikkuvan laajakaistan sekä viihdepuolen palveluiden osalta. Järjestelmän ohjelmistokehitystyö tehdään vesiputousmallilla.

Järjestelmä 2 on ostettu Yritykselle pilvipalveluna, ja siihen liittyvän kehitystyön pääasiallinen tarkoitus on ylläpitää ja kehittää asiakkaille tarjottavia mobiiliratkaisuja. Järjestelmä 2 on muihin tutkittaviin järjestelmiin nähden poikkeuksellinen siinä mielessä, että sen testausprosessi on lähes kokonaan järjestelmän toimittajan vastuulla. Yrityksen vastuulla on pääasiassa toiminnan valvominen sekä ohjaaminen.

Järjestelmä 3 pyörii Yrityksen omilla palvelimilla, ja siihen liittyvän kehitystyön pääasiallinen tarkoitus on kehittää ja ylläpitää Yrityksen verkkokauppaa ja itsepalvelua. Järjestelmä 3 eroaa muista tutkituista järjestelmistä siten, että sen osalta otettiin Yrityksessä ensimmäisenä käyttöön ketterät kehitysmenetelmät.

Järjestelmä 4 on ostettu Yritykselle valmiina palvelualustana, jota on räätälöity hyvin vahvasti vastamaan Yrityksen tarpeisiin. Siihen liittyvän kehitystyön pääasiallinen tarkoitus on ylläpitää ja kehittää erilaisia asiakkaille tarjottavia tv-puolen viihderatkaisuja, kuten esimerkiksi mobiilisovellusta ja videovuokraamoja. Järjestelmä 4 toimii pilvipalveluna.

6 TUTKIMUSTULOKSET JA ANALYYSI

Tässä luvussa esitetään tutkimustulokset yritystasolla sekä järjestelmätasolla. Alaluvussa 6.1 esitetään yritystason tutkimustulokset, joissa käsitellään saavutetut kypsyystasot, ylätason ryhmien tulokset täyttyneiden tarkistuspisteiden osalta kypsyysasteittain, kunkin yksittäisen avainalueen täyttyneet tarkistuspisteet sekä täyttyneiden tarkistuspisteiden osuudet tärkeysluokittain.

Alaluvussa 6.2 perehdytään järjestelmäkohtaisiin tutkimustuloksiin, joissa esitetään kunkin järjestelmän osalta jokaisen tarkistuspisteen tulokset, tulokset ylätason ryhmittäin ja kypsyysasteittain sekä nostetaan esille haastattelijoiden kommentteja ja mielipiteitä.

6.1 Yritystason tutkimustulokset

TPI-mallissa kullekin avainalueelle on määritetty ylätason ryhmä, ja tutkimustulosten perusteella avainalueille määritellään kypsyystaso. Taulukossa 6 on kuvattu ylätason ryhmittäin, kuinka moni avainalue saavutti minkäkin kypsyystason. Taulukosta nähdään, että ylätason ryhmä *testaajien asiantuntevuus* on saanut kypsyystasoja verrattaessa parhaat tulokset saavuttaen kontrolloidun tason kahdeksalla, tehokkaan viidellä sekä optimoidun kahdella avainalueella. Nämä olivat parhaat tulokset kaikki kolme ylätason ryhmää huomioiden. *Testaajien asiantuntevuus* -ryhmällä oli myös vertailussa vähiten alkuvaiheen kypsyystasolle jääneitä avainalueita, joita oli yhteensä viisi.

Ylätason ryhmä *testauksen johtaminen* taas keräsi avainalueiden määrään suhteutettuna eniten alkuvaiheen kypsyystasolle jääneitä avainalueita sekä vähiten alkuvaihetta korkeampia kypsyystasoja, joten ylätason ryhmien saavuttamien kypsyystasojen vertailussa sen voidaan todeta olevan Yrityksen heikoiten huomioima osa-alue. *Testauksen johtaminen* sai alkuvaiheen kypsyystason avainalueita yhteensä 13 kappaletta, kontrolloituja kaksi, tehokkaita neljä ja optimoituja yhden.

Ylätason ryhmä *sidosryhmäsuhteet* sai tutkimuksessa hieman *testauksen johtamista* paremman tuloksen saavuttaen kontrolloidun kypsyystason neljästi ja tehokkaan kolmesti. *Sidosryhmäsuhteet*-ryhmässä alkuvaiheen kypsyystasolle jäi yhdeksän avainaluetta eikä optimoitua kypsyystasoa saavutettu yhdessäkään avainalueessa. Vertailussa tulee kuitenkin huomioida, että kahdesta muusta ylätason ryhmästä poiketen *sidosryhmäsuhteisiin* luokiteltiin kaikki neljä tutkittua järjestelmää huomioon ottaen yhteensä 16 avainaluetta, kun kahdessa muussa ylätason ryhmässä avainalueita oli yhteensä 20.

Taulukko 6. Saavutetut kypsyystasot ylätason ryhmittäin.

	Alkuvaihe	Kontrolloitu	Tehokas	Optimoitu
Sidosryhmäsuhteet	9	4	3	0
Testauksen johtaminen	13	2	4	1
Testaajien asiantuntevuus	5	8	5	2
Yhteensä	27	14	12	3

Taulukossa 7 on kuvattu kaikkien tutkittujen neljän järjestelmän yhteenlasketut prosenttiosuudet kolmen ylätason ryhmän täyttyneistä tarkistuspisteistä TPI-mallin mukaisesti kypsyystasoihin jaoteltuina. Yritystasolla kolmesta ylätason ryhmästä *testaajien asiantuntevuus* täytti suurimman osuuden tarkistuspisteistä kaikki kypsyystasot huomioiden 65,87 prosenttiosuudella. *Sidosryhmäsuhteet*-ryhmän kaikista tarkistuspisteistä täyttyi 60,00 %, kun taas *testauksen johtaminen* -ryhmän osuus oli ainoastaan 44,39 %.

Testaajien asiantuntevuuden kontrolloidun kypsyystason tarkistuspisteiden vaatimuksista täyttyi 92,11 %, joka on reilusti enemmän kuin *sidosryhmäsuhteiden* 73,44 % tai *testauksen johtamisen* 55,00 %. Myös tehokkaan kypsyystason tarkistuspisteitä täyttyi eniten ylätason ryhmässä *testaajien asiantuntevuus*, 64,47 %, mutta ero *sidosryhmäsuhteiden* 64,29 prosenttiosuuteen ei ollut merkittävä. Sen sijaan *testauksen johtamisessa* tehokkaan kypsyystason tarkistuspisteistä täyttyi ainoastaan 45,59 %. Optimoidun kypsyystason tarkistuspisteistä ylätason ryhmä *sidosryhmäsuhteet* täytti 35,50 %, joka on hyvin lähellä *testaajien asiantuntevuus* -ryhmän tulosta 32,14 %.

Testauksen johtaminen sai heikoimman tuloksen myös optimoidulla kypsyytasolla, jonka tarkistuspisteistä täyttyi 25,00 %.

Taulukko 7. Täyttyneiden tarkistuspisteiden osuus ylätason ryhmittäin yritystasolla.

	Sidosyhmäsuhteet	Testauksen johtaminen	Testaajien asiantuntemus
Kontrolloitu	73.44%	55.00%	92.11%
Tehokas	64.29%	45.59%	64.47%
Optimoitu	32.50%	25.00%	32.14%
Yhteensä	60.00%	44.39%	65.87%

Taulukossa 8 on eritelty yritystasolla kunkin avainalueen täyttyneiden tarkistuspisteiden määrä suhteessa tarkistuspisteiden määrään. Taulukosta nähdään, että *vikojen hallinta* ja *suorituskykytestaus* ovat tutkittujen neljän järjestelmän osalta selkeästi parhaiten hallinnassa. Sekä Pressman että Myers (Myers 2004: 105–106) nostivat *vikojen hallinnan* esille osana hyvää testausstrategiaa ja ohjelmistotestauksen suunnittelua. *Vikojen hallinnalla* todettiin olevan vaikutuksia myös projektin kustannuksiin sekä projektin etenemiseen varsinkin projekteissa, joissa on laajoja integraatioyhteyksiä. Suorituskyky taas nousi alaluvussa 2.3 esille vahvana laatutekijänä. Esimerkiksi McCall (McCall ym. 1977: 3–11) suosittelee yhdeksi mitattavaksi laatuominaisuudeksi ohjelmiston tehokkuutta.

Avainalueiden *tietoturvatestausta* (72,73 %), *projektin sitoutuminen testaukseen* (70,00 %) ja *testitapausten suunnittelu* (68,18 %) tarkistuspisteistä täyttyi lähes kolme neljästä. Kuten *suorituskykytestausta*, myös *tietoturvatestausta* voidaan pitää yhtenä tärkeänä laatuominaisuuksien tekijänä. McCallin laatuominaisuuksista *tietoturvatestausta* indikoi ohjelmiston oikeellisuutta, luotettavuutta sekä eheyttä. *Projektin sitoutuminen testaukseen* taas on vahvasti yhdistettävissä luvun 2.1 kuvassa 1 esitettyyn Tasseyn arvioon (Tassey, 2002) löydettyjen virheiden kustannuksista eri ohjelmistokehityksen vaiheissa. Mitä sitoutuneempi projekti on panostamaan koko ohjelmistokehityksen kaaren kattavaan testaukseen ja siihen valmistautumiseen, sitä todennäköisemmin projektin budjetti ei vaarannu. Avainalueen *Testitapausten suunnittelu* tarkistuspisteissä on vahvoja yhtäläisyyksiä Myersin listaan (Myers 2004: 105–106) hyvän testauksen suunnittelun ominaispiirteistä. Yrityksen käyttämässä TPI-mallissa *testitapausten*

suunnittelu ja *testauksen suunnittelu* on jaettu omiksi avainalueikseen. Avainalueen *testauksen suunnittelu* tarkistuspisteistä täyttyi 62,50 %.

Sen sijaan avainalueiden *mittaristot* sekä *työmääräarviointi ja suunnittelu* tarkistuspisteistä ei täytynyt kuin muutama, ja molempien avainalueiden osalta saavutettiin alle 10 % määritetyistä tarkistuspisteistä. Myersin listaamissa hyvän testauksen suunnittelun ominaispiirteissä nostettiin esille aikataulut ja testauksen seuranta, jotka eivät selvästi ole Yrityksen osalta tutkimustulosten perusteella täyttyneet. Loput seitsemän avainaluetta sijoittuvat vertailussa täyttyneiden tarkistuspisteiden osalta reilun 11 %:n sisään toisistaan saavuttaen 52,78–63,89 % asetetuista tarkistuspisteistä.

Taulukko 8. Täyttyneiden tarkistuspisteiden osuus avainalueittain yritystasolla.

Avainalue	Tarkistuspisteet	
Vikojen hallinta	33/40	82.50%
Suorituskykytestaus	31/40	77.50%
Tietoturvatestaus	32/44	72.73%
Projektin sitoutuminen testaukseen	28/40	70.00%
Testitapausten suunnittelu	30/44	68.18%
Testausprosessin johtaminen	23/36	63.89%
Testauksen suunnittelu	25/40	62.50%
Raportointi	24/40	60.00%
Testausautomaatio	25/44	56.82%
Kommunikaatio	20/36	55.56%
Testiympäristöt	24/44	54.55%
Testaajien ammattitaito	19/36	52.78%
Mittaristot	4/44	9.09%
Työmääräarviointi ja suunnittelu	2/36	5.56%
Yhteensä	320/564	56.74%

Taulukkoon 9 on kuvattu täyttyneiden tarkistuspisteiden prosenttiosuus kunkin luvussa 4.3 määritetyn tärkeysluokan mukaan. Taulukosta nähdään, että tärkeimmäksi määritetyn A-tärkeysluokan tarkistuspisteistä on yritystasolla täytetty suurin prosenttiosuus, kun taas vähiten tärkeän M-tärkeysluokan täyttyneiden tarkistuspisteiden prosenttiosuus on pienin.

Täyttyneiden tarkistuspisteiden osuuksien lasku siirryttäessä tärkeimmästä A-tärkeysluokasta vähiten tärkeään M-tärkeysluokkaan ei kuitenkaan ole lineaarista, vaan välillä B–L on paljon vaihtelua. Esimerkiksi I-tärkeysluokan tarkistuspisteistä täyttyi ainoastaan 18,75 %, kun taas H-tärkeysluokan tarkistuspisteistä täyttyi 62,50 % ja J-tärkeysluokan tarkistuspisteistä 60,00 %.

Taulukko 9. Täyttyneiden tarkistuspisteiden osuus tärkeysluokittain.

	A	B	C	D	E	F	G	H	I	J	K	L	M
Järjestelmä 1	83.33%	78.57%	64.71%	77.78%	66.67%	81.25%	42.86%	50.00%	0.00%	70.00%	38.46%	27.27%	16.67%
Järjestelmä 2	100.00%	78.57%	64.71%	66.67%	33.33%	75.00%	71.43%	66.67%	50.00%	80.00%	69.23%	54.55%	33.33%
Järjestelmä 3	91.67%	64.29%	70.59%	88.89%	66.67%	68.75%	57.14%	66.67%	25.00%	60.00%	30.77%	27.27%	25.00%
Järjestelmä 4	91.67%	64.29%	70.59%	88.89%	66.67%	68.75%	57.14%	66.67%	25.00%	60.00%	30.77%	27.27%	25.00%
Yhteensä	89.58%	73.21%	61.76%	77.78%	58.33%	71.88%	50.00%	62.50%	18.75%	60.00%	40.38%	29.55%	18.75%

Tarkistuspisteitä, joiden vaatimukset eivät täyttyneet yhdenkään tutkitun järjestelmän osalta, oli yhteensä 19. Niistä seitsemän kuului avainalueeseen *työmääräsuunnittelu ja arviointi* ja kahdeksan avainalueeseen *mittaristot*. Avainalueen *testaajien ammattitaito* optimoidun kypsyystason kaksi ensimmäistä tarkistuspistettä eivät myöskään täyttyneet yhdenkään järjestelmän osalta. Näiden tarkistuspisteiden vaatimukset olivat ”Testaajat osallistuvat jatkuvasti alan seminaareihin tai lukevat alan kirjallisuutta pitääkseen ammattitaitonsa korkealla” sekä ”Testaajille on organisaation sisäinen uraputki”. Kaikkien järjestelmien osalta täyttymättä jäivät myös avainalueen *testitapausten suunnittelu* optimoidun kypsyystason viimeinen tarkistuspiste ”Testitapausten suunnittelutekniikoita arvioidaan ja kehitetään jatkuvasti” sekä avainalueen *suorituskykytestaus* optimoidun kypsyystason ensimmäinen tarkistuspiste ”Käytettyä työkalua sekä testauksen suunnittelua arvioidaan suorituskykytestauksen jälkeen”.

Huomioitavaa on, että *sidosryhmäsuhteet*-ryhmästä ei löytynyt kaikki tutkitut neljä järjestelmää huomioon ottaen yhtään tarkistuspistettä, joka ei täyttynyt yhdenkään järjestelmän osalta. *Testaajien asiantuntevuus* -ryhmässä oli yhteensä neljä tarkistuspistettä, jotka eivät täyttyneet yhdenkään tutkitun järjestelmän osalta, mutta ne olivat kaikki optimoidun kypsyystason tarkistuspisteitä. *Testauksen johtaminen* -ryhmässä yhdenkään järjestelmän osalta täyttymättömiä tarkistuspisteitä oli yhteensä 15,

ja ne kaikki kuuluivat joko avainalueeseen *työmääräarviointi ja suunnittelu* tai *mittaristot*.

6.2 Järjestelmätason tutkimustulokset

Taulukossa 10 on kuvattu kunkin järjestelmän avainalueiden saavuttamien kypsyystasojen yhteenlaskettu summa. Taulukosta nähdään, että Järjestelmä 2 on jäänyt alkuvaiheen kypsyystasolle ainoastaan viidesti, kun muut järjestelmät jäivät kyseiselle kypsyystasolle kuudesti tai useammin. Järjestelmä 2 oli myös ainut järjestelmä, joka saavutti optimoidun kypsyystason: se pääsi optimoidulle tasolle kolmella eri avainalueella. Järjestelmä 1 taas jäi alkuvaiheen tasolle yhdeksällä eri avainalueella ja näyttäytyy kypsyystasovertailussa heikoimpana järjestelmänä.

Taulukko 10. Järjestelmien saavuttamat kypsyystasot.

	Alkuvaihe	Kontrolloitu	Tehokas	Optimoitu
Järjestelmä 1	9	3	2	0
Järjestelmä 2	5	2	4	3
Järjestelmä 3	6	4	4	0
Järjestelmä 4	7	5	2	0

6.2.1 Järjestelmä 1

Järjestelmän 1 osalta täyttyneet tarkistuspisteet on esitetty maalauksella korostettuna taulukossa 11. Taulukosta nähdään, että järjestelmä 1 on saavuttanut tehokkaan kypsyystason kahdella avainalueella, jotka ovat *vikojen hallinta* ja *suorituskykytestaus*. Esimerkiksi avainalueella *projektin sitoutuminen testaukseen* tehokasta tasoa ei saavutettu, koska kontrolloidun kypsyystason toinen tarkistuspiste ei täyty. Kontrolloitu kypsyystaso saavutettiin avainalueilla *testitapausten suunnittelu*, *raportointi* ja *testausprosessin johtaminen*. TPI-mallin avainalueessa *Testausprosessin johtaminen* on paljon yhtäläisyyksiä van Ewijkin (van Ewijk ym. 2013: 32–33) määritelmiin

johdonmukaisesta testauksen suunnittelusta tapauskohtaisen johtamisen sijaan. Avainalueella *raportointi* Järjestelmä 1 jäi yhden tarkistuspisteen päähän optimoidusta kypsyystasosta. TPI-mallissa *raportointi*-avainalue liittyy erityisesti riskeihin ja niiden huomioimiseen projektin päätöksenteossa, mikä nousi myös esille Ewijkin määritelmässä johdonmukaisesta johtamisesta.

Yksikään avainalue ei saavuttanut optimoitua kypsyystasoa. Avainalueet *projektin sitoutuminen testaukseen*, *kommunikaatio*, *testiympäristöt*, *testauksen suunnittelu*, *työmääräarviointi ja suunnittelu*, *mittaristot*, *testaajien ammattitaito*, *tietoturvatilasto* ja *testausautomaatio* jäivät alkuvaiheen kypsyystasolle.

Taulukko 11. Järjestelmän 1 osalta täyttyneet tarkistuspisteet.

	Kontrolloitu				Tehokas				Optimoitu		
Projektin sitoutuminen testaukseen	A*	A	B*	D*	F*	F*	G*	K	K*	M	
Kommunikaatio	B*	C	C	D	F	F*	J*	L		M	
Raportointi	A*	C*	C*	D*	F*	G*	I	J*	K*	L*	
Testiympäristöt	C*	C*	D	E*	G	H*	J*	J	L	M	
Testauksen suunnittelu	A	B*	C*	C*	F*	G*	H	J*	K*	M	
Testausprosessin johtaminen	A*	A*	B*	B*	F*	G*	H	K*		L	
Työmääräarviointi ja suunnittelu	B	C*	C	E	G	H	J	M		M	
Mittaristot	B*	B	C	D*	F	G	I	I	K	M	
Vikojen hallinta	A*	A*	B*	B*	F*	F*	H*	L	L*	L*	
Testaajien ammattitaito	C*	C*	C		G	H*	I	K	L	L	
Testitapausten suunnittelu	A*	A*	B*	E*	F*	G	J*	J*	K	M*	
Tietoturvatilasto	A*	B	D*	D*	F*	G	G*	K	K*	L	
Testausautomaatio	A*	B*	B*	C	F*	F	G	J	K	L	
Suorituskykytestaus	C*	C*	D*	D*	F*	F*	G*	J*	K	M*	

Vaikka Järjestelmä 1 saavutti kontrolloidun tai tehokkaan kypsyystason ainoastaan viidellä eri avainalueella, nähdään taulukosta, että muutamalla pienellä korjausliikkeellä voitaisiin saavuttaa tehokas tai jopa optimoitu kypsyystaso. Esimerkiksi *raportoinnissa* Järjestelmä 1 jäi yhden tarkistuspisteen päähän optimoidusta tasosta, mutta tämän takia kyseinen avainalue luokitellaan kontrolloidulle kypsyystasolle. Myös *testitapausten suunnittelu* jäi yhden tarkistuspisteen päähän tehokkaasta kypsyystasosta, vaikka kyseinen avainalue sai haastateltavalta erityistä kiitosta esimerkiksi negatiivisten testitapausten suunnittelun osalta.

Yritystasolla vähiten tarkistuspisteitä täyttäneet osa-alueet *työmääräarviointi ja suunnittelu* sekä *mittaristot* ovat täyttäneet myös Järjestelmässä 1 vain yhdestä kahteen tarkistuspistettä. Haastateltava kommentoi työmääräarviointia testauksen osalta: ”Ei

tehdä, ei olla koettu tarpeelliseksi, mutta voisi auttaa suunnittelussa varsinkin pidemmällä tähtäimellä.” Myös *kommunikaatio*-avainalueen täyttyneiden tarkistuspisteiden määrä on pieni varsinkin kontrolloidun kypsyystason tarkistuspisteiden osalta.

Taulukossa 12 on vertailtu Järjestelmän 1 ylätasoin ryhmien täyttyneiden tarkistuspisteiden määrää suhteessa kaikkiin tutkittuihin järjestelmiin. Taulukosta nähdään, että *sidosryhmäsuhteet*-ryhmän osalta Järjestelmä 1 on hyvin lähellä keskiarvoa. Kontrolloidun kypsyystason tarkistuspisteistä on täyttynyt hieman keskiarvoa vähemmän, kun taas tehokkaalla kypsyystasolla täyttyneitä tarkistuspisteitä on hieman keskiarvoa enemmän.

Taulukko 12. Järjestelmän 1 ylätasoin ryhmien täyttyneiden tarkistuspisteiden vertailu koko Yritykseen.

Kypsyystaso	Sidosryhmäsuhteet		Testauksen johtaminen		Testaajien asiantuntemuus	
	Järjestelmä 1	Yritys	Järjestelmä 1	Yritys	Järjestelmä 1	Yritys
Kontrolloitu	68.75%	73.44%	70.00%	55.00%	84.21%	92.11%
Tehokas	71.43%	64.29%	47.06%	45.59%	57.89%	64.47%
Optimoitu	30.00%	32.50%	33.33%	25.00%	21.43%	32.14%
Yhteensä	60.00%	60.00%	53.06%	44.39%	57.69%	65.87%

Ylätasoin ryhmän *testauksen johtaminen* osalta Järjestelmä 1 pärjäsi hyvin vertailussa koko Yritykseen. Kontrolloidun kypsyystason tarkistuspisteistä täyttyi 15 % enemmän kuin keskimäärin, ja tehokkaalla sekä optimoidulla kypsyystasolla täytettiin keskimääräistä enemmän tarkistuspisteitä. *Testaajien asiantuntemuudessa* Järjestelmä 1 jäi kaikilla kolmella kypsyystasolla Yrityksen keskimääräisistä vertailuluvuista jälkeen, huomattavimmin optimoidulla kypsyystasolla, jossa keskiarvosta jäätin yli 10 % päähän.

6.2.2 Järjestelmä 2

Järjestelmän 2 osalta täyttyneet tarkistuspisteet on esitetty maalauksella korostettuna taulukossa 13. Taulukosta nähdään, että Järjestelmä 2 on saavuttanut optimoidun kypsyystason avainalueissa *testausprosessin johtaminen*, *tietoturvatestaus* ja *testiautomaatio*. Tehokkaan kypsyystason Järjestelmä 2 on saavuttanut *raportoinnissa*,

vikojen hallinnassa, testitapausten suunnittelussa sekä suorituskykytestauksessa. Kontrolloidulla tasolla ovat avainalueet *testauksen suunnittelu* ja *testaajien ammattitaito*, joten alkuvaihetasolle jää viisi eri avainaluetta. Järjestelmä 2:n osalta tulee kuitenkin huomioida, että sen toimintamalli eroaa huomattavasti Yrityksen muista järjestelmistä. Muista järjestelmistä poiketen Järjestelmä 2:n osalta kaikki, myös testaus, on ostettu järjestelmän toimittajalta palveluna. Tämä vaikutti selkeästi avainalueiden *projektin sitoutuminen testaukseen* sekä *kommunikaatio* pisteytykseen TPI-mallissa.

Taulukko 13. Järjestelmän 2 osalta täyttyneet tarkistuspisteet.

	Kontrolloitu				Tehokas				Optimoitu		
Projektin sitoutuminen testaukseen	A*	A*	B*	D	F*	F	G		K*	K*	M
Kommunikaatio	B*	C	C	D	F	F	J*		L		M
Raportointi	A*	C*	C*	D*	F*	G*	I*	J*	K		L
Testiympäristöt	C	C*	D*	E	G	H*	J*	J*	L*	M*	M*
Testauksen suunnittelu	A*	B*	C*	C*	F*	G*	H*	J	K*		M*
Testausprosessin johtaminen	A*	A*	B*	B*	F*	G*	H*		K*		L*
Työmääräarviointi ja suunnittelu	B	C	C	E	G	H	J		M		M
Mittaristot	B	B	C	D	F	G	I	I	K	M	M
Vikojen hallinta	A*	A*	B*	B*	F*	F*	H*		L*	L*	L
Testaajien ammattitaito	C*	C*	C*		G*	H	I*		K	L	L
Testitapausten suunnittelu	A*	A*	B*	E*	F*	G*	J*	J*	K*	M	M
Tietoturvatestaus	A*	B*	D*	D*	F*	G*	G*	G*	K*	K*	L*
Testausautomaatio	A*	B*	B*	C*	F*	F*	G*	J*	K*	K*	L*
Suorituskykytestaus	C*	C*	D*	D*	F*	F*	G*	J*	K		M*

Järjestelmä 2 oli ainoa tutkituista järjestelmistä, joka saavutti optimoidun kypsyyden tasolla yhdelläkään avainalueella, ja saavutti sen peräti kolmella eri avainalueella. Kolmen optimoidun avainalueen lisäksi kolme muuta avainaluetta jäi yhden tarkistuspisteen päähän optimoidusta tasosta. Haastateltavan vastaukset erityisesti *testausautomaatio*-avainalueeseen olivat hyvin vakuuttavia: Järjestelmä 2:lla pyörii ajastetusti yli 2000 testitapausta, joiden onnistumisprosenttia monitoroidaan säännöllisesti. Ohjelmiston testattavuuden tulee olla hyvällä tasolla, jotta testiautomaation rakentaminen on mahdollista. Testattavuus on myös yksi McCallin (McCall ym. 1977: 3–11) määrittämistä laatuominaisuuksista.

Kuten muidenkin järjestelmien osalta, avainalueet *työmääräarviointi ja suunnittelu* sekä *mittaristot* jäivät arvioinnin mukaan Järjestelmässä 2 heikoimmalle tasolle. Haastateltavan mukaan testauksen työmääräarvioinnille ei koeta olevan tarvetta. Testauksen etenemistä seuraavan mittariston rakentamisen todettiin olevan mahdollista,

mutta tarvittaessa saatavat suulliset raportit koetaan riittäviksi. Avainalue *testiympäristöt* jää tutkimuksessa Järjestelmän 2 osalta hieman harhaanjohtavasti alkuvaihe-tasolle, sillä 11 tarkistuspisteestä täyttyi yhteensä kahdeksan. *Testiympäristöt* nousivat esille tutkielman taustamateriaalissa Myersin (Myers 2004: 105–106) hyvän testauksen suunnittelun ominaispiirteissä, joissa on mainittu riittävän infrastruktuurin tärkeys. *Testiympäristöjä* voidaan pitää myös mahdollistajina useille McCallin laatuominaisuuksille (McCall ym. 1977: 3–11), esimerkiksi ylläpidettävyydelle ja testattavuudelle.

Taulukossa 14 on vertailtu Järjestelmän 2 ylätasoin ryhmien täyttyneiden tarkistuspisteiden määrää suhteessa kaikkiin tutkittuihin järjestelmiin. Taulukosta nähdään, että Järjestelmä 2 menestyy vertailussa muihin järjestelmiin lähes poikkeuksetta positiivisesti. Ainoastaan *sidosryhmäsuhteet*-ryhmän kontrolloidun kypsyystason tarkistuspisteiden osalta Järjestelmä 2 on saanut keskiarvoa heikommat tulokset, jääden keskiarvosta hieman yli 10 prosenttia.

Taulukko 14. Järjestelmän 2 ylätasoin ryhmien täyttyneiden tarkistuspisteiden vertailu koko Yritykseen.

	Sidosryhmäsuhteet		Testauksen johtaminen		Testaajien asiantuntemus	
	Järjestelmä 2	Yritys	Järjestelmä 2	Yritys	Järjestelmä 2	Yritys
Kypsyystaso						
Kontrolloitu	62.50%	73.44%	60.00%	55.00%	100.00%	92.11%
Tehokas	64.29%	64.29%	52.94%	45.59%	94.74%	64.47%
Optimoitu	50.00%	32.50%	50.00%	25.00%	57.14%	32.14%
Yhteensä	60.00%	60.00%	55.10%	44.39%	86.54%	65.87%

Taulukosta nähdään myös, että Järjestelmä 2 on saanut muihin järjestelmiin verrattuna paremmat pisteet kaikissa ylätasoin ryhmässä erityisesti optimoidun kypsyystason tarkistuspisteissä. Sekä *testauksen johtamisessa* että *testaajien asiantuntemuudessa* optimoidun kypsyystason tarkistuspisteistä täyttyi Järjestelmä 2:n osalta 25 % keskiarvoa enemmän. *Testaajien asiantuntemuuden* tarkistuspisteistä täyttyi tehokkaalla kypsyystasolla yli 30 % enemmän muihin järjestelmiin verrattuna, ja koko ylätasoin ryhmän osalta tarkistuspisteistä täyttyi yli 20 % enemmän keskiarvoon nähden.

6.2.3 Järjestelmä 3

Taulukossa 15 on esitetty Järjestelmän 3 osalta täyttyneet tarkistuspisteet. Taulukosta nähdään, että Järjestelmä 3 saavutti tehokkaan kypsyystason avainalueissa *testiympäristöt, vikojen hallinta ja suorituskypyttestaus*. Tehokas kypsyystaso saavutettiin myös avainalueella *testaajien ammattitaito*, jossa Järjestelmä 3 täytti kaikkien tutkittujen järjestelmien osalta eniten tarkistuspisteitä. Sekä Pressman (Pressman ym. 2012: 466–467) että Myers (Myers 2004: 16) painottavat testaajan roolin tärkeyttä. He korostavat erityisesti, että testaus tulee suorittaa erillisen testausryhmän toimesta ja että testaus on luovuutta vaativa toimenpide.

Kontrolloitu kypsyystaso saavutettiin *projektin sitoutumisessa testaukseen, raportoinnissa, testitapausten suunnittelussa* sekä *testiautomaatiossa*. Huomattavaa kuitenkin on, että Järjestelmä 3 jäi yhden tarkistuspisteen päähän optimoidusta kypsyystasosta sekä *projektin sitoutumisessa testaukseen* että *kommunikaatiossa*. Tästä johtuen *projektin sitoutumisen testaukseen* tulkittiin TPI-mallin mukaisesti olevan kontrolloidulla kypsyystasolla ja *kommunikaation* alkuvaiheen kypsyystasolla. Erityisesti *projektin sitoutuminen testaukseen* sai haastateltavalta kiitosta, sillä hän kehui Järjestelmän 3 ohjelmistokehitykseen sitoutuneiden jäsenten yhteistyötä ja valmiutta auttaa työkavereita erinomaiseksi.

Kuten muutkin tutkitut järjestelmät, Järjestelmä 3 sai heikoimmat tulokset avainalueissa *työmääräarviointi ja suunnittelu* sekä *mittaristot*, mitkä myös haastateltava tunnisti selkeästi Järjestelmän 3 testausprosessin heikkouksiksi. Näillä avainalueilla Järjestelmä 3 jäi alkuvaiheen kypsyystasolle. Alkuvaiheen tasolle jäätiin myös *testausprosessin johtamisessa* ja *tietoturvatestauksessa*, jotka jäivät kahden tarkistuspisteen päähän kontrolloidusta kypsyystasosta. Alkuvaiheen kypsyystasolle jäivät lisäksi avainalueet *testauksen suunnittelu* ja *tietoturvatestaus*.

Taulukko 15. Järjestelmän 3 osalta täyttyneet tarkistuspisteet.

	Kontrolloitu				Tehokas				Optimoitu		
Projektin sitoutuminen testaukseen	A*	A*	B*	D*	F	F*	G*	K*	K*	M*	
Kommunikaatio	B*	C*	C	D*	F*	F*	J*	L*		M*	
Raportointi	A*	C*	C*	D*	F	G	I	J	K	L	
Testiympäristöt	C*	C*	D*	E*	G*	H*	J*	J*	L	M	M
Testauksen suunnittelu	A*	B*	C	C*	F	G*	H*	J	K*	M	
Testausprosessin johtaminen	A	A*	B*	B	F*	G*	H	K		L	
Työmääräarviointi ja suunnittelu	B	C	C	E	G	H	J	M		M	
Mittaristot	B	B	C	D*	F	G	I	I	K	M	M
Vikojen hallinta	A*	A*	B*	B*	F*	F*	H*	L	L*	L	
Testaajien ammattitaito	C*	C*	C*	C*	G*	H*	I*	K	L	L*	
Testitapausten suunnittelu	A*	A*	B*	E*	F*	G	J*	J	K	M	M
Tietoturvatestaus	A*	B	D*	D	F*	G*	G	G*	K	K*	L
Testausautomaatio	A*	B*	B*	C*	F	F*	G	J*	K	K	L
Suorituskykytestaus	C*	C*	D*	D*	F*	F*	G*	J*	K		M*

Taulukossa 16 on kuvattu Järjestelmän 3 täyttyneiden tarkistuspisteiden määrä verrattuna koko Yrityksen tuloksiin. Taulukosta nähdään, että *sidosryhmäsuhteet*-ryhmässä Järjestelmä 3 pärjää vertailussa vahvasti. Kontrolloidulla kypsyytasolla ero koko Yrityksen täyttyneisiin tarkistuspisteisiin on yli 20 prosenttia ja optimoidulla tasolla 17,5 prosenttia. Ylätason ryhmässä *testaajien asiantuntemys* Järjestelmä 3 sijoittui vertailussa kaikki kypsyytasot huomioiden hyvin lähelle keskiarvoa, vaikka eroavaisuuksia kypsyytasojen välillä olikin. Ylätason ryhmä *testauksen johtaminen* on selkeästi Järjestelmän 3 heikoin osa-alue, sillä se jäi kaikilla kypsyytasolla alle koko Yrityksen keskiarvon.

Taulukko 16. Järjestelmän 3 ylätason ryhmien täyttyneiden tarkistuspisteiden vertailu koko Yritykseen.

	Sidosryhmäsuhteet		Testauksen johtaminen		Testaajien asiantuntemys	
	Järjestelmä 3	Yritys	Järjestelmä 3	Yritys	Järjestelmä 3	Yritys
Kypsyytaso	93.75%	73.44%	50.00%	55.00%	89.47%	92.11%
Kontrolloitu	93.75%	73.44%	50.00%	55.00%	89.47%	92.11%
Tehokas	64.29%	64.29%	41.18%	45.59%	73.68%	64.47%
Optimoitu	50.00%	32.50%	16.67%	25.00%	21.43%	32.14%
Yhteensä	72.50%	60.00%	38.78%	44.39%	65.38%	65.87%

6.2.4 Järjestelmä 4

Järjestelmä 4:n osalta täyttyneet tarkistuspisteet on esitetty taulukossa 17. Järjestelmä 4 saavutti tehokkaan kypsyytason kahdella avainalueella, jotka olivat *vikojen hallinta* ja *kommunikaatio*. TPI-mallissa avainalueella *kommunikaatio* tarkoitetaan pääasiassa testaustoimenpiteitä suorittavien työntekijöiden esteetöntä pääsyä kaikkeen vaadittavaan

dokumentaatioon ja mahdollisuutta viestiä löydöksistä ja epäselvistä tilanteista oleellisille tahoille. Heikko kommunikaatio voi johtaa herkästi muun muassa Haasin (Haas 2008: 137) esiin nostamiin aikataulullisiin viivästyksiin.

Kontrolloitu kypsyystaso saavutettiin avainalueilla *projektin sitoutuminen testaukseen, testaajien ammattitaito, testitapausten suunnittelu, tietoturvatestaus ja testiautomaatio*. Loput seitsemän avainaluetta jäivät haastattelutulosten perusteella alkuvaiheen kypsyystasolle.

Taulukko 17. Järjestelmän 4 osalta täyttyneet tarkistuspisteet.

	Kontrolloitu				Tehokas				Optimoitu		
Projektin sitoutuminen testaukseen	A*	A*	B*	D*	F*	F*	G		K	K	M
Kommunikaatio	B*	C*	C*	D*	F*	F*	J*		L		M
Raportointi	A*	C	C	D	F	G*	I	J*	K		L
Testiympäristöt	C	C	D*	E*	G	H*	J	J	L	M	M
Testausprosessin suunnittelu	A	B	C	C*	F*	G	H*	J	K		M
Testausprosessin johtaminen	A*	A	B*	B	F*	G	H		K		L
Työmääräarviointi ja suunnittelu	B*	C	C	E	G	H	J		M		M
Mittaritot	B	B	C	D	F	G*	I	I	K	M	M
Vikojen hallinta	A*	A*	B*	B*	F*	F*	H*		L	L	L
Testaajien ammattitaito	C*	C*	C*	C*	G	H	I		K	L	L
Testitapausten suunnittelu	A*	A*	B*	E*	F*	G*	J	J	K*	M	M
Tietoturvatestaus	A*	B*	D*	D*	F*	G*	G	G	K*	K*	L*
Testausautomaatio	A*	B*	B*	C*	F	F	G	J	K	K	L
Suorituskykytestaus	C*	C	D*	D*	F	F	G	J*	K		M

Avainalueet *työmääräarviointi ja suunnittelu* sekä *mittaritot* olivat muiden tutkittujen järjestelmien tavoin Järjestelmä 4:n kaksi heikointa avainaluetta. Muista järjestelmistä poiketen Järjestelmällä 4 oli kahden edellä mainitun avainalueen lisäksi kaksi muuta avainaluetta, joiden tarkistuspisteistä täyttyi ainoastaan kolme tai vähemmän. Muissa tutkituissa järjestelmissä tällaisia avainalueita oli yhteensä ainoastaan kaksi. Haastateltava totesi kiireen säätelevän pitkälti tekemistä, mikä on resurssipulan myötä johtanut siihen, että kaikkea haluttua ei aina saada tehtyä. Haastateltava toivoi myös, että etenkin suorituskykytestauksessa panostettaisiin enemmän ulkoiseen apuun.

Taulukossa 18 on kuvattu Järjestelmän 4 täyttyneiden tarkistuspisteiden määrä verrattuna koko Yrityksen tuloksiin. Ylätason ryhmissä *sidosryhmäsuhteet* sekä *testausprosessin johtaminen* Järjestelmä 4 jää vertailussa alle keskiarvon. Selkeimmin ero muihin järjestelmiin korostuu optimoidulla kypsyystasolla, jolla Järjestelmä 4 ei täyttänyt yhtäkään tarkistuspistettä kummassakaan ylätason ryhmässä.

Taulukko 18. Järjestelmän 4 ylätason ryhmien täyttyneiden tarkistuspisteiden vertailu koko Yritykseen.

Kypsyystaso	Sidosryhmäsuhteet		Testauksen johtaminen		Testaajien asiantuntemuus	
	Järjestelmä 4	Yritys	Järjestelmä 4	Yritys	Järjestelmä 4	Yritys
Kontrolloitu	68.75%	73.44%	40.00%	55.00%	94.74%	92.11%
Tehokas	57.14%	64.29%	41.18%	45.59%	31.58%	64.47%
Optimoitu	0.00%	32.50%	0.00%	25.00%	28.57%	32.14%
Yhteensä	47.50%	60.00%	30.61%	44.39%	53.85%	65.87%

Sen sijaan ylätason ryhmässä *testaajien asiantuntemuus* Järjestelmä 4 sai Yrityksen keskiarvoa paremmat tulokset kontrolloidun kypsyystason tarkistuspisteissä ja pääsi melko lähelle keskiarvoa optimoidulla tasolla. Toisaalta tehokkaalla kypsyystasolla Järjestelmä 4 jäi koko Yrityksen keskiarvosta yli 30 % päähän.

7 DISKUSSIO

Tutkielman tavoitteena oli arvioida Yrityksen ohjelmistotestauksen laatua ja löytää tutkimustulosten perusteella järjestelmäkohtaisia testausprosessin kehityskohteita sekä selkeitä yrityskohtaisia testausprosessien heikkouksia. Tutkimus perustui TPI-malliin, jota muokkaamalla luotiin Yrityksen tarpeita vastaava viitekehys testausprosessin kypsyysasteen määrittämiseksi. Tutkimusta varten muodostettu TPI-mallin mukainen viitekehys Yrityksen järjestelmäkohtaisen testausprosessin kypsyysasteen määrittämiseksi on esitetty tarkistuspistetasolla liitteessä 2. Tutkimustavoite saavutettiin perehtymällä TPI-analyysiin ja ohjelmistotestaukseen liittyvään kirjallisuuteen sekä suorittamalla tapaustutkimus, johon kerättiin tutkimusaineistoa puolistrukturoiduilla haastatteluilla järjestelmien järjestelmävastaavien ja omistajien kanssa.

TPI-analyysin avulla saatuihin tuloksiin vaikuttavat aina vahvasti tutkittavan yrityksen ja työntekijöiden kulttuuri sekä toimiala, eivätkä ne näin ollen ole yleistettävissä laajemmin. Tutkimuksen kohteena ollut Yritys voi kuitenkin halutessaan hyödyntää järjestelmäkohtaisia tutkimustuloksia sisäisesti testauksen laadun parantamiseen ja käyttää heidän tarpeisiinsa räätälöityä TPI-mallin viitekehystä myös tutkimuksen ulkopuolisten järjestelmien testausprosessin kypsyysasteen selvittämiseen. Aiempia julkisia julkaisuja TPI-analyysiin pohjautuvia tutkimuksista ei ole saatavilla.

Selkeimmät yrityskohtaiset testausprosessien heikkoudet löytyivät haastattelutulosten perusteella TPI-mallin määrittämistä avainalueista *mittaristot* sekä *työmääräarviointi ja suunnittelu*. Ne saivat yli 40 % vähemmän täyttyneitä tarkistuspisteitä kuin kolmanneksi vähiten saanut avainalue. Nämä avainalueet nousivat haastatteluissa myös erittäin selkeästi esille aiheina, joihin ei Yrityksen testausprosesseissa käytetä aikaa eikä resursseja.

Kolmesta tutkitusta ylätason ryhmästä, *sidosryhmäsuhteet, testauksen johtaminen ja testaajien asiantuntevuus*, heikoimmat tutkimustulokset sai *testauksen johtaminen*. Tämä johtui kuitenkin siitä, että kyseinen ylätason ryhmä sisälsi molemmat tutkimuksessa heikoimmat tulokset saaneet avainalueet eli *mittaristot* sekä *työmääräarvioinnin ja suunnittelun*. Muiden avainalueiden osalta *testauksen johtaminen* sai tutkimuksessa enemmän täyttyneitä tarkistuspisteitä kuin kaksi muuta ylätason ryhmää ja menestyi siis keskimääräistä paremmin.

Tutkimustuloksista nähdään, että Järjestelmät 1, 2 ja 3 saivat tutkimuksessa huomattavasti paremmat tulokset kuin Järjestelmä 4. Vaikka varsinaisten testausprosessin korjaustoimenpiteiden ehdottaminen rajattiin tutkimuksen ulkopuolelle, tutkimustuloksista nähdään selkeästi, että Järjestelmien 1, 2 ja 3 osalta korjaustoimenpiteet voidaan keskittää muutamaaan avainalueeseen tai jopa yksittäisiin tarkistuspisteisiin, kun taas Järjestelmän 4 osalta korjaustoimenpiteet testausprosessin tehostamiseksi olisivat huomattavasti laajemmat. Tarkat tutkimustulokset on esitetty luvussa 6.

Avainalueiden *mittaristot* sekä *työmääräarviointi ja suunnittelu* lisäksi järjestelmäkohtaisia heikkouksia löytyi kustakin tutkitusta järjestelmästä. Järjestelmien 1 ja 2 osalta selkeästi heikoin avainalue oli *kommunikointi*. Järjestelmän 3 kohdalla kontrolloidun kypsyystason tarkistuspisteitä jäi täyttymättä *testauksen johtamisen ja tietoturvatestauksen* osalta. Sen sijaan Järjestelmästä 4 löytyi useampi avainalue, jossa kontrolloidun kypsyystason tarkistuspisteitä jäi täyttymättä kaksi tai useampia. Tällaisia avainalueita olivat *raportointi, testiympäristöt, testauksen suunnittelu ja testausprosessin johtaminen*.

Tutkimustuloksissa on paljon yhteneväisyyksiä esimerkiksi Brian Marickin artikkeliin ”Classic Testing Mistakes” (Marick 1997), jossa hän erittelee tyypillisiä ohjelmistotestauksen ongelmia. Marick listaa työmääräarvioinnin yhdeksi viidestä klassisesta virheestä, ja kuten tämän tutkimuksen tuloksista käy ilmi, työmääräarviointi oli myös Yrityksen heikoiten menestynyt avainalue. Muut Marickin listaamat virheet ovat rinnastettavissa tutkimuksessa käytettyihin avainalueisiin *testaajien ammattitaito*,

testitapausten suunnittelu ja testausprosessin johtaminen. Näistä testaaajien ammattitaito täytti yritystasolla vain hieman yli puolet tarkistuspisteistä, *testitapausten suunnittelu* ja *testausprosessin johtaminen* taas täyttivät hieman yli 60 prosenttia tarkistuspisteistä. Marickin viides teema, teknologian levinneisyys, ei ole rinnastettavissa mihinkään tässä tutkimuksessa käytettyyn avainalueeseen.

Tämän tutkimuksen suurimmaksi haasteeksi muodostui haastattelutilanteiden kulku. Tutkimuksen tehnyt haastattelija tunsii haastateltavat ja heidän työtaustansa entuudestaan, joten joidenkin kysymysten osalta vastaukset olivat ajoittain humoristisia ja ”kyllähän sä tiedät” -tyyppisiä, mikä hankaloitti varsinaisten perusteluiden saamista. Muutamien tarkistuspisteiden osalta vastausta jouduttiin hakemaan useammalta henkilöltä tai sopimuspapereista.

Tutkimuksessa haettiin järjestelmistä eroavaisuuksia sekä järjestelmäarkkitehtuurin osalta että liiketoiminnallisten osastojen väliltä, mutta johtopäätöksiä testausprosessia parhaiten tukevasta järjestelmäarkkitehtuuriratkaisusta tai liiketoimintayksiköstä ei voida näin pienellä järjestelmäotannalla vetää. Jatkotutkimuksia ajatellen olisi mielenkiintoista tehdä tutkimus laajemmalla otannalla ja selvittää, vaikuttaako esimerkiksi järjestelmäarkkitehtuuri testausprosessin laatuun. Myös tutkimuksen uudelleen teettäminen vuoden päästä olisi mielenkiintoista, ja vuosittainen uusiminen voisi varmasti johtaa Yrityksen testausprosessien kehittymiseen ainakin siihen sitoutuvien yksittäisten järjestelmien osalta.

LÄHDELUETTELO

- Aaltola, J. & R. Valli (2007). *Ikkunoita tutkimusmetodeihin 2*. 2. painos. 18 s. Helsinki: WSOY. ISBN 9524511657
- Fast Company (2012). How One Second Could Cost Amazon \$1.6 Billion In Sales [online]. [12.12.2017]. Saatavissa: <https://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales>
- International Software Qualifications Board (2011). Certified Tester Foundation Level Syllabus [online]. [15.1.2017]. Saatavissa: <http://www.istqb.org/downloads/send/2-foundation-level-documents/3-foundation-level-syllabus-2011.html4>
- International Software Qualifications Board (2011). What is Software Quality? [online]. [6.1.2018]. Saatavissa: <http://istqbexamcertification.com/what-is-software-quality/>
- Jonassen Hass, A.M (2008). *Guide to Advanced Software Testing*. Artech House Books. ISBN 9781596932852
- Kan, S.H (2004). *Metrics and Models in Software Quality Engineering*. 2. painos. 1 s. Adisson-Wesley. ISBN 0201729156
- Kasurinen, J. O. Taipale & K. Smolander (2010). *Software Test Automation in Practice: Empirical Observations*
- Marick, Brian. (1997). *Classic Testing Mistakes*. [online]. [15.2.2018]. Saatavissa: http://sceweb.sce.uhcl.edu/helm/TesterTestingCourse/myfiles/Module10/15_TST_170_AppendixC.pdf
- McCall, J. P. Richards & G. Walters (1977) *Factors in Software Quality*. 1. painos. [online]. [6.1.2018]. Saatavissa: <http://www.dtic.mil/dtic/tr/fulltext/u2/a049014.pdf>

- Myers G.J. (2004). *The Art of Software Testing*. 2. painos. John Wiley & Sons. ISBN 0-471-46912-2
- Pressmann, S. & B. Maxim (2012). *Software Engineering a Practioner's Approach*. 8. painos. McGraw Hill education. ISBN 978-0-07-802212-8
- Robson, C (2002). *Real World Research*. 2. painos. Blackwell. 270-274 s. ISBN 0631213058
- Saunders, M., P. Lewis & A. Thornhill (2007). *Research Methods For Business Students*. 4. painos. FT Pretince Hall. SBN 0273701487
- Tassey, G. (2002). *The Economic Impacts of Inadequate Infrastructure*. U.S National Institute of Standards and Technology.
- Torrington, D., L. Hall & S. Taylor. (2008). *Human Resource Management*. 7. painos. FT Pretince Hall. 47. ISBN 0273710753
- van Ewijk, A., B. Linker, M. van Oosterwijk, B. Visser, G. de Vries, L. Wihelmus & R. Marselis. *TPI NEXT Business Driven Test Process Improvement* (2013). Sogeti. ISBN 9789075414721

LIITTEET

Liite 1

Omistajien sitoutuminen

Tarkistuspisteet kontrolloidulla tasolla:

1. Palvelun pääasiallinen omistaja on määritetty ja testaaajien tiedossa.
2. Testaajiin sidottu budjetti on määritetty, ja siitä voi tarvittaessa neuvotella palvelun omistajan kanssa.
3. Testaukseen sidotut resurssit ovat aidosti käytettävissä.
4. Palvelun pääasiallinen omistaja on vastuussa dokumentoidusta riskianalyysistä, joka on johdettu testaus suunnitelmasta.

Tarkistuspisteet tehokkaalla tasolla:

1. Kaikki relevantit omistajat on määritetty ja testaaajien tiedossa.
2. Omistajat hankkivat aktiivisesti tietoja testausprosessien laadusta ja testauksen tavoitteista.
3. Omistajat myötävaikuttavat proaktiivisesti testausprosesseihin vaikuttaviin asioihin, kuten esimerkiksi testattavien ominaisuuksien toimittamiseen tai projektin vaiheen sisältöön.

Tarkistuspisteet optimoidulla tasolla:

1. Johto tiedostaa, että testausprosessien kehitys tarvitsee aikaa ja resursseja.
2. Omistajat ovat valmiita muokkaamaan projektin työskentelytapoja – esimerkiksi vaatimustenmäärittelyä – testausprosessien kehittämiseksi.

3. Mukautettuja toimintatapoja arvioidaan yhdessä omistajien ja testausorganisaation kanssa.

Sitoutuminen

Tarkistuspisteet kontrolloidulla tasolla:

1. Testaustehtävät, -laajuus ja lähestyminen sovitaan projektin alussa omistajien kanssa.
2. Testaukseen liittyvät toimenpiteet aloitetaan hyvissä ajoin ennen varsinaisen testauksen suorittamista.
3. Testaajat ovat mukana projektin suunnittelussa ja ymmärtävät riippuvuudet muihin prosesseihin ja järjestelmiin.
4. Testaajat ovat mukana projektin riskien analysoimisessa ja pyrkivät estämään niitä muun projektiryhmän kanssa.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaajat ovat mukana muutospyyntöjen riskianalyyssissä ja ymmärtävät niiden vaikutukset testaustehtäviin.
2. Testaajat ottavat kantaa virheiden vakavuusasteisiin.
3. Testaajat optimoivat jatkuvasti testaus suunnitelmaa.

Tarkistuspisteet optimoidulla tasolla:

1. Testaajat ovat mukana projektin arvioinnissa. Projektin aikana opittuja asioita käytetään tulevien projektien testausprosessien kehittämiseen.
2. Testausorganisaatio on tärkeä osa kehitysaktiviteetteja, ja sitä arvostetaan sekä sen näkökulmia kunnioitetaan.

Testausstrategia

Tarkistuspisteet kontrolloidulla tasolla:

1. Pääasiallinen omistaja hyväksyy dokumentoidun testaussuunnitelman.
2. Testaussuunnitelmassa otetaan huomioon riskianalyysi.
3. Testaustasot, -tavat ja testauksen kattavuus on huomioitu.
4. Uudelleen testauksen ja regressiotestauksen toimintatavat on suunniteltu.

Tarkistuspisteet tehokkaalla tasolla:

1. Kaikki relevantit omistajat hyväksyvät dokumentoidun testaussuunnitelman.
2. Testaustapojen ja testauksen kattavuuden erot eri testaustasoilla on tarkkaan huomioitu.
3. Testaussuunnitelma sisältää tarkkaan valitut testausmetodit.

Tarkistuspisteet optimoidulla tasolla:

1. Testaussuunnitelman luomisen prosessia tarkastellaan ja parannetaan jatkuvasti.
2. Testaussuunnitelmaa arvioidaan tuotannon virhedatan perusteella.

Testausorganisaatio

Tarkistuspisteet kontrolloidulla tasolla:

1. Muu organisaatio tietää, miten testausorganisaation palvelut ovat saatavilla.
2. Testausorganisaatiolla on selkeät vastualueet.
3. Tehtävät ja vastuut on jaettu, dokumentoitu sekä osoitettu vastuullisille henkilöille tai ryhmille.
4. Testausorganisaation palvelut ja työkalut ovat selviä muulle organisaatiolle.

Tarkistuspisteet tehokkaalla tasolla:

1. Testauspäälliköt ja testaajat pystyvät itseohjautuvasti organisoimaan työnsä.
2. Testausosastot pystyvät tarjoamaan tarvittavat resurssit ja palvelut projekteille.
3. Testausorganisaatio on yritysraenteessa hyvin harkitussa paikassa.
4. Testausorganisaatio noudattaa yhtenäisiä sääntöjä.

Tarkistuspisteet optimoidulla tasolla:

1. Testausorganisaation käyttämiä työkaluja ja sen tarjoamia palveluja arvioidaan jatkuvasti sekä tarvittaessa harkitaan uusia kustannustehokkaita ratkaisuja.
2. Testausorganisaatiota pidetään vastuussa työstään.
3. Testausorganisaation suoritusta verrataan jatkuvasti ulkoisiin toimittajiin tai vastaaviin organisaatioihin.

Kommunikaatio

Tarkistuspisteet kontrolloidulla tasolla:

1. Jokainen projektin jäsen on tietoinen tehdyistä päätöksistä ja kehityksen vaiheista.
2. Testaajat hankkivat aktiivisesti tietoja projektin omistajilta.
3. Testaajien tekemät toimenpiteet ja päätökset dokumentoidaan.
4. Testaajat kommunikoivat omistajien kanssa aktiivisesti kehityksestä, ohjelmiston laadusta ja mahdollisista uhkista.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaajat tunnistavat, mitä tietoja ja havaintoja tulee kommunikoida omistajille.
2. Testaajat osallistuvat oleellisiin kokouksiin omistajien kanssa.
3. Testaajilla on tarvittavat keinot kommunikoida omistajien kanssa virallisia kaavakkeita käyttäen tai epävirallisesti.

Tarkistuspisteet optimoidulla tasolla:

1. Parhaita kommunikoinnin opittuja käytäntöjä arvioidaan ja kehitetään jatkuvasti sekä kuluvaan että tulevia projekteja varten.
2. Organisaatio tutkii ja kokeilee uusia tapoja kommunikoida sekä tarvittaessa määrittää käytäntöjä.

Raportointi

Tarkistuspisteet kontrolloidulla tasolla:

1. Raportoinnissa huomioidaan aika, kustannukset, tulokset sekä riskit.
2. Raportoinnin tiheys on riittävä tukemaan omistajien päätöksentekoa.
3. Raportointi tehdään kirjallisesti.

Tarkistuspisteet tehokkaalla tasolla:

1. Omistajien vaatimukset täyttävän raportoinnin edellyttämä työpanos on tasapainossa raportointiin käytettävän työpanoksen kanssa.
2. Raportoinnissa huomioidaan testausprossin edistyminen ja projektin riskit ja tehdään suosituksia niihin perustuen.
3. Raportoinnissa huomioidaan testauksen tavoitteet ja tuote- sekä ohjelmistokohtaiset riskit ja tehdään suosituksia niihin perustuen.

Tarkistuspisteet optimoidulla tasolla:

1. Raportoinnilla tuotetaan dataa, jonka avulla voidaan kehittää nykyistä ja tulevia testausprosesseja sekä ohjelmistokehityksen elinkaarta.
2. Raportoinnin data luovutetaan projektin jälkeen johdolle kehitysehdotusten kanssa.

Testausprosessien johtaminen

Tarkistuspisteet kontrolloidulla tasolla:

1. Projektin alussa luodaan testaussuunnitelma, joka sisältää vähintään testustehtävät, testauksen laajuuden, testitapausten suunnittelun sekä roolit ja vastuut.
2. Testaussuunnitelma hyväksytetään pääasiallisella projektin omistajalla.
3. Jokainen testustehtävä suoritetaan kontrolloidusti, ja sitä muokataan tarvittaessa.
4. Testaussuunnitelma hyväksytetään kaikilla projektiin liittyvillä omistajilla.

Tarkistuspisteet tehokkaalla tasolla:

1. Mahdollisista muuttujista keskustellaan kaikkien projektiin liittyvien omistajien kanssa.
2. Testaussuunnitelmaan tehtävät muokkaukset dokumentoidaan.
3. Testauspäälliköllä on mandaatti allokoida testaajia ja muita resursseja vaadittaviin tehtäviin.

Tarkistuspisteet optimoidulla tasolla:

1. Testausprosessin johtamista arvioidaan jatkuvasti sisäisesti testausorganisaatiossa sekä omistajien kanssa.
2. Testausprosessin johtamista kehitetään jatkuvasti aiempien projektien oppien kautta.

Arviointi ja suunnittelu

Tarkistuspisteet kontrolloidulla tasolla:

1. Testauksen työmääräarviointiin käytetään vähintään yhtä yksinkertaista metodia.
2. Jokaiselle testaustehtävälle on suunniteltu suoritusajankohta, vaadittavat resurssit ja hyväksymiskriteerit. Testaustehtäviksi mielletään testauksensuunnittelu, -johtaminen, testitapausten määrittäminen ja niiden suorittaminen.
3. Testausvaiheiden ja -tehtävien riippuvuudet on huomioitu ja niiden osittaiset päällekkäisyydet on sallittu.
4. Arvioista ja suunnittelusta keskustellaan pääasiallisen omistajan kanssa.

Tarkistuspisteet tehokkaalla tasolla:

1. Työmääräarvioinneissa käytetään vähintään kahta eri metodia.
2. Testausvaiheiden kesto arvioidaan muodollisella tekniikalla.
3. Arvioinneissa hyödynnetään mittareita.
4. Testauksen suunnittelussa huomioidaan järjestelmän testattavuus.

Tarkistuspisteet optimoidulla tasolla:

1. Testauksen suunnittelussa huomioidaan testausmateriaalin mahdollinen uudelleenkäyttö.
2. Työmääräarvioinneissa käytetään organisaatiotasolla yhtenäisiä tekniikoita ja periaatteita.
3. Työmääräarvioinneissa käytetään organisaatiotasolla yhtenäisiä mittaristoja.

Mittaristot

Tarkistuspisteet kontrolloidulla tasolla:

1. Mittaristoja hyödynnetään projektin testauksen suunnittelussa ja hallinnassa.
2. Mittariston syötteet tallennetaan ja määritellyt tulokset dokumentoidaan säännöllisesti.
3. Mittariston data pystytään todistamaan oikeaksi.

Tarkistuspisteet tehokkaalla tasolla:

1. Datan keräämiseen ja analysointiin käytettyä työmäärää verrataan siitä saavutettuun hyötyyn.
2. Datan kerääminen ei vaaranna projektin etenemistä.
3. Testausprossin tehokkuutta mitataan.
4. Mittaristojen tuloksista keskustellaan omistajien kanssa.

Tarkistuspisteet optimoidulla tasolla:

1. Tapaa, jolla mittaristot tuottavat tietoa, monitoroidaan.
2. Muutokset informaationtarpeessa johtavat mittaristojen muokkaamiseen.

Vikojen hallinta

Tarkistuspisteet kontrolloidulla tasolla:

1. Vikojen elinkaari on hallittu ja sitä noudatetaan.
2. Vikaraportille kirjataan aina vähintään seuraavat asiat: tunnistenumero, liittyvä testitapaus, raporttija, aika, vakavuusaste, toistamistapa, odotetut ja varsinaiset tulokset sekä status.
3. Vika on osoitettu vastuuhenkilölle.
4. Kaikilla osallisilla on pääsy vikojenhallintaohjelmaan.

Tarkistuspisteet tehokkaalla tasolla:

1. Vikojenhallintaohjelmassa on selkeät siirtymät eri statusten välillä.
2. Kaikki vastuulliset henkilöt käyttävät samaa vikojenhallintaohjelmaa tai vastaavaa toimivalla integraatiolla.
3. Vikojenhallintaohjelmasta saa muodostettua mukautettuja raportteja.
4. Vikojenhallintaohjelma kykenee tunnistamaan trendejä hyödyntämällä esimerkiksi vian juurisyytä, prioriteetteja, ohjelmistoversioita ja testitapauksia.

Tarkistuspisteet optimoidulla tasolla:

1. Organisaatiossa käytetään yhtenäistä tapaa raportoida viat.
2. Vikojenhallinta on selkeästi roolitettu, ja testausprosessi tarjoaa siihen riittävästi tietoa.
3. Löydetyt viat analysoidaan, niistä etsitään yhtäläisyyksiä ja tästä saatua tietoa käytetään vikojen estämiseksi.

Testausmateriaalin hallinta

Tarkistuspisteet kontrolloidulla tasolla:

1. Testauksen lähdedokumentti, testauksen kohde ja testausmateriaalit ovat nimetty ja versioitu.
2. Jokainen testitapaus kohdistuu johonkin testauksen lähdedokumentin kohtaan.
3. Testaajilla on pääsy kaikkeen testausmateriaaliin.

4. Testaajilla on tiedossa, miten testauksen lähdedokumenttia, testauksen kohdetta ja testausmateriaalia käytetään.

Tarkistuspisteet tehokkaalla tasolla:

1. Testauksen lähdedokumenttiin, testauksen kohteeseen ja testausmateriaaleihin viitataan nimellä ja versiolla.
2. Vaatimusten ja testitapausten välillä on selvät suhteet.
3. Testausmateriaalin hallintaa tuetaan loogisella tallennustavalla, rooleilla ja vastuilla.

Tarkistuspisteet optimoidulla tasolla:

1. Projektin alussa mietitään, mitä testausmateriaalia on mahdollista uudelleenkäyttää tulevaisuudessa.
2. Ohjeet testausmateriaalin uudelleenkäyttöön ovat saatavilla ja uudelleenkäyttöä mitataan.
3. Uudelleenkäytettävä testausmateriaali on projektin päättyessä helposti eroteltavissa hävitettävästä testausmateriaalista.

Testausmetodit

Tarkistuspisteet kontrolloidulla tasolla:

1. Testausprosessi seuraa dokumentoituja testausmetodeita, jotka kuvaavat vaadittavat aktiviteetit.
2. Testausmetodit sopivat kehitystiimin toimintatapoihin.
3. Projektin testaajat kokevat testausmetodit projektiin sopiviksi.

Tarkistuspisteet tehokkaalla tasolla:

1. Dokumentoitu testausmetodi kuvaa tavoitteet, vastuuroolit, esivaatimukset ja kaikki tekniikat.

2. Testaajille annetaan ymmärrettävä ja kattava tehtäväsapluuna testauksen tueksi.
3. Testausmetodin toimenpiteet on jaoteltu pakollisiin, suositeltaviin ja vapaaehtoiisiin.
4. Pakolliset ja suositeltavat toimenpiteet suoritetaan.

Tarkistuspisteet optimoidulla tasolla:

1. Testaajat antavat jatkuvasti palautetta käytetyistä testausmetodeista.
2. Käytettyjä testausmetodeita kehitetään jatkuvasti.

Testaajien ammattitaito

Tarkistuspisteet kontrolloidulla tasolla:

1. Testaajat ovat saaneet ohjelmistotestaukseen liittyvää koulutusta ja/tai omaavat alaan liittyvää kokemusta.
2. Testaajat tuntevat käytettävät testausmetodit ja noudattavat niitä.
3. Testaajat saavat tarvittaessa tukea liiketoiminnan tai tekniikan asiantuntijoilta.
4. Testaajien ammattitaitoa sekä tietoteknisiä taitoja arvioidaan jatkuvasti.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaajilla on ISTQB- tai TMAP NEXT -sertifikaatti.
2. Testaajat pystyvät perustelemaan, miksi valittuja testausmetodeja käytetään.
3. Testaajat nauttivat työstään, ja heillä on hyvät välit projektin muiden osaamisalueryhmien kanssa.
4. Testaustehtävät määritellään, jaetaan ja suoritetaan odotusten mukaisesti.

Tarkistuspisteet optimoidulla tasolla:

1. Testaajat osallistuvat jatkuvasti alan seminaareihin tai lukevat alan kirjallisuutta pitääkseen ammattitaitonsa korkealla.
2. Testaajille on organisaation sisäinen uraputki.

3. Testaajat pyrkivät jatkuvaan kehittämiseen sekä kokevat vastuuta työstään.

Testitapausten suunnittelu

Tarkistuspisteet kontrolloidulla tasolla:

1. Testitapaukset dokumentoidaan ylätasolla.
2. Testitapaukset koostuvat alkutilanteesta, suoritettavista toiminnoista ja odotetuista tuloksista.
3. Testitapauksista selviää, mitä kohtaa testauksen lähdedokumentista tarkalleen testataan.

Tarkistuspisteet tehokkaalla tasolla:

1. Testitapaukset ovat helposti ymmärrettäviä, ja tarvittaessa muutkin kuin niiden kirjaaja pystyvät muokkaamaan niitä.
2. Testitapauksista pystytään johtamaan, kuinka suuri osa testauksen lähdedokumentista on katettu testitapauksilla.
3. Testitapauksien suunnitteluun käytetään muodollisia tekniikoita.
4. Laatuominaisuuksia, joita ei voida kattaa testitapauksilla, testataan muistilistojen avulla.

Tarkistuspisteet optimoidulla tasolla:

1. Tuotannossa löydetyt virheet analysoidaan, ja niiden avulla pyritään kehittämään testitapauksia sekä niiden kattavuutta.
2. Testitapaukset validoidaan yksittäin oikeellisuuden ja uudelleenkäytettävyyden varmistamiseksi.
3. Testitapausten suunnittelutekniikoita arvioidaan ja kehitetään jatkuvasti.

Testaustyökalut

Tarkistuspisteet kontrolloidulla tasolla:

1. Testaajilla on tarvittavat testaustyökalut testitapausten suorittamiseen.
2. Koko projektilla on tieto käytettävissä olevista testaustyökaluista.
3. Kaikki projektin jäsenet pitävät käytettäviä testaustyökaluja hyödyllisinä.

Tarkistuspisteet tehokkaalla tasolla:

1. Käytettävät testaustyökalut on valittu, jotta testaus olisi nopeampaa, edullisempaa ja jotta testausprosessi olisi paremmin hallittavissa.
2. Testaustyökalut ovat aina tarvittaessa käytettävissä.
3. Testaustyökalujen käyttö on taloudellisesti perusteltavissa.
4. Testaustyökalujen käyttö on integroitu testausprosessiin.

Tarkistuspisteet optimoidulla tasolla:

1. Työkalujenkäyttöpolitiikka on linjassa testauspolitiikan kanssa.
2. Opitut parhaat käytännöt ja asiantuntemus testaustyökaluista ovat aina tulevien projektien käytössä.
3. Testaustyökaluja arvioidaan jatkuvasti sen perusteella, miten niillä saavutetaan liiketoiminnan asettamat tavoitteet.

Testiympäristöt

Tarkistuspisteet kontrolloidulla tasolla:

1. Testiympäristöjen vaatimukset on dokumentoitu.
2. Ylläpitovastuut on sovittu testiympäristöjen toimittajien kanssa.
3. Testiympäristöt ovat käytettävissä sovittuina ajankohtina.
4. Testauspäällikölle ilmoitetaan ympäristöjen päivitysajankohdat.

Tarkistuspisteet tehokkaalla tasolla:

1. Testiympäristöt hyväksytään etukäteen suunnitellun muistilistan avulla.

2. Looginen tai toiminnallinen suunnitelma testiympäristön järjestelmäyhteyksistä on muodostettu. Suunnitelmassa huomioidaan järjestelmäintegraatiot muihin testiympäristöihin, ja tarvittaessa niitä mallinnetaan stubien avulla.
3. Testiympäristöjen toimittajat tuottavat teknisen järjestelmäsuunnitelman, jonka testauspäällikkö tai ympäristövastaava hyväksyy.
4. Testiympäristöjen laadusta on tehty palvelun tasoa koskeva sopimus.

Tarkistuspisteet optimoidulla tasolla:

1. Nimetty osasto hallitsee testiympäristöjä.
2. Testiympäristöjen käytöstä on sovittu toimittajien kanssa sopimuksella.
3. Sopimustasolla on eritelty, mitkä palvelut liittyvät testiympäristöön ja mitkä eivät.

Liite 2

Projektin sitoutuminen testaukseen

Tarkistuspisteet kontrolloidulla tasolla:

1. Projektin on resursoitu testauspäällikkö ja liiketoiminnan henkilöitä testauksen suorittamiseen.
2. Testaukseen sidotut resurssit ovat aidosti käytettävissä.
3. Projektin johto tiedostaa, että testaukseen tarvitaan aikaa ja resursseja.
4. Testaajin sidottu budjetti on määritetty, ja siitä voi tarvittaessa neuvotella palvelun omistajan kanssa.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaustehtävät, -laajuus ja lähestyminen sovitaan projektin alussa palvelunomistajien tai projektipäällikön kanssa.
2. Testaajat ovat mukana projektin suunnittelussa sekä riskianalyysissä. He ymmärtävät riippuvuudet muihin prosesseihin ja järjestelmiin.
3. Testaajat ovat mukana muutospyyntöjen riskianalyysissä ja ymmärtävät niiden vaikutukset testaustehtäviin sekä ottavat kantaa virheiden vakavuusasteisiin.

Tarkistuspisteet optimoidulla tasolla:

1. Omistajat tai projektipäällikkö myötävaikuttavat proaktiivisesti testausprosesseihin vaikuttaviin asioihin, kuten esimerkiksi testattavien ominaisuuksien toimittamiseen tai projektin vaiheen sisältöön.
2. Projekti on valmis muokkaamaan projektin työskentelytapoja – esimerkiksi vaatimustenmäärittelyä – testauksen edesauttamiseksi.
3. Testausorganisaatio on tärkeä osa kehitysakiviteetteja, sitä arvostetaan ja sen näkökulmia kunnioitetaan.

Testauksen suunnittelu

Tarkistuspisteet kontrolloidulla tasolla:

1. Testauspäällikkö muodostaa testaussuunnitelman, jonka projektiryhmä katselmoi ja hyväksyy.
2. Testaussuunnitelmassa otetaan huomioon testauksen tavoitteet ja riskianalyysi.
3. Suunnitelmassa huomioidaan testaustasot, -tavat, testauksen kattavuus ja hyväksyntätestauksen hyväksymiskriteerit.
4. Uudelleentestauksen ja regressiotestauksen toimintatavat on suunniteltu.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaussuunnitelmassa huomioidaan toimittajan tekemä systeemitestaus, integraatio- ja yksikkötestaus ja niiden hyväksymisen edellytykset.
2. Testauksen toimintatavat sopivat kehitystiimin toimintatapoihin.
3. Testaussuunnitelmassa huomioidaan tietoturva-, suorituskyky- sekä automaatiotestauksen tarve.
4. Testaustapojen ja testauksen kattavuuden erot eri testaustasoilla on tarkkaan huomioitu.

Tarkistuspisteet optimoidulla tasolla:

1. Testaussuunnitelman pääkohdat on huomioitu sopimustasolla.
2. Testaussuunnitelmaa arvioidaan tuotannon virhedatan perusteella.

Kommunikaatio

Tarkistuspisteet kontrolloidulla tasolla:

1. Jokainen testaukseen osallistuva projektin jäsen on tietoinen tehdyistä päätöksistä ja kehityksen vaiheista.
2. Testaajat hankkivat aktiivisesti tietoja projektin omistajilta.

3. Testaajien tekemät toimenpiteet ja päätökset dokumentoidaan.
4. Testaajat kommunikoivat omistajien tai projektipäällikön kanssa aktiivisesti kehityksestä, ohjelmiston laadusta ja sekä mahdollisista uhkista.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaajat tunnistavat, mitä tietoja ja havaintoja tulee kommunikoida omistajille tai projektipäällikölle.
2. Testaajat osallistuvat oleellisiin kokouksiin muun projektin kanssa.
3. Testaajilla on tarvittavat keinot kommunikoida omistajien tai projektipäällikön kanssa virallisia kaavakkeita käyttäen tai epävirallisesti.

Tarkistuspisteet optimoidulla tasolla:

1. Parhaita kommunikoinnin opittuja käytäntöjä arvioidaan ja kehitetään jatkuvasti sekä kuluvaan että tulevia projektia varten.
2. Projekti kokeilee uusia tapoja kommunikoida sekä tarvittaessa määrittää kommunikointikäytäntöjä.

Raportointi

Tarkistuspisteet kontrolloidulla tasolla:

1. Testauksen raportoinnissa käytetään yrityksen määrittämää työkalua.
2. Testauksen raportoinnissa huomioidaan aika, kustannukset, tulokset sekä riskit.
3. Testauksen raportoinnin tiheys on riittävä tukemaan omistajien päätöksentekoa.
4. Testauksen raportointi tehdään kirjallisesti tai se on löydettävissä sovitusta työkalusta.

Tarkistuspisteet tehokkaalla tasolla:

1. Raportoinnissa huomioidaan testausprosessin edistyminen ja projektin riskit ja tehdään suosituksia niihin perustuen.

2. Toimittajan tekemästä testauksesta saadaan raportti sovitussa muodossa.
3. Testausraporttien sisältöä pystytään muokkaamaan kesken projektin, jos se koetaan tarpeelliseksi.
4. Testauksen loppuraportti toimitetaan projektin johdolle projektin päättyttyä.

Tarkistuspisteet optimoidulla tasolla:

1. Raportoinnilla tuotetaan dataa, jonka avulla voidaan kehittää nykyistä sekä tulevia testausprosesseja sekä ohjelmistokehityksen elinkaarta.
2. Raportoinnin data luovutetaan projektin jälkeen projektin johdolle kehitysehdotusten kanssa.

Testausprosessien johtaminen

Tarkistuspisteet kontrolloidulla tasolla:

1. Projektin alussa luodaan testaussuunnitelma, joka sisältää vähintään testaustehtävät, testauksen laajuuden, testitapausten suunnittelun sekä roolit ja vastuut.
2. Testaustehtävät määritellään ja jaetaan projektiryhmän kesken.
3. Jokainen testaustehtävä suoritetaan kontrolloidusti, ja sitä muokataan tarvittaessa.
4. Testaukseen liittyvät toimenpiteet aloitetaan hyvissä ajoin ennen varsinaisen testauksen suorittamista.

Tarkistuspisteet tehokkaalla tasolla:

1. Mahdollisista muuttujista keskustellaan kaikkien projektiin liittyvien omistajien tai projektipäällikön kanssa.
2. Testauspäällikkö mukauttaa testaussuunnitelmaa ja suoritettavia toimenpiteitä suhteessa kehityksen edistykseen.
3. Testauspäälliköllä on mandaatti allokoida testaajia ja muita resursseja vaadittaviin tehtäviin.

Tarkistuspisteet optimoidulla tasolla:

1. Testausprosessin johtamista arvioidaan jatkuvasti projektin sisällä.
2. Testausprosessin johtamista kehitetään jatkuvasti kuluvan ja aiempien projektien oppien kautta.

Työmääräarviointi ja suunnittelu

Tarkistuspisteet kontrolloidulla tasolla:

1. Testauksen työmääräarviointiin käytetään vähintään yhtä yksinkertaista metodia.
2. Jokaiselle testaustehtävälle on suunniteltu suoritusajankohta, vaadittavat resurssit ja hyväksymiskriteerit. Testaustehtäviksi mielletään testauksensuunnittelu, -johtaminen, testitapausten määrittäminen ja niiden suorittaminen.
3. Arvioista ja suunnittelusta keskustellaan projektin johdon kanssa.
4. Edellisiä projekteja hyödynnetään arvioinnissa.

Tarkistuspisteet tehokkaalla tasolla:

1. Työmääräarvioinneissa käytetään vähintään kahta eri metodia.
2. Testausvaiheiden kesto arvioidaan muodollisella tekniikalla.
3. Mittareita hyödynnetään arvioinneissa.

Tarkistuspisteet optimoidulla tasolla:

1. Testauksen suunnittelussa huomioidaan testausmateriaalin mahdollinen uudelleenkäyttö.
2. Työmääräarvioiden toteutumia seurataan ja toimintatapoja kehitetään tulosten pohjalta.

Mittaristot

Tarkistuspisteet kontrolloidulla tasolla:

1. Mittaristoja hyödynnetään projektin testauksen suunnittelussa ja hallinnassa.
2. Hyväksyntätestauksessa havaittujen vikojen määrää mitataan.
3. Hyväksyntätestauksen onnistuneiden ja epäonnistuneiden testitapausten määrää mitataan.
4. Datan kerääminen ei vaaranna projektin etenemistä.

Tarkistuspisteet tehokkaalla tasolla:

1. Toimittajan systeemitestauksesta läpipäässeiden vikojen määrää mitataan.
2. Toimittajan systeemitestauksen onnistuneiden ja epäonnistuneiden testitapausten määrää mitataan.
3. Mittariston syötteet tallennetaan ja määritellyt tulokset dokumentoidaan säännöllisesti.
4. Testausprosessin tehokkuutta mitataan säännöllisin väliajoin.

Tarkistuspisteet optimoidulla tasolla:

1. Muutokset informaationtarpeessa johtavat mittaristojen muokkaamiseen.
2. Datan keräämiseen ja analysointiin käytettyä työmäärää verrataan siitä saavutettuun hyötyyn.
3. Tapaa, jolla mittaristot tuottavat tietoa, monitoroidaan.

Vikojen hallinta

Tarkistuspisteet kontrolloidulla tasolla:

1. Vikojen elinkaari on hallittu ja sitä noudatetaan.
2. Vikaraportille kirjataan aina vähintään seuraavat asiat: tunnistenumero, liittyvä testitapaus, raportoiija, havaintoaika, testausympäristö vakavuusaste, toistamistapa, odotetut ja varsinaiset tulokset sekä status.
3. Vika on osoitettu vastuuhenkilölle.
4. Kaikilla projektin jäsenillä on pääsy vikojenhallintaohjelmaan.

Tarkistuspisteet tehokkaalla tasolla:

1. Vikojenhallintaohjelmassa on selkeät siirtymät eri statusten välillä.
2. Kaikki vastuulliset henkilöt käyttävät samaa vikojenhallintaohjelmaa tai vastaavaa toimivalla integraatiolla.
3. Vikojenhallintaohjelmasta saa muodostettua mukautettuja raportteja.

Tarkistuspisteet optimoidulla tasolla:

1. Vikojenhallintaohjelma kykenee tunnistamaan trendejä hyödyntämällä esimerkiksi vian juurisyytä, prioriteetteja, ohjelmistoversioita ja testitapauksia.
2. Vikojenhallinta on selkeästi roolitettu, ja testausprosessi tarjoaa siihen riittävästi tietoa.
3. Löydetyt viat analysoidaan, niistä etsitään yhtäläisyyksiä ja tästä saatua tietoa käytetään vikojen estämiseksi.

Testaajien ammattitaito

Tarkistuspisteet kontrolloidulla tasolla:

1. Testaajat ovat saaneet ohjelmistotestaukseen liittyvää koulutusta ja/tai omaavat alaan liittyvää kokemusta.
2. Ammattitestaajat saavat tarvittaessa tukea liiketoiminnan tai tekniikan asiantuntijoilta.
3. Liiketoiminnan testaajat saavat tarvittaessa tukea testauksen ammattilaisilta.

Tarkistuspisteet tehokkaalla tasolla:

1. Testaajilla on ISTQB-sertifikaatti.
2. Testaajat pystyvät perustelemaan, miksi valittuja testausmetodeja käytetään.
3. Testaajat nauttivat työstään, ja heillä on hyvät välit projektin muiden osaamialueryhmien kanssa.

Tarkistuspisteet optimoidulla tasolla:

1. Testaajat osallistuvat jatkuvasti alan seminaareihin tai lukevat alan kirjallisuutta pitääkseen ammattitaitonsa korkealla.
2. Testaajille on organisaation sisäinen uraputki.
3. Testaajat pyrkivät jatkuvaan kehittymiseen sekä kokevat vastuuta työstään.

Testitapausten suunnittelu

Tarkistuspisteet kontrolloidulla tasolla:

1. Testitapaukset dokumentoidaan ylätasolla.
2. Testitapaukset koostuvat alkutilanteesta, suoritettavista toiminnoista ja odotetuista tuloksista.
3. Testitapausten suunnittelussa hyödynnetään vaatimuksia tai testitapauksia käytetään vaatimuksina.
4. Tarvittaessa päätelaite- ja selainkattavuus on huomioitu testitapauksissa.

Tarkistuspisteet tehokkaalla tasolla:

1. Testitapaukset ovat helposti ymmärrettäviä, ja tarvittaessa muutkin kuin niiden kirjaaja pystyvät muokkaamaan niitä.
2. Testitapauksista pystytään johtamaan, kuinka suuri osa testauksen lähdedokumentista on katettu testitapauksilla.
3. Testitapauksien suunnitteluun käytetään muodollisia tekniikoita.
4. Laatuominaisuuksia, joita ei voida kattaa testitapauksilla, testataan muistilistojen avulla.

Tarkistuspisteet optimoidulla tasolla:

1. Tuotannossa löydetty virheet analysoidaan, ja niiden avulla pyritään kehittämään testitapauksia sekä niiden kattavuutta.
2. Testitapaukset validoidaan yksittäin oikeellisuuden ja uudelleenkäytettävyyden varmistamiseksi.
3. Testitapausten suunnittelutekniikoita arvioidaan ja kehitetään jatkuvasti.

Testiympäristöt

Tarkistuspisteet kontrolloidulla tasolla:

1. Testiympäristöjen vaatimukset on dokumentoitu.
2. Testiympäristöt ovat käytettävissä sovittuina ajankohtina.
3. Testiympäristön testidataa pystytään tarvittaessa luomaan tai muokkaamaan, ja tehtävälle on määritelty vastuuhenkilö.
4. Testauspäällikkö hyväksyy ympäristöjen päivitysajankohdat.

Tarkistuspisteet tehokkaalla tasolla:

1. Testiympäristöt hyväksytään etukäteen suunnitellun muistilistan avulla.
2. Ylläpitovastuut on sovittu testiympäristöjen toimittajien kanssa.
3. Looginen tai toiminnallinen suunnitelma testiympäristön järjestelmäyhteyksistä on muodostettu. Suunnitelmassa huomioidaan järjestelmäintegraatiot muihin testiympäristöihin, ja tarvittaessa niitä mallinnetaan stubien avulla.
4. Testiympäristöjen laadusta on tehty palvelun tasoa koskeva sopimus.

Tarkistuspisteet optimoidulla tasolla:

1. Testiympäristöjen toimittajat tuottavat teknisen järjestelmäsuunnitelman, jonka testauspäällikkö tai ympäristövastaava hyväksyy.
2. Testiympäristöjen käytöstä on sovittu toimittajien kanssa sopimuksella.
3. Sopimustasolla on eritelty, mitkä palvelut liittyvät testiympäristöön ja mitkä eivät.

Tietoturvatestaus

Tarkistuspisteet kontrolloidulla tasolla:

1. Yrityksen tietoturvasuunnittelu ja tietoturvavaatimukset on otettu huomioon projektissa.
2. Tietoturvaseurantaa tehdään projektin toteutusvaiheessa.
3. Ohjelmistoon tehdään tai on tehty tietoturvatestausta osana hyväksyntätestausta yrityksen tai kolmannen osapuolen toimesta.
4. Toimittaja-auditoinnin tarve on selvitetty ja tarvittaessa toteutettu.

Tarkistuspisteet tehokkaalla tasolla:

1. Havaittujen tietoturvariskien elinkaari on hallittu, ja sitä noudatetaan.
2. Yrityksen määrittämä ohjelmiston tietoturvavaatimusten tarkistuslista on otettu huomioon projektissa.
3. Projektin tietoturvariskit on kartoitettu, ja niiden varalle on suunnitelma.
4. Yrityksen suosimia parhaita käytäntöjä noudatetaan kehitystyössä.

Tarkistuspisteet optimoidulla tasolla:

1. Ohjelmistoon tehdään tai on tehty tietoturvatestausta osana kehitystyötä esim. yksikötasolla.
2. Uusintatestaus tehdään havaittujen vikojen korjauksen jälkeen.
3. Tietoturvaseurantaa tehdään jatkokehityksen yhteydessä.

Testausautomaatio

Tarkistuspisteet kontrolloidulla tasolla:

1. Testiautomaation laajuus ja sisältö on sovittu yhdessä projektin tai järjestelmän kehittäjien kanssa.
2. Käytetään keskitettyä versionhallintaa testiskriptien ylläpitoon.
3. Testitapaukset päivitetään ajan tasalle järjestelmään havaittujen virheiden jälkeen.

4. Kehittäjät tai testaajat ylläpitävät ja ajavat automatisoidut testisetit käynnistämällä ne käsin.

Tarkistuspisteet tehokkaalla tasolla:

1. Käytetään CI-järjestelmää automatisoitujen testisettien käynnistämiseen aikataulutetusti.
2. Testitapaukset päivitetään järjestelmään tulleiden muutosten jälkeen.
3. Tiedotus virheeseen päätyvistä testeistä automaattisesti asianomaisille sähköpostilla.
4. Statuspalavereita säännöllisesti testiautomaation tilanteesta.

Tarkistuspisteet optimoidulla tasolla:

1. Automatisoidut testisetit on määritelty CI-järjestelmässä käynnistymään automaattisesti ohjelmiston uusimman version asennuksen valmistuttua.
2. Testitapaukset päivitetään ennalta valmiiksi heti, kun järjestelmään tulevat muutokset ovat tiedossa.
3. Testiajajen kaikki virheet analysoidaan, jotta opitaan optimoimaan testitapaukset ja edesauttamaan automatisointia jopa muokkaamalla ohjelmiston koodia.

Suorituskykytestaus

Tarkistuspisteet kontrolloidulla tasolla:

1. Suorituskykytestauksen tarpeen määrittäminen tehdään osana projektin suunnittelua ja tarvittaessa ylläpitovaiheessa.
2. Käyttötapaukset määritellään ja mallinnetaan työkalulla.
3. Käytettävä testiympäristö ja suorituskykytestaustyökalu on määritelty.
4. Suorituskykytestaus suoritetaan määrittysten mukaisesti halutussa kohdeympäristössä.

Tarkistuspisteet tehokkaalla tasolla:

1. Suorituskykytestauksesta muodostetaan raportti, joka sisältää mahdolliset vikahavainnot ja korjausehdotukset.
2. Mahdolliset viat korjataan ja uudelleentestataan.
3. Käyttötapauksiin määritetään käyttäjämäärät ja halutut vasteajat.
4. Kartoitetaan mahdollinen tarve ulkoisten resurssien käyttämiselle.

Tarkistuspisteet optimoidulla tasolla:

1. Käytettyä työkalua sekä testauksen suunnittelua arvioidaan suorituskykytestauksen jälkeen.
2. Suorituskykytestaus ei kohdistu ainoastaan yhteen järjestelmään, vaan ottaa aidosti huomioon kaikki integroidut järjestelmät kokonaisuutena.