

Lappeenrannan teknillinen yliopisto
Teknistaloudellinen tiedekunta
Tietotekniikan koulutusohjelma

Kandidaatintyö

Esa Kuparinen

**SUUNNITELMA OHJELMISTOTESTAUKSEN ADAPTIIVISEN
REFERENSSIMALLIN ARVIOINTIIN**

Työn tarkastaja: Tekniikan tohtori Ossi Taipale

Työn ohjaaja: Tekniikan tohtori Ossi Taipale

Lappeenranta, 23.8.2010

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto

Teknistaloudellinen tiedekunta

Tietotekniikan koulutusohjelma

Esa Kuparinen

Suunnitelma ohjelmistotestauksen adaptiivisen referenssimallin arviointiin

Kandidaatintyö

2010

30 sivua, 5 kuvaa, 2 taulukkoa

Työn tarkastaja: Tekniikan tohtori Ossi Taipale

Hakusanat: ohjelmistotestaus, referenssimalli, arviointi, ISO/IEC 29119, ISO/IEC 15504

Keywords: software testing, reference model, assessment, ISO/IEC 29119, ISO/IEC 15504

Tämän kandidaatintyön tavoitteena on laatia suunnitelma ISO/IEC 29119 testausstandardiin perustuvan ohjelmistotestauksen adaptiivisen referenssimallin arviointia varten. Arviointia ei kuitenkaan suoriteta tämän työn puitteissa, vaan myöhemmin MASTO-projektissa. Työssä esitellään ISO/IEC 29119 ohjelmistotestausstandardi ja ISO/IEC 15504 prosessinarviointistandardi sekä perehdytään yleisesti ohjelmistotestauskäytännöissä esiintyviin ongelmiin kirjallisuuskatsauksen keinoin. Näiden taustatietojen pohjalta tehdään arviointisuunnitelma, joka arvioi referenssimallia kahdelta kannalta. Ensiksi arvioidaan sen määrittämän testausprosessin kyvykkyyttä ja toiseksi sen yleistä hyödyllisyyttä ja käyttökelpoisuutta.

ABSTRACT

Lappeenranta University of Technology
Faculty of Technology Management
Degree Program in Information Technology

Esa Kuparinen

Plan for the Assessment of the Adaptive Reference Model of Software Testing

Bachelor's Thesis

2010

30 pages, 5 figures, 2 tables

Examiner: D.Sc. Ossi Taipale

Keywords: software testing, reference model, assessment, ISO/IEC 29119, ISO/IEC 15504

The aim of this bachelor's thesis is to make a plan for the assessment of the adaptive reference model of software testing, which is based on the ISO/IEC 29119 testing standard. The assessment is not carried out in the scope of this thesis, but later in the related MASTO project. The ISO/IEC 29119 testing standard and the ISO/IEC 15504 process assessment standard are explained and a literary review is done about problems in software testing practices. Based on this background information a plan for the assessment is made. The assessment plan assesses the reference model from two perspectives. First the capability of the testing process described by the reference model is assessed and then the practical usefulness of the model is assessed.

ALKUSANAT

Tämä kandidaatintyö on tehty Lappeenrannan teknillisen yliopiston TBRC-yksikön MASTO-projektiin liittyen. Kiitän kaikkia niitä, jotka ovat tuellansa edesauttaneet tämän työn valmistumista. Erityisesti haluan kiittää työni ohjaajana toiminutta Ossi Taipalettä sekä MASTO-projektissa työskennellyttä Leah Riungua heidän ohjauksestaan ja kehitysideoistaan työhöni liittyen.

Lappeenrannassa 23.8.2010

Esa Kuparinen

SISÄLLYSLUETTELO

1	JOHDANTO	3
1.1	TAUSTA	3
1.2	TAVOITTEET JA RAJAUKSET	3
1.3	TYÖN RAKENNE	4
2	TUTKIMUSALUEEN ESITTELY	5
2.1	ISO/IEC 29119 OHJELMISTOTESTAUSSTANDARDI	5
2.1.1	<i>Osa 1: Käsitteet ja sanasto</i>	6
2.1.2	<i>Osa 2: Testausprosessi</i>	7
2.1.3	<i>Osa 3: Testausdokumentaatio</i>	13
2.1.4	<i>Osa 4: Testaustekniikat</i>	14
2.2	ISO/IEC 15504 PROSESSINARVIOINTISTANDARDI	15
2.3	ONGELMAT OHJELMISTOTESTAUSKÄYTÄNNÖISSÄ	16
2.4	MASTO-PROJEKTIN 4. KIERROKSEN HAASTATTELUT	18
3	ARVIOINTISUUNNITELMAN ESITTELY	20
3.1	TESTAUSPROSESSIN KYVYKKYYDEN ARVIOINTI.....	20
3.2	KÄYTTÖKELPOISUUDEN JA HYÖDYLLISYYDEN ARVIOINTI	21
4	POHDINTA JA TULEVAISUUS	24
5	YHTEENVETO	25
	LÄHTEET	26

SYMBOLI- JA LYHENNELUETTELO

IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MASTO	Adaptive Reference Model for Software Testing Organizations
SPICE	Software Process Improvement and Capability Determination
TBRC	Technology Business Research Center

1 JOHDANTO

1.1 Tausta

Tämä kandidaatintyö liittyy MASTO-projektin (Adaptive Reference Model for Software Testing Organizations) syksyn 2010 työpakettiin. Projektista vastaa Lappeenrannan teknillisen yliopiston TBRC-yksikkö (Technology Business Research Center). MASTO-projektissa kehitetään muun muassa adaptiivista referenssimallia ohjelmistotestauksen prosessien ohjaamiseen ja kehittämiseen organisaatioissa. Mallin päämääränä on samanaikaisesti sekä parantaa ohjelmistojen laatua että alentaa testauskustannuksia. Ohjelmistojen laatuun vaikuttavia tekijöitä ovat ainakin ohjelmistotestausprosessien tila organisaatioissa ja laadunvarmistuksen toimivuus. Aiempien tutkimustulosten perusteella ohjelmistotestaus voi olla jopa kallein yksittäinen osa ohjelmistokehitysprojektissa kuluttaen arviolta noin 30 % kaikista resursseista. MASTO-projektissa käytetään tiedonhankintaan yhteistyöyrityksille tehtyjä kyselyjä ja haastatteluja, joiden tuloksia analysoimalla kartoitetaan yritysten ohjelmistotestausprosessien yleistä tilaa. [1, 2]

MASTO-hankkeen syksyn 2010 työpaketissa on tarkoitus arvioida hankkeen aikana kehitettyä ohjelmistotestauksen adaptiivista referenssimallia ja tässä kandidaatintyössä tullaan laatimaan suunnitelma tuon arvioinnin toteuttamistavasta. Edellä mainittu ohjelmistotestauksen adaptiivinen referenssimalli pohjautuu kehitteillä olevaan ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) 29119 ohjelmistotestausstandardiin. Kyseinen standardi määrittelee ohjelmistotestaukselle yleisen prosessimallin, jota voi soveltaa kaikenlaisiin ohjelmistokehitysprojekteihin riippumatta ohjelmiston elinkaarimallista tai kehitykseen käytettävästä vaihejakomallista. Tätä ISO/IEC 29119 testausstandardin määrittelemää prosessimallia kutsutaan ohjelmistotestauksen adaptiiviseksi referenssimalliksi. [2, 3]

1.2 Tavoitteet ja rajaukset

Työssä esitellään ohjelmistotestausstandardi ISO/IEC 29119 ja siihen perustuva ohjelmistotestauksen adaptiivinen referenssimalli, jota MASTO-projektissa on kehitetty. Lisäksi esitellään standardiin ISO/IEC 15504 liittyvä prosessinarviointimalli, jota on

lopulta tarkoitus käyttää ohjelmistotestauksen adaptiivisen referenssimallin arvioimiseen ainakin osittain. Työssä perehdytään myös tutkimuksiin ohjelmistotestauksen käytännöistä ja ongelmista, sekä tutustutaan MASTO-projektin aikana suoritettujen haastattelujen neljännen kierroksen vastauksiin. Näiden standardien, mallien, tutkimusten ja haastattelujen pohjalta laaditaan suunnitelma siitä miten adaptiivista referenssimallia voitaisiin arvioida.

On tärkeää huomioida rajaus, että tässä työssä ei tulla suorittamaan kyseistä arviointia. Tämän työn puitteissa vain laaditaan suunnitelma arvioinnin toteuttamista varten. Myöskään ohjelmistotestauksen alaa ja siihen liittyviä käsitteitä ei tässä työssä juurikaan ryhdytä esittelemään johtuen puhtaasti työn hyvin rajallisesta laajuudesta.

1.3 Työn rakenne

Tämän kandidaatintyön luvussa 2 kerrotaan taustatietoa tutkimusalueesta. Tässä luvussa esitellään aiheeseen keskeisesti liittyvät standardit ISO/IEC 29119 ja ISO/IEC 15504 hyvin yleisesti ja niiltä osin kuin ne työn aihepiiriin sekä tavoitteisiin soveltuvat. Lisäksi luvussa käsitellään muun muassa ohjelmistotestauskäytännöissä yleisesti havaittuja ongelmia ja MASTO-projektin aikana suoritettujen haastattelujen antia. Luvussa 3 esitetään suunnitelma ohjelmistotestauksen adaptiivisen referenssimallin arvioimiseksi. Luvussa 4 pohditaan työn tuloksia ja onnistuneisuutta sekä arvioidaan hieman tulevaisuutta. Luvussa 5 tehdään yhteenveto koko aiheesta.

2 TUTKIMUSALUEEN ESITTELY

Tämän työn tutkimusongelmana on laatia suunnitelma, jonka avulla ohjelmistotestauksen adaptiivista referenssimallia voidaan arvioida MASTO-projektin resurssien puitteissa. Tutkimusmenetelmänä käytetään kirjallisuuskatsausta.

ISO/IEC 29119 testausstandardi on käytännössä ohjelmistotestauksen adaptiivisen referenssimallin määritelmä, joten se on erittäin keskeinen osa tutkimusaluetta ja saa siksi suurimman huomion. ISO/IEC 15504 arviointistandardi on syytä esitellä, koska sitä suunnitellaan hyödynnettävän referenssimallin arvioimisessa. Lisäksi myös tietoja ohjelmistotestauskäytännöissä yleisesti esiintyvistä ongelmista ja MASTO-projektin 4. kierroksen haastatteluja käytetään hyväksi arvioinnin suunnittelemisessa.

2.1 ISO/IEC 29119 ohjelmistotestausstandardi

ISO/IEC 29119 ohjelmistotestausstandardi on vasta kehitysvaiheessa. Valmistuessaan kyseinen standardi tulee sisältämään neljä osaa: käsitteet ja sanasto, testausprosessi, testausdokumentaatio sekä testaustekniikat. Kolme ensimmäistä osaa näistä on tätä kirjoitettaessa julkaistu draft-versioina.

Standardin tarkoituksena on määrittää ohjelmistotestaukselle yleinen prosessimalli, jota mikä tahansa organisaatio voi käyttää suorittaessaan minkälaista ohjelmistotestausta tahansa. Valmistuessaan neliosaisen standardin on tarkoitus tarjota testaajille käyttökelpoinen kehys, jonka avulla testausta voidaan hallita ja suorittaa. Lyhyesti sanottuna ISO/IEC 29119 testausstandardi määrittelee muun muassa ohjelmistotestaukseen liittyvää sanastoa ja käsitteitä sekä yleisen testausprosessimallin, jota voi soveltaa ohjelmistonkehitysprojekteihin. Malli kuvaa testausprosesseja kolmella eri organisaatioyksikön tasolla ja määrittää minkälaisia toimintoja eri prosesseihin kuuluu sekä minkälaista dokumentaatiota niistä tulisi tuottaa. Tulevissa versioissa standardi tulee määrittelemään myös hyödyllisiä käytännön testaustekniikoita. Seuraavissa aliluvuissa käydään hieman tarkemmin läpi mitä kukin standardin osa pitää sisällään. [3, 4]

2.1.1 Osa 1: Käsitteet ja sanasto

Standardin ensimmäinen osa esittelee runsaasti ohjelmistotestaukseen liittyvää sanastoa selityksineen sekä kuvailee keskeisiä testaukseen liittyviä käsitteitä, ja tarjoaa siten pohjan, jonka päälle koko standardi rakentuu. Sanasto-osio on yksinkertaisesti vain aakkosjärjestyksellinen listaus testaussanastosta selityksineen, mutta lähes sadan sivun pituudestaan johtuen myös varsin kattava sellainen. Käsitteiden selvittäminen aloitetaan tarjoamalla yksi määritelmä ohjelmistotestaukselle ja perehtymällä sitten hieman testauksen yleisiin syihin ja päämääriin. Testausta kuvaillaan prosessina ja samalla pohditaan muun muassa sen erilaisia ominaisuuksia ja lajeja sekä testauksen roolia organisaatio- ja projektitasoilla. Standardin toisen osan tarkemmin määrittelemästä testauksen prosessimallista annetaan myös lyhyt kuvaus jo tässä ensimmäisessä osassa. [4]

Lisäksi standardin ensimmäisessä osassa kuvaillaan yleinen ohjelmiston elinkaarimalli ja käsitellään testausta yhtenä sen osana toimivana tukitoimena. Riskipohjaisen testaamisen käsite esitellään, koska ISO/IEC 29119 testausstandardi pohjautuu ajatukseen, että jokaisessa vaiheessa suoritetaan paras mahdollinen testi tunnistamalla eri näytteiden suhteellinen tärkeys perustuen niiden aiheuttamaan riskiin. Testaus suunnitellaan siis siten, että korkein prioriteetti annetaan aina suurimman riskin omaavalle kohteelle ja huomio kohdistetaan siihen. Tämän jälkeen standardissa kerrotaan lyhyesti mitä testausvaiheet ja testaustyytit ovat. Niiden yleisesti sisältämät piirteet on erikseen määritelty hieman tarkemmin. Yhteisiä piirteitä molemmille ovat muun muassa testin päämäärä, testin kohde sekä suoritettavat testiprosessit ja käytettävät testitekniikat. Erilaisista testausvaiheista (esim. yksikkö- tai järjestelmätestaus) ja testaustyypeistä (esim. suorituskykytestaus) annetaan myös yksityiskohtaisia esimerkkejä. [4]

ISO/IEC 29119-1 (vaihtoehtoinen merkintätapa, joka tarkoittaa samaa kuin ISO/IEC 29119, osa 1) antaa myös ohjeistuksen siitä, kuinka koko standardin määrittelemää testausmallia voidaan soveltaa erilaisiin ohjelmistokehityksen elinkaarimalleihin. Esimerkkeinä näistä malleista tarjotaan ketterä menetelmä (agile), evoluutiomalli (evolutionary) sekä perinteinen peräkkäismalli (sequential). Tämän jälkeen ohjeistetaan kuinka testausmallia voi soveltaa mainittuihin elinkaarimalleihin pohjautuviin

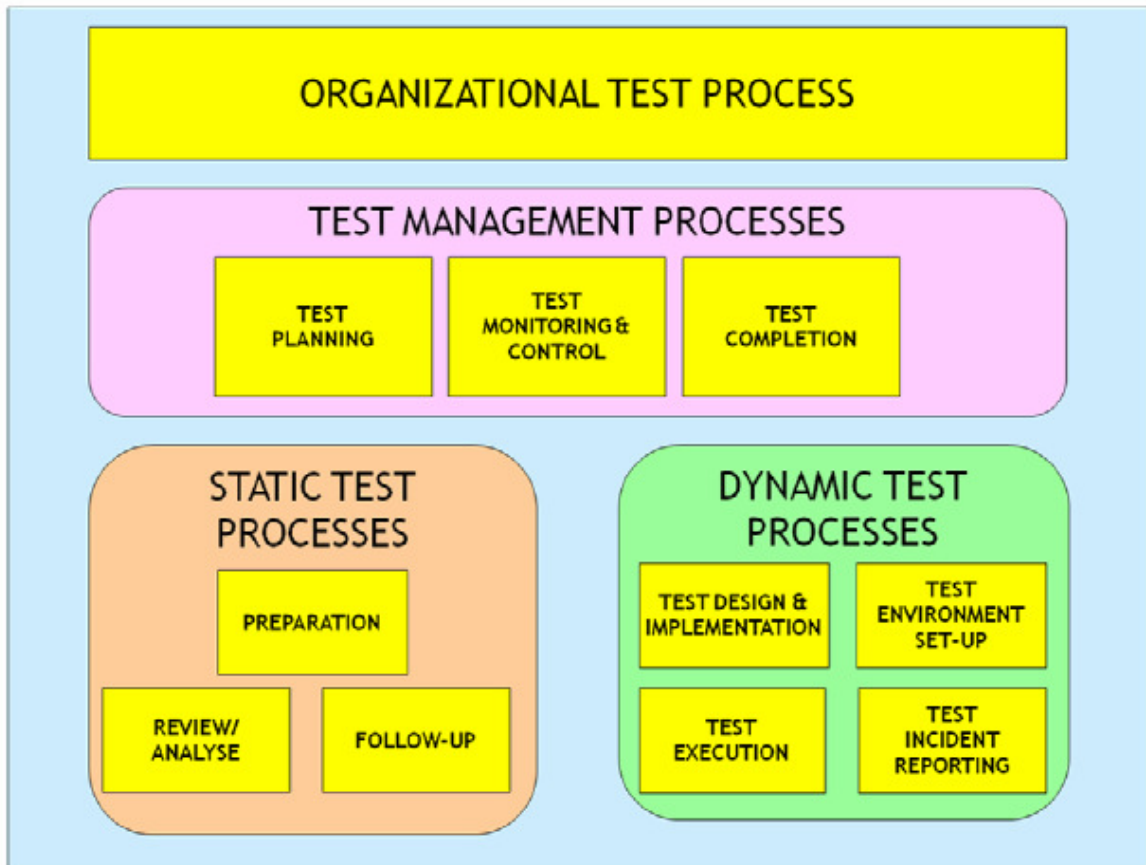
ohjelmistonkehitysprojekteihin. [4]

Standardin ensimmäisen osan lopussa käydään vielä läpi ohjelmistotestaukseen liittyviä erilaisia rooleja ja vastuualueita, sekä käsitellään testaajien ryhmädynamiikkaa tarjoamalla suuntaviivoja esimerkiksi testaajien väliselle kommunikaatiolle ja testausryhmien muodostamiselle. Myös testauksen mittaamiseen tehdään lyhyt katsaus. Siinä käydään läpi perusteita testauksen mittaamisesta, mittauksen suunnittelusta ja mittaustulosten esittämisestä. [4]

2.1.2 Osa 2: Testausprosessi

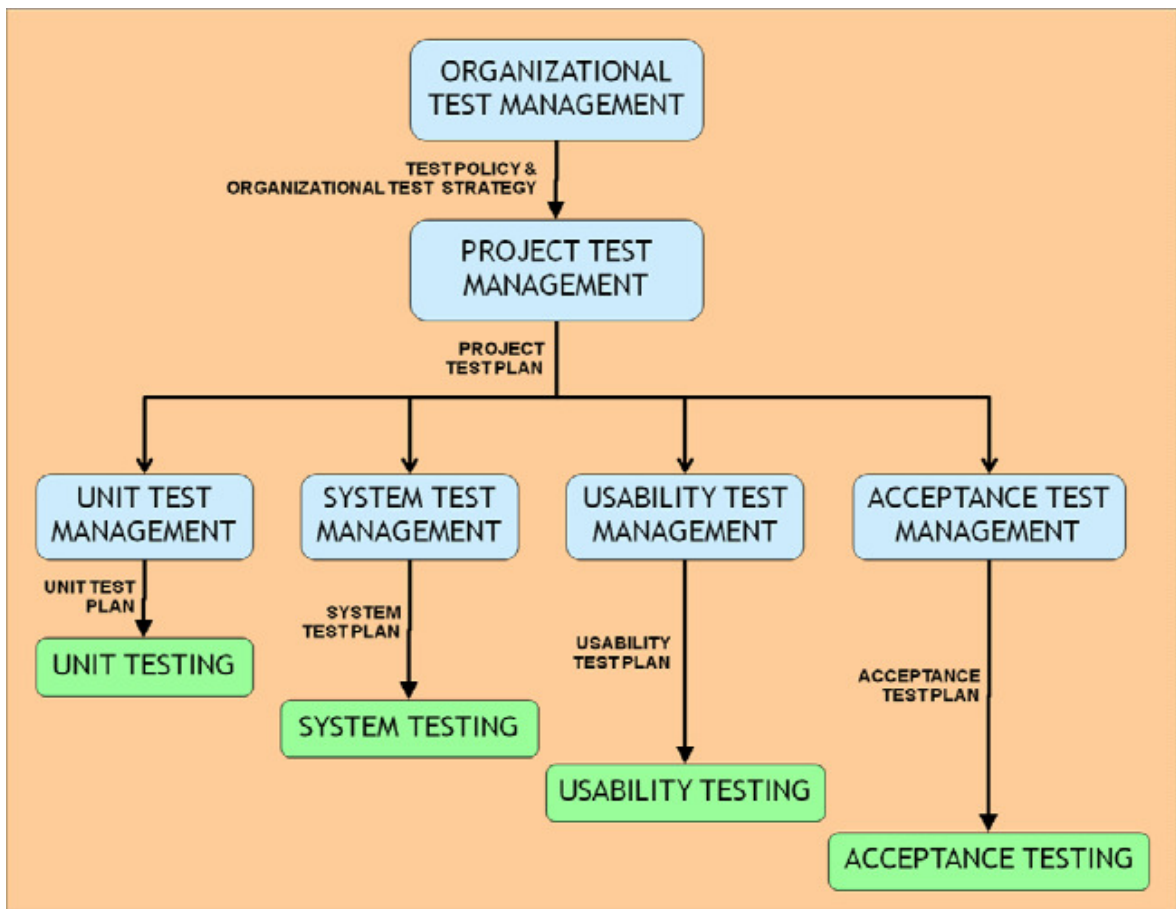
ISO/IEC 29119 standardin toinen osa määrittelee ohjelmistotestaukselle yleisen prosessimallin, joka koostuu kolmesta eri prosessitasosta. Ylimmän tason eli organisaatiotason prosessit ohjaavat yleisesti koko organisaatiossa tehtävää testaustyötä. Keskimmäisen eli testauksenhallintatason prosessit hallitsevat testausta projektitasolla tai yksittäisessä testausvaiheessa tai testaustyyppissä jonkin testausprojektin sisällä. Alimman tason eli testaustason prosesseissa suoritetaan varsinaiset käytännön testausaktiviteetit. Testaustason prosessit jaetaan kahteen ryhmään: staattisiin ja dynaamisiin testausprosesseihin. Kuvassa 1 on havainnollistettu testauksen prosessimalli. Kuvassa näkyvät kolmen päätason lisäksi myös näiden sisältämät prosessit, joista kerrotaan lisää seuraavaksi. [3]

Organisaatiotason testausprosessi kehittää, hallitsee ja ylläpitää koko organisaation laajuudelta sovellettavia testauspesifikaatioita. Ne eivät siis ole projektikohtaisia, vaan ovat käytössä koko organisaatiossa. Esimerkkejä tällaisista spesifikaatioista ovat organisaation testauspolitiikka (test policy) ja organisaation testausstrategia (test strategy) (näistä ja muista tässä luvussa myöhemmin mainittavista dokumenteista kerrotaan tarkemmin luvussa 2.1.3). Organisaatiotason testiprosessin tehtävänä on myös valvoa, että kyseisten dokumenttien määräyksiä noudatetaan organisaatiossa. [3]



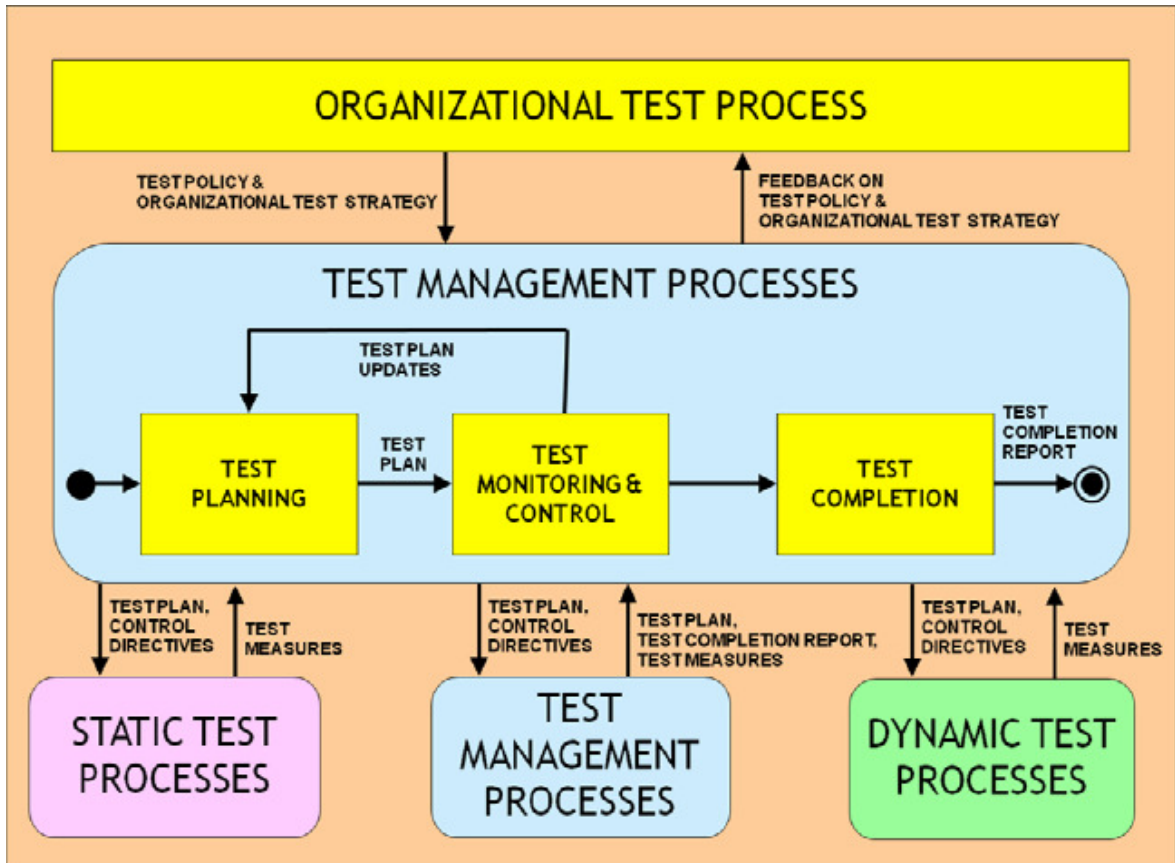
Kuva 1. Ohjelmistotestauksen prosessimalli. [3]

Testauksenhallintatason prosesseja voidaan siis soveltaa hallitsemaan testausta sekä projektitasolla että erilaisten testausvaiheiden ja testautyyppien suorittamisessa. Projektitasolla testausta hallitaan projektin testaussuunnitelman (test plan) mukaisesti. Testausvaiheiden ja testautyyppien tapauksessa taas käytetään hyväksi niitä vastaavia testaussuunnitelmia (esim. yksikkötestaussuunnitelmaa yksikkötestauksen kohdalla). Testauksenhallintaprosesseilla on se erikoisuus, että niitä voidaan soveltaa myös toisiin testauksenhallintaprosesseihin. Tätä havainnollistaa hyvin kuva 2, jossa on esitetty esimerkki projektin hallintarakenteesta. Kuvasta voidaan nähdä, että projektitason testauksenhallintaprosessi hallitsee suoraan sen alapuolella olevia hallintaprosesseja, kuten esimerkiksi yksikkötestauksenhallintaprosessia. Tällaisessa tilanteessa projektitason testauksenhallintaprosessi luo testaussuunnitelman, jota sitten alapuolella oleva yksikkötestauksenhallintaprosessi noudattaa luodessaan testaussuunnitelman yksikkötestausta varten. [3]



Kuva 2. Esimerkki projektin hallintarakenteesta. [3]

Kuvassa 3 on esitetty testauksenhallintatason prosessien keskinäiset suhteet sekä niiden vuorovaikutussuhteet muihin mallin tasoihin. ISO/IEC 29119-2:n määrittelemän ohjelmistotestauksen prosessimallin rakenne on periaatteessa hierarkkinen sikäli, että mallin ylemmät tasot määrittävät ja ohjaavat alempien toimintaa tuottamiensa dokumenttien kautta. Esimerkiksi organisaatiotason dokumentit (kuten testauspolitiikka ja testausstrategia) ohjaavat testauksenhallintatason prosesseja ja tämän tason tuottamat testaussuunnitelmat ohjaavat jälleen alempien tasojen prosessien toimintaa. Kuitenkin standardin määrittelemässä mallissa kommunikaatio toimii myös ylöspäin, kuten kuvasta 3 voi havaita. Esimerkiksi testauksenhallintatason prosessit voivat antaa organisaatiotasolle palautetta testauspolitiikan ja testausstrategian toimivuudesta. Tämä mahdollistaa koko organisaation testausprosessin jatkuvan kehittämisen. [3]

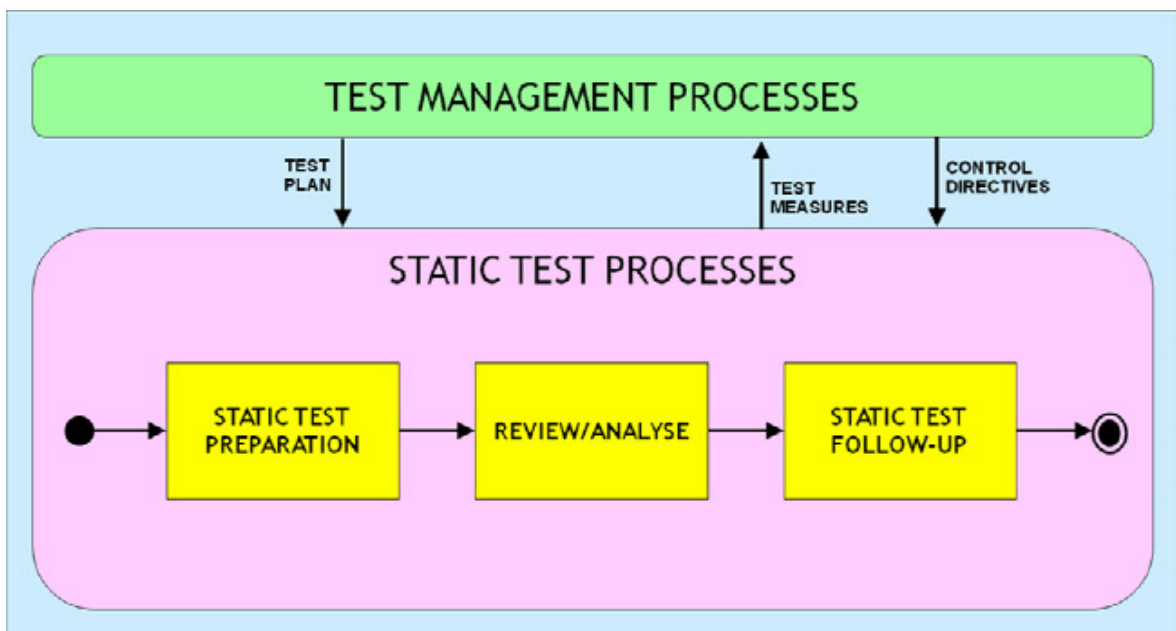


Kuva 3. Testauksenhallintatason prosessien suhteet toisiinsa ja muihin mallin tasoihin. [3]

Ensimmäinen testauksenhallintatason prosesseista on testauksen suunnittelu. Tämä prosessi tuottaa testaussuunnitelmadokumentin, jonka tyyppi riippuu siis siitä, että missä tämä hallintaprosessi sijaitsee projektissa. Toinen testauksenhallintatason prosesseista on testauksen tarkkailu ja hallitseminen, joka varmistaa, että testaus suoritetaan testaussuunnitelman ja organisaatiotason määritysten mukaisesti. Tämä prosessi myös ehdottaa tarvittaessa päivityksiä testaussuunnitelmaan. Kolmas testauksenhallintatason prosesseista on testauksen viimeistely. Se suoritetaan, kun hallinnan alaisena oleva testausprosessi on saatu valmiiksi. Testauksen viimeistelyn tarkoituksena on varmistaa muun muassa, että hyödylliset, uudelleenkäytettävissä olevat resurssit (esim. käytetyt testaussuunnitelmat) säilötään myöhempää käyttöä varten ja testitulokset dokumentoidaan ja toimitetaan eteenpäin niistä kiinnostuneille asianosaisille. [3]

ISO/IEC 29119 standardin toisen osan määrittelemän ohjelmistotestauksen prosessimallin alimman tason eli varsinaisen testaustason prosesseista käydään ensin läpi staattiset

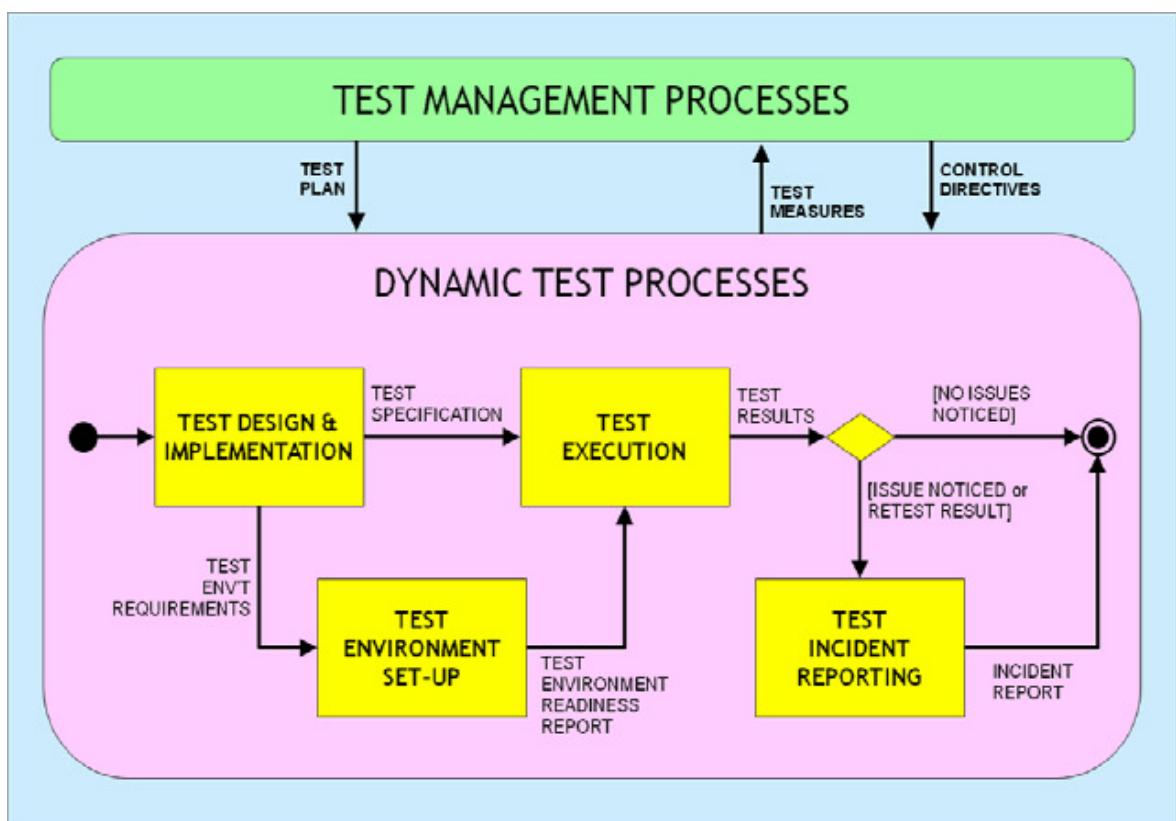
testausprosessit. Staattisella testauksella tarkoitetaan komponentin tai järjestelmän testaamista määrittelyn tai toteutuksen tasolla ilman ohjelmakoodin suorittamista eli esimerkiksi tarkastuksia/katselmuksia ja ohjelmakoodin staattista analysointia. Staattisia testausprosesseja on kolme, ja kuvassa 4 on esitetty niiden keskinäiset suhteet sekä suhteet testauksenhallintatasoon. Ensimmäinen prosesseista on staattisen testauksen valmistelu. Sen tarkoituksena on tunnistaa ja määrittellä staattisen testauksen vaatimat resurssit etukäteen ennen testaamisen aloittamista ja varmistaa, että kyseiset resurssit ovat käytettävissä. Toinen staattisista testausprosesseista on katselmointi/analysointi, jossa varsinainen staattinen testaus suoritetaan. Testattavaa kohdetta siis tarkastellaan esimerkiksi katselmuksen keinoin tai muuten analysoimalla. Tarkoituksena on löytää kohteesta mahdolliset virheet ja mitata samalla sen laatua. Löydetyt virheet voidaan joko korjata tai raportoida eteenpäin. Kolmas staattisista testausprosesseista on staattisen testauksen jälkiseuranta. Tämä prosessi pitää huolen siitä, että edellisessä vaiheessa havaitut ongelmat todella korjataan. [3, 4]



Kuva 4. Staattisten testausprosessien suhteet toisiinsa ja muihin mallin tasoihin. [3]

Ohjelmistotestauksen prosessimallin alimman tason toinen prosessiluokka on dynaamiset testausprosessit. Dynaamiseen testaukseen kuuluu komponentin tai järjestelmän ohjelmakoodin suorittaminen. Dynaamiset testausprosessit kuvaavat kuinka dynaamista testausta suoritetaan tietyssä testausvaiheessa (esim. yksikkö- tai järjestelmätestaus) tai

tietyssä testaustyyppissä (esim. suorituskyky- tai käytettävyytestaus). Dynaamisia testausprosesseja on neljä, ja kuvassa 5 on havainnollistettu niiden keskinäisiä vuorovaikutussuhteita sekä suhteita ylöspäin testauksenhallintatasoon. Ensimmäisenä prosesseista tulee testauksen suunnittelu ja toteutus. Tämä prosessi kuvaa kuinka testitapaukset ja testausproseduurit määritellään. Toinen dynaamisista testausprosesseista on testausympäristön luominen, jonka tarkoituksena on tuottaa tarkoitukseen sopiva testausympäristö ja ylläpitää sitä. Kolmas prosessi on nimeltään testauksen suorittaminen. Se pitää sisällään aiemmin määriteltyjen testausproseduurien suorittamisen käyttäen hyväksi aiemmin kehitettyä testausympäristöä. Neljäs dynaamisista testausprosesseista on testisattumusten raportointi. Tähän prosessiin päädytään, jos testauksessa tapahtuu jotain odottamatonta ja testi epäonnistuu tämän takia tai mikäli jostain muusta syystä vaaditaan uudelleentestausta. Prosessin tarkoituksena on raportoida lisätoimenpiteitä vaativat ongelmat eteenpäin niiden selvittämisestä vastaaville tahoille. [3, 4]



Kuva 5. Dynaamisten testausprosessien suhteet toisiinsa ja muihin mallin tasoihin. [3]

2.1.3 Osa 3: Testausdokumentaatio

ISO/IEC 29119 standardin kolmannen osan tarkoituksena on kuvailla edellisessä osassa määriteltyjen ohjelmistotestausprosessien aikana tuotettava testausdokumentaatio. Kolmas osa määrittelee erityyppisten testidokumenttien sisällön ja rakenteen, sekä tarjoaa malliksi esimerkkidokumentteja. Dokumenttien jaotteluun käytetään luonnollisesti samaa jakoa kolmeen eri tasoon kuin standardin toisessa osassa esitellyssä ohjelmistotestauksen prosessimallissakin. [5]

Organisaatiotasolla tuotetaan ja ylläpidetään kahta dokumenttia: organisaation testauspolitiikkaa ja organisaation testausstrategiaa. Testauspolitiikka määrittelee organisaatiossa suoritettavan testaamisen laajuuden, peruseriaatteet ja päämäärän (mitä testauksella saavutetaan). Sen asettamia toimintatapoja ja sääntöjä tulee noudattaa koko organisaation laajuudella. Testauspolitiikka tarjoaa myös kehyksen itse politiikan laatimiselle ja jatkuvalla kehittämiselle organisaatiossa. Organisaation testausstrategian tulee olla linjassa testauspolitiikan kanssa. Nämä kaksi dokumenttia ovat hyvin lähellä toisiaan, ja joissain tapauksissa voi olla myös mahdollista, että testauspolitiikka on itse asiassa sisällytetty testausstrategiaan eikä erillistä politiikkaa ole dokumentoitu lainkaan. Testausstrategia määrittelee uudelleenkäytettäviä suuntaviivoja testauksen suorittamiselle organisaatiossa. Myös testausstrategiaa sovelletaan koko organisaation testaustoimintoihin, joten politiikan tavoin se ei ole sidoksissa mihinkään tiettyyn projektiin. [5]

Testauksenhallintatasolla tuotetaan ja ylläpidetään kolmea erilaista dokumenttityyppiä: testaussuunnitelmaa, testauksen tilaraporttia ja testauksen valmistumisraporttia. Testaussuunnitelma tarjoaa yleisen suunnitelman testauksen suorittamisesta ja hallinnasta. On olemassa useita erityyppisiä testaussuunnitelmia riippuen siitä, missä kohdassa projektia ne on tuotettu (esim. projektin testaussuunnitelma tai yksikkötestaussuunnitelma). Testaussuunnitelma määrittelee yleensä muun muassa testauksen laajuuden, siinä käytettävät resurssit, testattavat välituotteet ja ominaisuudet sekä testauksen aikataulun. Organisaatiossa tulee olla yksi määritetty formaatti, jonka mukaan kaikki testaussuunnitelmat laaditaan. Testauksen tilaraportti tekee yhteenvedon tiettyjen testausaktiviteettien tilasta tai tuloksista, sekä mahdollisesti tarjoaa arvioita ja suosituksia

näiden pohjalta. Tilaraportti laaditaan usein tarvittaessa jonkin tahon erillisestä pyynnöstä ja se on luonteeltaan hyvin tilapäinen. Testauksen valmistumisraportti sisältää tiettyjen testaustoimien tulokset. Valmistumisraportin yksityiskohtaisuus voi vaihdella suuresti riippuen raportoidun testin tyypistä ja tuloksista. Esimerkiksi yksikkötestauksen tuloksen voidaan vain todeta olevan hyväksytty tai hylätty, kun taas käytettävyydestä testauksen valmistumisraportti voi olla huomattavasti yksityiskohtaisempi. [5]

Testaustasolla tuotetaan ja ylläpidetään neljää dokumenttityyppiä: testausspesifikaatioita, testausympäristön vaatimuksia, testausympäristön valmiusraporttia ja testisattumusraporttia. Testausspesifikaatioita on olemassa kolmenlaisia: yksi testisuunnitelmien kuvaamiseen, toinen testitapausten kuvaamiseen ja kolmas testiproseduurien määrittämiseen. Testisuunnitelmaspesifikaatio määrittää testattavat ominaisuudet sekä suoritettavat testiaktiviteetit. Testitapausspesifikaatio sisältää tarvittavat tiedot syötteistä ja tulosteista, sekä kaikki suoritettavat testitapaukset. Testiproseduurispesifikaatio määrittää tietyn kohteen testaamista varten suoritettavat toimenpiteet. Testausympäristön vaatimukset -dokumentti kuvailee sekä testausympäristöltä vaaditut että siltä toivotut ominaisuudet ja tarjoaa tarvittaessa muuta asiaan kuuluvaa tietoa tai viittauksia muihin tietolähteisiin. Testausympäristön valmiusraporttia käytetään testausympäristön valmiuden dokumentointiin, arviointiin ja hyväksyntään organisaatiossa. Testisattumusraporttiin dokumentoidaan testauksen aikana sattuneita ja lisätutkimusta vaativia tapahtumia. Testisattumusraporttia voidaan kutsua myös esimerkiksi ongelma- tai virheraportiksi. [5]

2.1.4 Osa 4: Testaustekniikat

ISO/IEC 29119 standardin neljännessä osasta ei ole vielä tätä kirjoitettaessa julkaistu edes draft-versiota, joten sen sisällöstä ei voi tässä vaiheessa vielä kertoa mitään kovin varmaa. Standardia valmistelevan työryhmän verkkosivujen perusteella neljäs osa tulee kuitenkin esittelemään muun muassa staattisia (esim. katselmukset, tarkistukset, läpikäynnit), toiminnallisia (esim. mustalaatikko-, lasilaatikkotestaus), ei-toiminnallisia (esim. suorituskyky-, turvallisuus-, käytettävyydestä testaus) ja kokemuspohjaisia (esim. virheiden arvailu, tutkiva testaus) testaustekniikoita. [6]

2.2 ISO/IEC 15504 prosessinarviointistandardi

Ohjelmistotestauksen adaptiivisen referenssimallin arviointia suunniteltaessa käytetään hyväksi ISO/IEC 15504 standardin määrittämiä. Joskus tätä standardia kutsutaan myös nimellä SPICE (Software Process Improvement and Capability Determination). Kyseinen standardi tarjoaa kehyksen ohjelmistotuotannon prosessien arvioimista varten. Standardin viides osa sisältää esimerkinomaisen prosessinarviointimallin ja tässä luvussa keskitytään pääasiassa tutustumaan siihen, koska se tarjoaa pohjan suunnitelmanmukaisen prosessiarvioinnin tekemiseen. [7]

ISO/IEC 15504 standardin viides osa esittelee prosessinarviointimallin, joka on luonteeltaan kaksiulotteinen. Se sisältää prosessiulottuvuuden ja kyvykkyydulottuvuuden. Prosessiulottuvuus määrittelee tarkasteltavat prosessit ja jakaa ne erilaisiin kategorioihin. Standardin viidennessä osassa on määritelty suuri joukko erilaisia prosesseja, jotka on jaettu järjestelmän elinkaaren prosesseihin ja ohjelmiston elinkaaren prosesseihin. Näiden kahden kategorian alla prosessit on vielä jaettu useisiin pienempiin prosessiryhmiin. Prosessit on määritelty siten, että jokaisella niistä on ainutlaatuinen tarkoituksen ja tulosten yhdistelmä. [7]

Kyvykkyydulottuvuus määrittelee joukon prosessiattributteja, jotka on lajiteltu eri kyvykkyydustasojen tasolle. Prosessiattribuutit ovat mitattavia tunnusmerkkejä prosessien kyvykkyydestä. Tiettyä kyvykkyydustasoa kuvaa joukko prosessiattributteja, jotka yhdessä mahdollistavat selkeän parannuksen kyvykkyydessä suorittaa jokin prosessi. Kyvykkyydustasojen tasoa on yhteensä kuusi kappaletta (tasot 0-5) ja määritettyjä prosessiattributteja yhdeksän. Standardi määrittelee myös erilaisia mittareita, joiden avulla on tarkoitus saada objektiivista näyttöä prosessiattribuuttien saavutusasteista kulloinkin arvioitavan prosessin tapauksessa. Arvosteltaessa sitä, kuinka hyvin prosessi saavuttaa tiettyjä prosessiattributteja, käytetään neliportaista asteikkoa, joka on määritelty ISO/IEC 15504 standardin toisessa osassa. Tämän asteikon avulla jokaisen prosessiattribuutin saavuttamisasteelle voidaan määrittää erillinen arvosana. Näistä prosessille muodostuu arvosanajakauma, jota kutsutaan prosessiprofiiliksi. Prosessinarviointimalli määrittää mitkä prosessiattribuuteista tietyn prosessin pitää saavuttaa yltääkseen tietylle kyvykkyydustasolle. Lyhyesti sanottuna prosessi saa sitä

paremman kyvykkyystasoluokituksen mitä paremmin määritelty ja hallittu se on, ja aivan korkeimmilla tasoilla prosessin on myös kehitettävä itseään jatkuvasti. [7, 8]

2.3 Ongelmat ohjelmistotestauskäytännöissä

Yleisen käsityksen saamiseksi ohjelmistotestauskäytännöissä esiintyvistä ongelmista tutustuttiin erityisesti kahteen aiheeseen liittyvään tieteelliseen artikkeliin. Ensimmäisessä artikkelissa [9] analysoidaan tutkimusta varten haastateltujen ohjelmistotuotannon alan organisaatioyksiköiden testauskäytännöissä havaittuja ongelmia. Tutkimus oli case study -tyyppinen ja sen haastatteluihin osallistui kaikkiaan 26 organisaatiota. Toisessa artikkelissa [10] esitellään tutkimus, jossa tutkittiin ohjelmistotestauskäytäntöjen tilaa Australialaisissa ohjelmistotuotannon organisaatioissa. Samalla saatiin selville myös niiden testauskäytännöissä esiintyviä ongelmia. Tämäkin tutkimus oli tyypiltään case study ja siinä järjestettyyn kyselyyn vastasi edustajia yhteensä 65 eri organisaatiosta.

Ensimmäisessä artikkelissa määritettiin tehdyn tutkimuksen perusteella seitsemän erilaista ongelmakategoriaa testaukseen liittyvistä aihealueista. Nämä kategoriat kuvaavat ohjelmistotestauskäytännöissä eniten ongelmia aiheuttavia työvaiheita tai asioita. Määritetyt kategoriat ja esimerkkejä eri kategorioiden sisältämistä ongelmista on kerätty taulukkoon 1. [9]

Näiden ongelmakategorioiden lisäksi tutkimuksessa kehitettiin havaittujen ongelmien pohjalta neljä hypoteesia, joita noudattamalla ohjelmistotestauskäytäntöjen suurimpia ongelmia voitaisiin ohjelmistotuotantoalan organisaatioissa välttää. Määritetyt hypoteesit ovat:

1. Arkkitehtuuri- ja tuotesuunnittelussa pitäisi keskittyä myös testattavuuteen.
2. Testausprosesseille tulee selkeästi määrittää niiden tarvitsemat resurssit, ja ne on myös erotettava projektin muista resursseista.
3. Testaustyökalujen ja testausautomaation valinnan tulisi perustua työkalujen käytettävyyteen ja konfiguroitavuuteen.
4. Testaus tulisi suorittaa siihen erikoistuneen henkilöstön toimesta. [9]

Taulukko 1. Ongelmakategoriat ja esimerkkiongelmia. [9]

Kategoria	Esimerkkiongelmät
Testaustyökalut	Monimutkaiset testaustyökalut aiheuttavat virheitä
Testausautomaatio	Ei sopivaa henkilöstöä käytettävissä, soveltamisen korkeat kustannukset
Tiedonkulku osakkaiden välillä	Huono dokumentaatio ja kommunikaatio aiheuttavat väärinkäsityksiä ja turhia investointeja
Tuotesuunnittelu	Epäselvät ominaisuudet, myöhäiset muutokset, huono testattavuus
Testausstrategia ja testauksen suunnittelu	Huonon aikataulu- tai resurssisuunnittelun takia kaikkia suunniteltuja testivaiheita ei voida suorittaa tai tarvitaan lisäaikaa niitä varten
Testaushenkilöstö	Ammattitaidon puute, testaamiseen erikoistuneen henkilöstön puute
Testausresurssit	Testausinfrastruktuurin puute ja sen hankkimisen korkeat kustannukset

Toisessa käsiteltävässä artikkelissa tutkittiin siis Australialaisten ohjelmistotuotanto-organisaatioiden testauskäytäntöjä. Tutkimuksesta käy ilmi myös muutamia yleisimpiä ongelmia, joita käytännöissä esiintyy. Testaustyökalujen käyttöönotto aiheuttaa selkeästi paljon ongelmia johtuen esimerkiksi työkalujen korkeista kustannuksista ja käyttöönoton vaikeudesta. Uusien työkalujen tai tekniikoiden käyttämisen opettelu vie nimittäin usein runsaasti aikaa. Nämä samat syyt hidastavat myös esimerkiksi testauksen mittaamisenmenetelmien tai testausstandardien laajempaa käyttöönottoa. [10]

Testaushenkilöstön puutteellinen ammattitaito on yksi merkittävä ongelma käytännön testauksessa. Monet testaajat eivät ole saaneet asianmukaista koulutusta testausmenetelmistä ja kouluttamisen suuret kustannukset eivät houkuttele yrityksiä

tarjoamaan testaajille lisäkoulutusta. Lisäksi testausbudjetit ovat harvoin realistisia, mikä aiheuttaa tietenkin ongelmia koko testausprosessin suorittamisen kannalta. Tutkimukseen osallistuneista organisaatioista vain yksi viidesosa mainitsi onnistuvansa pysymään budjettiansa määrittämässä rajoissa testauksen osalta. Tutkimuksessa havaittiin myös tarvetta testaamisen lopettamissääntöjen määrittämiselle ja niiden käyttöönotolle, sillä monissa organisaatioissa edelleen suoritettiin testausta periaatteella ”testataan kunnes resurssit loppuvat”. [10]

2.4 MASTO-projektin 4. kierroksen haastattelut

MASTO-projektin neljännellä haastattelukierroksella haastateltiin kymmenen eri ohjelmistoalan organisaation edustajia. Heiltä kyseltiin hieman heidän edustamiensa organisaatioiden testauskäytännöistä sekä myös erikseen mielipidettä ISO/IEC 29119 ohjelmistotestausstandardista. Tässä luvussa käydään läpi lyhyesti muutamia kiinnostavimpia huomioita, joita haastatteluaineistosta tehtiin. Näitä havaintoja pyritään myös käyttämään hyödyksi ohjelmistotestauksen adaptiivisen referenssimallin arviointia suunniteltaessa. [11]

Haastatteluissa monet vastaajat arvioivat, että ISO/IEC 29119 testausstandardi voisi olla hyödyllinen. Muun muassa sikäli, että se yhdistää useiden muiden käytössä olevien standardien ja käytäntöjen sisältämiä asioita kätevästi yhden standardin alle. Monissa organisaatioissa on selkeästi tarvetta testauskäytäntöjen parantamiselle monin tavoin, kuten esimerkiksi dokumentoinnin tai testausprosessien jatkuvan kehittämisen suhteen, koska liian harvat organisaatiot kehittävät prosessejaan aktiivisesti. Useilla organisaatioilla on käytössä jonkinlaisia testauspolitiikkoja ja -strategioita, mutta ne vaihtelevat suuresti sisältönsä ja laatunsa suhteen. Osa organisaatioista ei edes dokumentoi kyseisiä ohjeistuksia ja suuntaviivoja lainkaan. ISO/IEC 29119 standardin tarjoamat dokumenttimäärittelyt nähtäisiin sen takia hyödyllisinä. Ne yhtenäistäisivät organisaatioiden välisiä ja sisäisiä testauskäytäntöjä. Erityisesti korkean tason testaussuunnitteluprosessit ja testauksen parempi organisointi niiden avulla koettiin potentiaalisesti erittäin hyödylliseksi. [11]

Haastatteluaineistosta voi havaita, että standardin mahdolliseen käyttöönottoon

suhtaudutaan erikokoisissa organisaatioissa eri tavalla. Ongelmaksi nähdään etenkin sen käyttöönotto kokonaisuudessaan pienissä organisaatioissa, joiden testausresurssit ovat usein varsin rajalliset. Tämän takia standardiin toivottiin muun muassa jonkinlaisia prioriteettimääryksiä standardin kuvailemille aktiviteeteille, jotta sovellettavaksi voitaisiin valita esimerkiksi vain ne ehdottomasti tärkeimmät asiat. Eräänä kehityskohtena standardissa pidettiin myös sitä, että testauksen ja kehityksen välinen yhteys tuotaisiin selkeämmin esiin. Tämä on linjassa myös edellisessä luvussa käsiteltyjen testauskäytäntöjen ongelmien kanssa, koska siellähän painotettiin testattavuuteen keskittymistä ohjelmiston suunnittelu- ja kehitysvaiheissa. [11]

3 ARVIOINTISUUNNITELMAN ESITTELY

Ohjelmistotestauksen adaptiivista referenssimallia päädyttiin lopulta lähteä arvioimaan kahdelta eri kannalta. Ensiksi arvioidaan referenssimallin eli ISO/IEC 29119 standardin kuvaaman testausprosessin kyvykkyyttä käyttäen hyväksi ISO/IEC 15504 standardin viidennessä osassa määritettyä prosessinarviointimallia. Toiseksi arvioidaan referenssimallin yleistä käyttökelpoisuutta ja hyödyllisyyttä testauskäytännöistä havaittujen ongelmien pohjalta eli arvioidaan sitä, kuinka hyvin malli voisi auttaa organisaatioita kyseisten ongelmien välttämiseksi. Seuraavissa aliluvuissa esitellään tarkemmin suunnitelmat näiden kahden eri arviointitavan suorittamiseen.

3.1 Testausprosessin kyvykkyuden arviointi

Ohjelmistotestauksen adaptiivisen referenssimallin määrittämisen testausprosessin kyvykkyyttä tullaan arvioimaan ISO/IEC 15504 standardin viidennen osan prosessiarviointimallin avulla. Lähtökohtana on ajatus siitä, että referenssimallin eli ISO/IEC 29119 standardin määrittämää testausprosessia tarkastellaan yhtenä prosessina, vaikka se pitääkin sisällään suuren määrän pienempiä aliprosesseja. Tämän tarkoituksena on se, että tällä tavalla kyvykkyysarviointi pystytään kohdistamaan nimenomaan koko testausprosessin laajuudelle ja sen kyvykkyudesta saadaan kerralla kattava kokonaiskuva.

Ongelmana tämän kyvykkyysarvioinnin suorittamisessa on se, ettei standardi ole vielä yleisesti käytössä. Siitä syystä arviointia varten ei ole saatavilla varsinaista todistusaineistoa prosessin soveltamisesta käytännössä. Arvioinnin suorittamiseksi joudutaankin tyytymään vain ISO/IEC 29119 ohjelmistotestausstandardin testausprosessia koskeviin määrityksiin ja arvioimaan testausprosessia siis pelkästään standardidokumentin pohjalta.

Koska arvioinnin kohteena on nimenomaan testausprosessi, niin sen arviointia varten tulee valita arviointimallin määrittämistä prosessiluokista käyttöön Software verification ja Software validation -luokat. Nämä sopivat tarkoitukseen, sillä jo Edward Kit aikoinaan määritteli ohjelmistotestauksen koostuvan nimenomaan verifioinnista ja validoinnista

yhdessä ohjelmistotestauksen alan perusteoksista [12]. ISO/IEC 15504 standardin viidennen osan viides luku tarjoaa määriykset ja mittarit eri prosessiluokkien ykköstason kyvykkyuden arviointiin. Tästä luvusta tulee siis valita käytettäväksi aliluvut 5.6.4 Software verification ja 5.6.5 Software validation, joiden pohjalta arvioidaan saavuttaako ohjelmistotestauksen adaptiivisen referenssimallin prosessimalli ISO/IEC 15504 standardin määrittelemän ykköstason kyvykkyuden. Arviointia tehtäessä arvioijan kannattaa käyttää ISO/IEC 15504-2:ssa määritettyä neliportaista arvosteluasteikkoa jokaisen prosessiattribuutin saavuttamisen arviointiin. [7]

Ohjelmistotestauksen adaptiivisen referenssimallin testausprosessin kyvykkyuden arvioimiseen ykköstasoa korkeampien tasojen osalta tarvitaan ISO/IEC 15504 standardin viidennen osan kuudennen luvun määrittelemiä kyvykkyysmittareita. Kyseinen luku määrittelee mitä vaatimuksia prosessin tulee täyttää saavuttaakseen tietyn kyvykkyystason, ja kyvykkyysmittarit tarjoavat näyttöä tiettyjen prosessiattribuuttien saavuttamisesta helpottaen siten prosessiattribuuttien saavuttamisen arvioimista. ISO/IEC 15504-5:n kuudennessa luvussa esitetyt kyvykkyysmittarit ovat samat kaikille erilaisille prosesseille, joten siitä ei tarvitse erikseen etsiä ohjelmistotestausprosesseihin liittyviä mittareita. Käyttämällä näitä määritettyjä kyvykkyysmittareita arvioija pystyy muodostamaan arvosanjakauman eri prosessiattribuuttien saavuttamisasteista eli ns. prosessiprofiilin adaptiivisen referenssimallin testausprosessille. Tämän pohjalta voidaan myös määrittää prosessin saavuttama kyvykkyystaso. [7]

Mikäli ohjelmistotestauksen adaptiivisen referenssimallin määrittelemän testausprosessin arvioiminen kokonaisuudessaan osoittautuu liian raskaaksi, niin testaaja voi tarvittaessa oman harkintansa mukaan rajoittaa arvioinnin koskemaan esimerkiksi vain jotakin tiettyä testausprosessin osaa. Tällä tavalla saataisiin näyttöä edes mallin jonkin osan kyvykkyydestä, vaikka sen arvioiminen kokonaisuudessaan ei olisikaan mahdollista.

3.2 Käyttökelpoisuuden ja hyödyllisyyden arviointi

Tutustumalla sekä artikkeliin ohjelmistotestauskäytännöissä esiintyvien ongelmien analysoinnista että ohjelmistotestauskäytäntöjen tilaa Australiassa kartoittaneeseen tutkimukseen, saatiin hyvä käsitys niistä ongelmista, joita ohjelmistokehitysorganisaatiot

ohjelmistotestauksen alalla käytännössä kohtaavat. Tämän lisäksi käytiin vielä läpi MASTO-projektin neljännen kierroksen haastatteluja lisäinformaation toivossa. Näiden tietolähteiden pohjalta laadittiin lista kysymyksistä, joiden avulla ohjelmistotestauksen adaptiivista referenssimallia eli ISO/IEC 29119 standardia voitaisiin arvioida. Kysymysten tarkoituksena on siis kartoittaa sitä kuinka hyvin referenssimalli vastaa käytännön ongelmiin ohjelmistotestauksessa, jotta saadaan tietoa mallin käyttökelpoisuudesta ja hyödyllisyydestä käytännössä. Taulukossa 2 on esitetty kysymykset, joita suunnitellaan käytettävän ohjelmistotestauksen adaptiivisen referenssimallin arvioimiseen.

Kysymykset 1-7 perustuvat artikkeliin ohjelmistotestauksen käytännönongelmien analysoinnista [9]. Kysymykset 8-10 pohjautuvat Australian testauskäytäntöjä tutkineen tutkimuksen antiin [10]. Osa tutkimuksessa havaituista ongelmista oli päällekkäisiä edellisen artikkelin sisällön kanssa, ja siksi siitä ei saatu laadittua kovin montaa uutta kysymystä. Kysymykset 11-13 on muodostettu MASTO-projektin 4. kierroksen haastatteluaineistosta tehtyjen havaintojen pohjalta [11].

Arviointi on tarkoitus suorittaa siten, että arvioija vastaa kysymyksiin, ja pohtii kunkin kysymyksen kohdalla sitä, että miten ohjelmistotestauksen adaptiivinen referenssimalli eli ISO/IEC 29119 standardi ottaa kantaa kyseiseen ongelmakohtaan ja miten se voisi auttaa organisaatiota välttämään tuon nimenomaisen ongelman käytännössä. Kirjattuaan vastaukset ylös kaikkiin kysymyksiin hän voi sitten näitä vastauksia analysoimalla muodostaa jonkinlaisen käsityksen referenssimallin yleisestä hyödyllisyydestä ja käyttökelpoisuudesta.

Taulukko 2. Kysymykset käytettäväksi referenssimallin arvioinnissa.

1.	Miten standardi auttaa testaustyökalujen ja -automaation valinnassa sekä soveltamisessa?
2.	Miten standardi kehittää ohjelmistotestaukseen liittyvän tiedon kulkua organisaatioissa?
3.	Miten standardi ottaa kantaa tuotekehityksen ja testauksen suhteeseen? Auttaako se varmistamaan tuotteen hyvän testattavuuden?
4.	Miten standardi auttaa testaamisen suunnittelussa eri organisaatiotasoilla?
5.	Miten standardi ottaa kantaa testaushenkilöstön valintaan ja ammattitaidon kehittämiseen?
6.	Tarjoaako standardi ohjeistusta testausinfrastruktuurin luomiselle organisaatioissa?
7.	Miten standardi auttaa määrittämään testauksen vaatimat resurssit ja pitämään huolen, että ne pysyvät erillään projektin muista resursseista?
8.	Miten standardi pitää huolen siitä, että sen käyttöönotto on mahdollisimman helppoa?
9.	Miten standardi auttaa realistisen testausbudjetin laatimisessa?
10.	Tarjoaako standardi ohjeistusta testaamisaktiviteettien lopettamiskäytäntöihin tai määritteleekö se jotain käyttökelpoisia lopettamissääntöjä?
11.	Miten standardi auttaa erilaisen testaukseen liittyvän dokumentaation tuottamisessa ja ylläpitämisessä?
12.	Miten standardi edistää testausprosessien jatkuvaa kehittämistä organisaatioissa?
13.	Miten standardi skaalautuu sovellettavaksi tehokkaasti erikokoisissa organisaatioissa? Tarjoaako se mahdollisuuden valita helposti vain tärkeimmät osat otettaviksi käyttöön mikäli sitä ei halua soveltaa kokonaisuudessaan?

4 POHDINTA JA TULEVAISUUS

Tämän kandidaatintyön tarkoituksena oli laatia suunnitelma ohjelmistotestauksen adaptiivisen referenssimallin arvioimista varten. Valittuun lähdemateriaaliin tutustumalla saatiin varmasti suhteellisen todenperäinen kuva ohjelmistotestauskäytännöissä esiintyvistä ongelmista. Tästä syystä näiden mainittujen havaintojen perusteella laadittu arviointisuunnitelma tekee mahdolliseksi arvioida ohjelmistotestauksen adaptiivisen referenssimallin hyödyllisyyttä uskottavalla ja hyvin perustellulla tavalla. Referenssimallin määrittelemän testausprosessin arviointi ISO/IEC 15504 standardin (SPICE) pohjalta tulee myös antamaan vakuuttavaa näyttöä testausprosessin kyvykkyydestä, koska kyseinen arviointistandardi on laajalti käytetty. Arviointisuunnitelma on myös laajuudeltaan sopivan kompakti, jotta se voisi toimia hyvin MASTO-projektissa käytettävissä olevilla resursseilla. Kaiken kaikkiaan työn voidaan siis sanoa saavuttaneen päämääränsä ja tulos vaikuttaa varsin onnistuneelta.

Tämän työn puitteissa laadittua arviointisuunnitelmaa tullaan käyttämään ohjelmistotestauksen adaptiivisen referenssimallin arvioimiseen MASTO-hankkeen syksyn 2010 työpaketin mukaisesti. Arvioija voi yksinkertaisesti seurata tässä työssä kuvattua suunnitelmaa saadakseen jo näkyviä tuloksia referenssimallin hyödyllisyydestä ja sen määrittämän testausprosessin kyvykkyydestä. Ja mikäli arvioija ei halua noudattaa laadittua suunnitelmaa kirjaimellisesti, niin tämä suunnitelma tarjoaa kuitenkin käyttökelpoisen kehyksen, joka auttaa vähintäänkin hahmottamaan ja ohjaamaan arviointiprosessia.

5 YHTEENVETO

ISO/IEC 29119 ohjelmistotestausstandardi määrittelee ohjelmistotestaukselle yleisen prosessimallin, jota mikä tahansa organisaatio voi käyttää suorittaessaan minkälaista ohjelmistotestausta tahansa. Tätä mallia kutsutaan ohjelmistotestauksen adaptiiviseksi referenssimalliksi. Malli kuvaa erilaisia testausprosesseja, jotka toimivat eri tasoilla organisaatioyksikössä. Se myös määrittää minkälaisia toimintoja eri prosesseihin kuuluu ja minkälaista dokumentaatiota niistä tulisi tuottaa.

ISO/IEC 15504 prosessinarviointistandardi (SPICE) tarjoaa kehyksen ohjelmistotuotannon prosessien arvioimista varten. Se sisältää arviointimallin, joka määrittelee erilaisia prosessiluokkia, prosessien kyvykkyystasoja sekä prosessiattributteja ja mittareita, joita voidaan käyttää prosessien arvioimiseen. Lopputuloksena arvioiduille prosesseille määritetään kyvykkyystaso ja mahdollisesti myös tarkempi prosessiprofiili, jotka kertovat kuinka hyvin määriteltyjä ja hallittuja prosessit ovat.

Ohjelmistokehitysalan organisaatioiden testauskäytännöissä esiintyy paljon ongelmia. Kehitettävää löytyy esimerkiksi korkean tason testauksenhallintaprosesseista, dokumentointikäytännöistä, testauksen suunnittelusta, testausautomaation hyödyntämisestä sekä resurssien määrittämisestä ja hallinnasta.

Taustatutkimuksen perusteella laadittiin suunnitelma ohjelmistotestauksen adaptiivisen referenssimallin arvioimista varten. Suunnitelmassa mallia arvioidaan kahdelta eri kannalta. Ensiksi arvioidaan referenssimallin kuvaaman testausprosessin kyvykkyyttä käyttäen hyväksi ISO/IEC 15504 standardin viidennessä osassa määritettyä prosessinarviointimallia. Toiseksi arvioidaan referenssimallin yleistä käyttökelpoisuutta ja hyödyllisyyttä testauskäytännöistä havaittujen ongelmien pohjalta, eli arvioidaan sitä, kuinka hyvin malli voisi auttaa organisaatioita kyseisten ongelmien välttämässä.

LÄHTEET

1. MASTO Research Plan, 23.11.2009
2. MASTO-projektin verkkosivut, <http://www2.it.lut.fi/project/MASTO/index.html> (viitattu 28.7.2010)
3. ISO/IEC, “ISO/IEC 29119 CD Part 2: Test Process”, May 2010
4. ISO/IEC, “ISO/IEC 29119 WD Part 1: Concepts and Vocabulary”, March 2010
5. ISO/IEC, “ISO/IEC 29119 WD Part 3: Test Documentation”, January 2010
6. ISO/IEC 29119 Software Testing, <http://softwaretestingstandard.org/part4.php> (viitattu 2.8.2010)
7. ISO/IEC, “ISO/IEC 15504, Part 5: An exemplar Process Assessment Model”, April 2010
8. ISO/IEC, “ISO/IEC 15504, Part 2: Performing an Assessment”, November 2002
9. Kasurinen, J., Taipale, O., Smolander, K., “Analysis of Problems in Testing Practices”, *Asia-Pacific Software Engineering Conference (APSEC)*, December 2009
10. Ng, S.P., Murnane, T., Reed, K., Grant, D., Chen, T.Y., “A Preliminary Survey on Software Testing Practices in Australia”, *Proceedings of the 2004 Australian Software Engineering Conference (ASWEC'04)*, 2004
11. MASTO-projektin 4. kierroksen haastattelumateriaali
12. Kit, E., “Software Testing in the Real World: Improving the Process”, *ACM Press, Addison-Wesley*, 1995