

Teija Alasalmi

Ohjelmistotestauksen opetuksen tila suomalaisissa
korkeakouluissa: opinto-oppaiden sisältöanalyysi ja ehdotus
kehitystoimenpiteiksi

Tietotekniikan pro gradu -tutkielma

25. toukokuuta 2020

Jyväskylän yliopisto

Tietotekniikan laitos

Tekijä: Teija Alasalmi

Yhteystiedot: teija.alasalmi@gmail.com

Ohjaaja: Antti-Juhani Kaijanaho

Työn nimi: Ohjelmistotestauksen opetuksen tila suomalaisissa korkeakouluissa: opintooppaiden sisältöanalyysi ja ehdotus kehitystoimenpiteiksi

Title in English: Status of the teaching of software testing in Finnish universities: content analysis of study guides and proposal for development measures

Työ: Pro gradu -tutkielma

Suuntautumisvaihtoehto: Koulutusteknologia

Sivumäärä: 94+2

Tiivistelmä: Tässä laadullisessa pro gradu -tutkimuksessa kartoitettiin aineistolähtöisen sisällönanalyysin avulla mitä ohjelmistotestauksesta opetetaan suomalaisissa IT-alaa opettavissa ammattikorkeakouluissa ja yliopistoissa, miten testauksen opetus vastaa kansainvälisiä ohjelmistotekniikan opetussuosituksia ja miten testauksen opetusta voitaisiin kehittää.

Tutkimuksen tuloksista selviää, että yliopistoissa testauksen kurseja järjestetään pääosin vapaaehtoisina 5 opintopisteen laajuisina syventävinä opintoina maisteritasolla, kun taas ammattikorkeakouluissa testauksen kurssit ovat pääosin pakollisia, 3–5 opintopisteen laajuisia alempaan ammattikorkeakoulututkintoon kuuluvia ydinopintoja tai syventäviä ammattiopintoja. Yliopistoissa opetuksen painotus on testauksen peruskäsitteiden ja testausmenetelmien teoreettisessa opiskelussa sekä ohjelmistoturvallisuudessa, ammattikorkeakouluissa kurssisisällöt vaihtelivat enemmän testauksen perusteista käytettävyy-, peli-, haavoittuvuus- tai penetraatiotestaukseen ja sosiaaliseen hakkerointiin.

IT-alaa opettavissa yliopistoissa annettu testauksen opetus vastaa pääosin varsin hyvin kansainvälisiä suosituksia sekä määrällisesti että laadullisesti. Sen sijaan ainoastaan noin joka viidennessä IT-alaa opettavassa ammattikorkeakoulussa ohjelmistotestauksen opetus vastaa määrällisesti ja laadullisesti kansainvälisiä suosituksia. Testausopetuksessa on useimmissa

korkeakouluissa puutteita erityisesti seuraavissa aiheissa: suorituskykytestaus, testiautomaatio ja sen työkalut, mustalaatikkotestaustekniikat, testauksen hallinta sekä tietoturvatestaust. Tässä tutkimuksessa luotiin lisäksi 25 opintopisteen ohjelmistotestauksen perusopintokokonaisuus ratkaisuehdotukseksi korkeakoulujen ohjelmistotestauksen opetuksen kehittämiseen.

Avainsanat: ohjelmistotestaus, perusopinnot, testauksen opetus, aineistolähtöinen sisällönanalyysi

Abstract: The purpose of the present qualitative study was to find out what kinds of software testing related courses are taught at those Finnish universities of applied sciences and universities who arrange studies on IT field. Other aims were to find out how the teaching of software testing is in line with international guidelines and how the teaching of software testing can be developed.

The data was analyzed using content-based analysis. The results of the study show that in universities, testing courses are mainly arranged as volunteered studies consisting of 5 ECTS at the Master's level, whereas in applied universities the testing courses are essentially mandatory undergraduate studies consisting of 3-5 ECTS. In universities, the emphasis on teaching is on theoretical study of the basic concepts of testing and test methods and software security. In universities of applied sciences, the course contents varied more from the basics of software testing to usability, game testing, vulnerability or penetration testing and social hacking.

Teaching of software testing at the universities is largely in line with the international recommendations both quantitatively and qualitatively. Only about in every fifth university of applied sciences the software testing teaching met qualitative and quantitative recommendations. Teaching of software testing had deficiencies, particularly in the following topics: performance testing, test automation and its tools, black box testing techniques, test management and security testing. As a recommendation for developing software testing teaching in Finland, a basic study module on software testing consisting of 25 ECTS was created.

Keywords: software testing, study module, teaching software testing, content-based analysis

Termiluettelo

ACM	Association for Computing Machinery. Maailmanlaajuinen tieteellinen yhteisö, jonka tarkoituksena on edistää tietotekniikan tutkimusta sekä -opetusta.
AHOT	Aiemmin hankitun osaamisen tunnistaminen ja tunnustaminen.
FiSTB	Finnish Software Testing Board. ISTQB:n suomalainen jäsenjärjestö, joka vastaa ISTQB-sertifikaattikokeiden järjestämisestä Suomessa.
IEEE	IEEE Institute of Electrical and Electronics Engineers. Kansainvälinen tekniikan alan järjestö.
IEEE-CS	IEEE Computer Society. IEEE:hen kuuluva asiantuntijaryhmä, jonka tarkoituksena on edistää tietojenkäsittelytieteen ja -teknologian teoriaa, käytäntöä ja soveltamista.
ISTQB	International Software Testing Qualifications Board. Voittoa tavoittelematon järjestö, jonka tehtävänä on kehittää kansainvälistä testaaajien sertifiointijärjestelmää.
SE2014	Ohjelmistotekniikan alemman korkeakoulututkinnon uusien kansainvälinen opetussuositus.
SEEK	Software Engineering Education Knowledge. Ohjelmistotekniikan tietämyksen runko SE2014-opetussuosituksessa.
SWEBOK	Software Engineering Body of Knowledge. Ohjelmistotekniikan yleisesti hyväksytty tietämyksen runko.

Kuviot

Kuvio 1. Testauksen keskeiset aihealueet SWEBOK V3.0 -mallissa (kuva suomennettu teoksesta Bourque ym. 2014: 4-2)	5
Kuvio 2. Ohjelmistojen verifiointi ja validointi SE2014-opetussuosituksessa (ACM & IEEE-CS 2015).....	19
Kuvio 3. Ohjelmistoturvallisuus SE2014-opetussuosituksessa (ACM & IEEE-CS 2015).	21
Kuvio 4. Ohjelmistojen laatu SE2014-opetussuosituksessa (ACM & IEEE-CS 2015).....	22
Kuvio 5. Ohjelmistotestauksen opetus Suomen korkeakouluissa 2016–2017	41

Taulukot

Taulukko 1. ISTQB-sertifikaattisisältöjen opetukseen käytettävä suositeltu minimiaika...	15
Taulukko 2. Testaukseen sekä siihen liittyvien sisältöjen opiskeluun kuluva minimiaika SE2014-opetussuosituksessa	24
Taulukko 3. Testauksen kurssit ammattikorkeakouluissa lukuvuonna 2016–2017	44
Taulukko 4. Testauksen kurssit yliopistoissa lukuvuonna 2016–2017	50
Taulukko 5. Testauskurssien luokittelu testauksen aihealueisiin	55
Taulukko 6. Ohjelmistotestauksen perusteiden opetus IT-alan korkeakouluissa	62

Sisältö

1	JOHDANTO	1
2	TESTAUSOSAAMISEN KESKEISET SISÄLLÖT JA AIHEALUEET	3
2.1	Testaus ohjelmistokehityksen tietopohjassa	3
2.1.1	Taustaa.....	3
2.1.2	Ohjelmistotestauksen osaamisalue.....	4
2.2	ISTQB:n sertifikaattisisältö	8
2.3	Testaus ohjelmistotekniikan opetussuosituksissa	16
2.3.1	Alemman korkeakoulututkinon opetussuosituksiset	17
2.3.2	Ylemmän korkeakoulututkinon opetussuosituksiset	25
2.4	Testaus suomalaisten korkeakoulujen opetussuunnitelmissa	26
2.5	Ohjelmistotestauksen opetukseen liittyvät aiemmat tutkimukset Suomessa....	29
3	TUTKIMUSMENETELMÄ.....	32
3.1	Tutkimuksen tarkoitus ja tutkimuskysymykset	32
3.2	Tutkimuksen käsitteet, tutkimuskohde ja tutkimuksen eteneminen	33
3.3	Testauksen aihesisältöjä kuvaavien dokumenttien valinta	34
3.4	Tutkimusaineisto ja tutkimusmenetelmät	35
3.4.1	Tutkimusaineiston kokoaminen.....	35
3.4.2	Tutkimusaineiston analyysi ja luokittelu.....	37
3.5	Tutkimuksen rajaus	38
4	TUTKIMUKSEN TULOKSET	40
4.1	Ohjelmistotestauksen opetus Suomen korkeakouluissa	40
4.1.1	Ohjelmistotestauksen kurssit ammattikorkeakouluissa.....	42
4.1.2	Ohjelmistotestauksen kurssit yliopistoissa	48
4.1.3	Testauskurssien aihealueet	54
4.1.4	Testausopetuksen heikkoudet ja puutteet	64
4.2	Ohjelmistotestauksen perusopintokokonaisuus	67
4.2.1	Opintokokonaisuuden kuvaus ja osaamistavoitteet	68
4.2.2	Opintokokonaisuuden kurssien kuvaus ja osaamistavoitteet	70
5	POHDINTA.....	74
6	YHTEENVETO.....	80
	LÄHTEET	82
	LIITTEET	95
A	Ammattikorkeakoulujen ja yliopistojen testauskurssien luokittelu	95
B	Ohjelmistotestauksen perusopintokokonaisuus	98

1 Johdanto

Ohjelmistoala on noussut merkittäväksi osaksi suomalaista elinkeinoelämää. Tilastokeskuksen viimeisimpien toimipaikkatilastojen mukaan vuoden 2015 lopussa ohjelmistoalan yrityksiä oli yhteensä 10 246. Ohjelmistoalalla työskenteli Suomessa noin 60 000 henkilöä ja alan yritysten yhteenlaskettu liikevaihto oli noin 6,6 miljardia euroa. Suomessa ohjelmistoala on erittäin voimakkaasti keskittynyt Uudellemaalle ja erityisesti pääkaupunkiseudulle, ja siten myös ohjelmistotestauksen työpaikat ovat useimmiten pääkaupunkiseudulla. (Työ- ja elinkeinoministeriö 2016.)

Ohjelmistotestauksen ja ammattitaitoisten testaaajien kysyntä on kasvanut suuresti etenkin 2000-luvulla. Syynä tähän on ohjelmistojen monimutkaistuminen, sillä ohjelmistotuotteista kehitetään nykyään entistä monipuolisempia. Lisäksi digitalisoituminen on nopeutunut viimeisen muutaman vuoden aikana, ja Internet-pohjaiset sovellukset ovat tulleet tietokoneiden lisäksi erilaisiin tablettitietokoneisiin, älypuhelimiin sekä älytelevisioihin. (Työ- ja elinkeinoministeriö 2016, Tekniikan maailma 2015, Tivi 2010.)

Ohjelmistojen kehittyessä tarvitaan entistä laajempaa testausosaamista, ja lisäksi muun muassa verkkokauppojen ja pilvipalveluiden yleistymisen on lisännyt testauksen tarvetta. IT-alan kilpailun kovenemisen myötä suomalaiset yritykset ovat alkaneet panostaa enemmän ohjelmistojen laatuun, mikä on näkynyt testausalan työpaikkojen lisääntymisenä. Varsinkin pienet ja keskisuuret palveluntarjoajat rekrytoivat aktiivisesti uusia testauksen ammattilaisia, ja testausta myös ulkoistetaan entistä enemmän kokonaisuudessaan testausalan palveluyrityksille. (Tivi 2010, Capgemini 2011, Marketvisio 2013.)

Ohjelmistotestaus erityisosaamista tarjoavana tietotekniikkapalveluna kuuluu Tilastokeskuksen TOL 2008 -toimialaluokituksen mukaiseen toimialaluokkaan 62:n alaluokkaan 6201 ohjelmistojen suunnittelu ja valmistus (Tilastokeskus 2016). Ohjelmistoalan osa-alueet voidaan yksinkertaistaa *ohjelmistotuotteisiin* (esim. mobiilisovellukset, pelit), *asiakaskohtaisiin ohjelmistoihin* (esim. yrityskohtaiset IT-ratkaisut) sekä *sulautettuihin ohjelmistoihin* (esim. matkapuhelimien ohjelmistot) (Työ- ja elinkeinoministeriö 2016). Ohjelmistotestauksessa

testataan kaikkia edellä mainittuja ohjelmistoja. Erilaisten ohjelmistojen testaus ja siihen liittyvät osa-alueet vievät tavallisesti arviolta yli puolet ohjelmistoprojektin resursseista. Lisäksi pahimmillaan huonosti suoritetusta tai puutteellisesta testauksesta läpi päässeet ohjelmistovirheet aiheuttavat suurimmat kustannukset. (Haikala & Mikkonen 2011.)

Vaikka testauksen merkitys ohjelmistokehityksessä on suuri ja testauksen tehtävät vaativat erikoisosaamista, testaus jätetään Suomessa melko vähälle huomiolle korkeakoulujen opetussuunnitelmissa. Lisäksi IT-alan koulutus ei ole pysynyt kasvaneen testauksen ammattilaisten kysynnän tahdissa, sillä esimerkiksi Suomessa yliopistojen tarjonnassa on tyypillisesti vain 1–2 testauksen kurssia (Kyttälä 2011). IT-alan oppilaitokset kouluttavat ohjelmisto-osaajia, mutta suomalaisista oppilaitoksista ei voi edelleenkään suoraan valmistua ohjelmistotestaajan ammattiin. (Tivi 2010, Tivi 2013.) Lisäksi työelämän näkökulmasta katsottuna korkeakouluista valmistuneiden perustiedot ohjelmistotestauksesta ovat usein puutteellisia, eikä vastavalmistuneilla ole edellytyksiä aloittaa testaustyötä käytännössä ilman maksullista testauksen lisäkoulutusta (Korjus 2015).

Tämän tutkimuksen yhtenä tavoitteena on kartoittaa ohjelmistotestauksen opetuksen nykytilannetta Suomessa selvittämällä millaisia ohjelmistotestauksen kursseja suomalaiset ammattikorkeakoulut ja yliopistot tarjoavat. Lisäksi selvitetään, miten korkeakoulujen testausopetus vastaa kansainvälisiä opetussuosituksia. Tutkimuksen päällisenä tavoitteena on selvittää, miten ohjelmistotestauksen opetusta voitaisiin kehittää.

Seuraavassa luvussa 2 kuvataan keskeisiä testausosaamisen sisältöjä määritteleviä dokumentteja. Luvussa 3 esitellään tämän tutkimuksen tarkoitus, tutkimuskysymykset, tutkimuskohde, tutkimusmenetelmät, tutkimusaineisto sekä aineiston rajaamiseen vaikuttaneet seikat. Luvussa 4 esitellään tutkimuksen tulokset eli ohjelmistotestauksen opetuksen nykytilanne Suomen korkeakouluissa. Lisäksi Luvussa 4 esitetään tutkimustulosten ja valitun lähdekirjallisuuden pohjalta laadittu ohjelmistotestauksen perusopintokokonaisuuden sisältö, opintokokonaisuuden tavoitteet sekä opintokokonaisuuteen kuuluvat kurssit ja kurssikohtaiset osaamistavoitteet. Luvussa 5 pohditaan tutkimuksen tuloksia ja luotettavuutta. Lopuksi Luvussa 6 tehdään yhteenvetoa tutkimuksesta ja annetaan jatkotutkimusideoita tutkimusaiheeseen liittyen.

2 Testausosaamisen keskeiset sisällöt ja aihealueet

Testausosaamista voidaan kuvata monella eri tapaa. Esimerkiksi työpaikkailmoituksissa listataan yleensä monenlaisia testaukseen liittyviä tietoja ja taitoja, joita testaustehtävään valittavalta henkilöltä odotetaan. Testauksen työtehtävät vaativat usein tietoteknistä erityisosaamista, ja tyypillisiä tehtävänimikkeitä ovat testaaja, testausinsinööri, testaussuunnittelija, testausvastaava, testauskoordinaattori, testauspäällikkö, testausasiantuntija, testausarkkitehti, testausautomaation asiantuntija, valkohattuhakkeri, tietoturvatestaaja tai testauskonsultti. Testausosaamista, testauksen sisältöjä ja aihealueita määritellään myös erilaisissa dokumenteissa, joita käytetään apuna esimerkiksi korkeakoulujen opetussuunnitelmien tai erilaisten kurssien suunnittelussa.

Tässä luvussa tarkastellaan testausosaamisen keskeisiä sisältöjä ja aihealueita määritteleviä dokumentteja, joita ovat SWEBOK, ISTQB:n sertifiointisisältö sekä ohjelmistotekniikan kansainvälinen SE2014-opetussuositus. Kyseisiä dokumentteja on hyödynnetty myöhemmin tässä tutkimuksessa aineistonanalyysissä sekä muodostaessa ohjelmistotestauksen perusopintokokonaisuuden sisältöjä. Lisäksi luvussa kuvataan, miten testaus näkyy suomalaisen korkeakoulujen opetussuunnitelmissa ja miten opintoja mitoitetaan. Lopuksi tarkastellaan Suomessa tehtyjä ohjelmistotestauksen opetukseen liittyviä aiempia tutkimuksia.

2.1 Testaus ohjelmistokehityksen tietopohjassa

Tässä alaluvussa kuvataan ensin lyhyesti ohjelmistokehityksen yleisesti hyväksytty tietopohja SWEBOK sekä sen tavoitteet ja rakenne. Luvun pääpaino on SWEBOK-malliin kuuluvan ohjelmistotestauksen osaamisalueen esittelyssä.

2.1.1 Taustaa

Ohjelmistokehityksen tietopohja, SWEBOK, (engl. *Software Engineering Body of Knowledge*) on IEEE Computer Society -organisaation julkaisema malli, joka määrittelee yleisesti hyväksytyn tietämyksen rungon ohjelmistotekniikan osaamisalueille. SWEBOK-mallissa

määritellyt osaamistavoitteet on suunnattu usean vuoden työelämässä olleille, Yhdysvaltalaisessa korkeakoulujärjestelmässä tutkinnon suorittaneille ohjelmistokehittäjille. (Bourque ym. 2014.) SWEBOK-mallin viisi tavoitetta (Bourque ym. 2014) ovat:

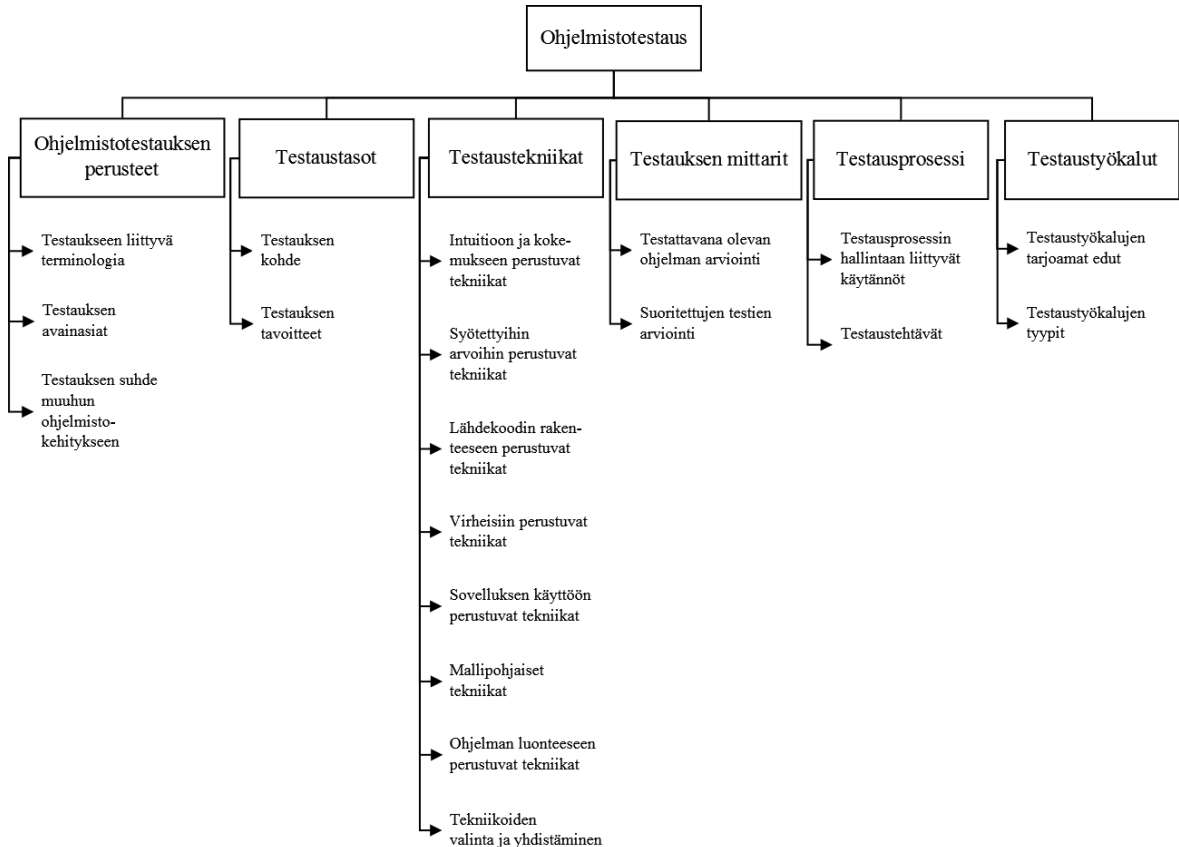
1. Edistää maailmanlaajuisesti yhtenäistä näkemystä ohjelmistotekniikkaan.
2. Määrittellä ja selventää ohjelmistotekniikan laajuus ja asema suhteessa muihin tieteenaloihin kuten tietotekniikkaan, projektinhallintaan, tietokonetekniikkaan ja matematiikkaan.
3. Kuvailta ohjelmistotekniikan sisältöjä.
4. Tarjota pääsy aihekohtaisiin kokonaisuuksiin jaettuun ohjelmistotekniikan tietopohjaan.
5. Tarjota perusta opetussuunnitelmien kehitykselle ja yksittäisten sertifikaattien materiaalille.

Ensimmäinen versio SWEBOK:sta julkaistiin vuonna 2004 ja viimeisin versio on julkaistu vuonna 2014. Alkuperäinen SWEBOK sisälsi kymmenen ydinosamisen tietoaletta (engl. *knowledge areas*). Mallin päivitetyssä vuoden 2014 versiossa on 15 osaamisaluetta. (Abran, Moore, Bourque, Dupuis & Tripp 2004; Bourque ym. 2014.) Kunkin osaamisalueen määrittelyssä kuvataan kyseisen osaamisalueen laajuus ja suhteet muihin osaamisalueisiin. Lisäksi määrittelyssä kuvataan kunkin osaamisalueen hierarkkinen rakenne, osa-alueet sekä keskeisimmät aihealueet, joista osaamisalueen tietämys rakentuu. Jokaisen osaamisalueen lopussa on viittaukset lähdekirjallisuuteen, joiden perusteella osaamisalue on luotu. Osaamisalueessa on lisäksi kuvattu taulukossa missä lähdemateriaalissa kutakin osaamisalueen aihetta on käsitelty. (Bourque ym. 2014.) SWEBOK:ssa testauksta käsitellään laajemmin Ohjelmistotestauksen osaamisalueessa (engl. *Software Testing*). Kyseisen osaamisalueen sisältöä kuvataan seuraavissa kappaleissa.

2.1.2 Ohjelmistotestauksen osaamisalue

SWEBOK-mallissa esitetty hierarkkinen rakenne ohjelmistotestauksen osaamisalueesta ja siihen sisältyvistä kuudesta keskeisestä aihealueesta on esitetty Kuviossa 1. Keskeisiä aihe-

alueita ovat ohjelmistotestauksen perusteet, testaustasot, testaustekniikat, testauksen mittarit, testausprosessi ja testaustyökalut. Nämä aihealueet jaetaan edelleen pienempiin osa-alueisiin, joita on yhteensä 19 kappaletta. Ohjelmistotestauksen osaamisalueen sisällöt on lyhyesti esitelty alla olevissa kappaleissa.



Kuvio 1. Testauksen keskeiset aihealueet SWEBOK V3.0 -mallissa (kuva suomennettu teoksesta Bourque ym. 2014: 4-2)

SWEBOK:in ohjelmistotestauksen osaamisalueen ensimmäinen aihealue käsittelee ohjelmistotestauksen perusteita (engl. *Software Testing Fundamentals*). Siihen kuuluvat testaukseen liittyvät määritelmät, termit ja sanasto sekä ohjelmistotestauksen avainasiat. SWEBOK-mallissa ohjelmistotestauksen avainasioiksi luetaan testitapausten valintamenetelmät, testauksen riittävyyskriteerit ja lopetusehtojen määrittely, testauksen tehokkuus ja tavoitteet, virhetilojen tunnistaminen, oraakkeliongelma, testauksen teoreettiset ja käytännön rajoitteet, saavuttamattomat ohjelman osat sekä testattavuus. Sanaston ja avainasioiden lisäksi aihealueessa käsitellään testauksen suhdetta muuhun ohjelmistokehitykseen. (Bourque ym. 2014.)

Toisessa aihealueessa käsitellään testaustasoja (engl. *Test Levels*). SWEBOK-mallissa testaustasot voidaan hahmottaa sekä testauksen kohteiden mukaan, että testauksen tavoitteiden mukaan. Testauskohteita ovat esimerkiksi yksittäinen ohjelman osa, ohjelman eri osien muodostama kokonaisuus tai kokonainen ohjelma. Näitä kohteita testataan vastaavilla testaustasoilla yksikkötestauksessa, integrointitestauksessa ja järjestelmätestauksessa. Testauksen tavoitteissa puolestaan kuvataan, miten testausta tehdään testattaessa ohjelman toiminnallisia ja ei-toiminnallisia vaatimuksia. Tällöin testaustasoja ovat esimerkiksi toiminnallinen testaus, suorituskykytestaus, luotettavuustestaus ja käytettävyydestestaus. (Bourque ym. 2014.)

Kolmannessa ohjelmistotestauksen aihealueessa kuvataan erilaisia testaustekniikoita (engl. *Test Techniques*). SWEBOK:ssa testaustekniikoiden jaottelu perustuu siihen, miten testitapausta halutaan kehittää. SWEBOK-mallissa testaustekniikat jaetaan 8 ryhmään: intuitioon ja kokemukseen perustuviin, syötettyihin arvoihin perustuviin, lähdekoodiin perustuviin, virheisiin perustuviin, sovelluksen käyttöön perustuviin, ohjelman luonteeseen perustuviin, tekniikoiden valintaan ja yhdistelyyn perustuviin sekä mallipohjaisiin testaustekniikoihin. Malliin on otettu mukaan yleisimmät testaustekniikat, jollaisia ovat esimerkiksi tutkiva testaus, virheenarvaus tai raja-arvoanalyysi. (Bourque ym. 2014.)

Neljäs aihealue perehdyttää testaukseen liittyviin mittareihin (engl. *Test-Related Measures*). Testauksen mittareita käytetään yleisesti ohjelmiston laadun analysoimisessa ja laadun parantamisessa, mutta myös testitapausten suunnittelussa ja suoritettavien testitapausten valinnassa. Lisäksi testauksen mittareita käytetään testausprosessin edistymisen valvonnassa. SWEBOK-mallissa testauksen mittarit jaetaan kahteen kategoriaan: mittareihin, joilla arvioidaan testauksen kohteena olevaa ohjelmaa sekä mittareihin, joilla mitataan testaamisen tehokkuutta ja kattavuutta. Ohjelman laadullista arviointia voidaan tehdä esimerkiksi analysoimalla ohjelman kokoa, rakennetta tai monimutkaisuutta, luokittelemalla löydetty virheet tyyppin ja vakavuuden mukaan, mittaamalla vikatiheyttä tai arvioimalla ohjelman luotettavuutta luotettavuuden kasvumallin perusteella. Testaamisen tehokkuutta ja kattavuutta voidaan mitata esimerkiksi erilaisilla kattavuusmittareilla, virheiden kylvämällä tai mutaatio-testauksen keinoin. (Bourque ym. 2014.)

Viides aihealue käsittelee testausprosessia (engl. *Test Process*), josta nostetaan esiin testausprosessin hallintaan liittyvät käytännöt sekä testauksen tehtävät. SWEBOK-mallissa testauksen hallintaan sisältyviä aiheita ovat testaukseen liittyvät asenteet ja testauslähtöinen ohjelmistokehitys, testaustiimin muodostus ja roolit, testauksen priorisointi ja kohdentaminen, testauksen organisointi, testauksen dokumentointi, testausprosessin mittaaminen ja kustannustehokkuuden arviointi sekä testauksen päättämiseen liittyvät lopetusehdot. SWEBOK-mallissa esitetyt testauksen keskeisimmät tehtävät ovat testauksen suunnittelu, testitapausten kehittäminen, testiympäristön rakentaminen, testien suorittaminen, testitulosten analysoiminen, havaintojen raportointi sekä virheiden analysointi ja hallinta. (Bourque ym. 2014.)

Kuudennessa aihealueessa käsitellään ohjelmistotestauksen työkaluja (engl. *Software Testing Tools*). SWEBOK-mallissa testaustyökaluihin liittyviä aiheita ovat työkalujen tuomat edut, työkalujen valintaan liittyvät seikat sekä erilaiset työkalutyypit. Mallissa testauksen työkalut jaotellaan seuraavaan 8 eri tyyppiin: testiympäristö, testigeneraattorit, nauhoitus- ja toistotyökalut, tiedonvertailutyökalut, koodianalysaattorit, jäljittimet, regressiotestaustyökalut ja luotettavuuden arviointityökalut. Testiympäristö koostuu tyngistä ja ajureista, joita tarvitaan testien suorittamiseen. Testigeneraattoreiden avulla voidaan automaattisesti generoida testitapauksia tai testidataa, ja nauhoitus- ja toistotyökalujen avulla voidaan puolestaan automatisoida uudelleentestausta. Erilaisia oraakkeli-, tiedostonvertailu- ja tarkastustyökaluja voidaan hyödyntää testitulosten analysoinnin automatisoinnissa. Koodianalysaattoreilla mitataan suoritetuilla testeillä saavutettua koodikattavuutta, ja jäljittimet tallentavat suoritettujen testien historiatietoja. Regressiotestaustyökalut tukevat testitapausten uudelleensuorittamista tapauksissa, joissa testauksen kohteena olevan ohjelmiston osa on muuttunut. Luotettavuuden arviointityökaluja voidaan hyödyntää testitulosten analysoinnissa ja graafisessa esittämisessä. (Bourque ym. 2014.)

SWEBOK-mallissa esitetty ohjelmistotestauksen aihealueiden kuvaus valittiin yhdeksi tämän tutkimuksen taustamateriaaliksi, koska malli soveltuu yleisellä tasolla tietotekniikan alan opetussuunnitelmien kehittämiseen. SWEBOK-mallia on jo aiemmin käytetty pohjana luotaessa kansainvälisiä tietotekniikan alan opetussuosituksia. Lisäksi SWEBOK-mallin yhtenä tavoitteena on tarjota perusta opetussuunnitelmien kehitykselle ja yksittäisten sertifi kaattien materiaalille, joten mallin oletetaan sopivan myös testauksen opintokokonaisuuden

kehittämiseen. SWEBOK-mallia voi lisäksi käyttää vapaasti myös ohjelmistotestauksen opetustarkoituksiin.

Vaikka SWEBOK-mallissa ohjelmistotestauksen perusasiat on melko kattavasti esitetty, on mallissa testauksen näkökulmasta myös heikkouksia. Kasurisen (2013) mukaan SWEBOK-malli ei esimerkiksi huomioi sitä, että testaustoiminta voidaan aloittaa jo ennen varsinaista testauksen suorittamista muun muassa kunnollisella arkkitehtuurisuunnittelulla. Lisäksi on syytä muistaa, että SWEBOK-malli on suunnattu valmistuneille, usean vuoden työelämässä olleille ohjelmistokehittäjille, eikä SWEBOK-mallin tavoitteena ole tarjota kattava katsaus ohjelmistotestauksen aiheisiin ohjelmistotestaaajia varten. SWEBOK-mallissa ohjelmistotestauksen erikoisosa-alueet kuten tietoturvatestaus, käytettävyytestaus tai testausautomaatio on kuvattu ainoastaan muutamilla lauseilla testauksen tavoitteet -luvun alla. Myös testauksen hallintaan liittyvä osuus on melko suppea verrattuna esimerkiksi aihetta käsitteleviin ISTQB:n sertifikaattimateriaaleihin. Näin ollen SWEBOK-mallissa esitetty ohjelmistotestauksen aihealueiden kuvaus ei selvästikään yksinomaan riitä, jos halutaan kehittää ohjelmistotestauksen opetusta kattamaan myös testauksen erikoisosaamisalueita. Tästä syystä seuraavissa luvuissa tarkastellaan ISTQB:n sertifikaattisisältöjä ja niissä esitettyjä testauksen aihealueita.

2.2 ISTQB:n sertifikaattisisältö

ISTQB (engl. *International Software Testing Qualifications Board*) on vuonna 2002 perustettu voittoa tavoittelematon järjestö, jonka tehtävänä on kehittää kansainvälistä testausalan ammattipätevyyden sertifiointijärjestelmää ja edistää testausta ammattina tuomalla yhteen testausalan ammattilaiset kansainvälisesti sekä kannustamalla testausalan tutkimukseen (ISTQB 2020a; ISTQB 2020b). ISTQB on myöntänyt yli 673 000 sertifikaattia 120 eri maassa. ISTQB-järjestöön kuuluu tällä hetkellä 63 paikallisjärjestöä, jotka edustavat 100 eri maata. (ISTQB 2019a.) Suomessa ISTQB:n paikallisjärjestönä toimii FiSTB (engl. *Finnish Software Testing Board*), joka järjestää Suomessa ISTQB perustason sertifikaattikokeita suomeksi ja englanniksi ja jatkotason sertifikaattikokeita englannin kielellä. Sertifikaattikokeet ovat monivalintakokeita, ja läpipääsyyn vaaditaan vähintään 65 % vastauksista oikein. Lisäksi FiSTB järjestää kansainvälisiä Testing Assembly -seminaareja ja tekee yhteistyötä

suomalaisten oppilaitosten kanssa järjestämällä vierailuluentoja oppilaitosten toivomista testausaiheista (K. Kakkonen, henkilökohtainen tiedonanto, 2.2.2018). Suomessa ISTQB:n sertifikaatteja on voinut suorittaa FiSTB:n järjestämässä kokeissa vuodesta 2010 lähtien, ja nykyään myönnettyjä sertifikaatteja on jo noin 4000. (FiSTB 2020; FiSTB 2016.)

ISTQB:n sertifiointijärjestelmällä on useita tavoitteita. Sertifiointijärjestelmä pyrkii edistämään ohjelmistotestauksen arvostusta ammattialana ja testausta ammattina. Se pyrkii tarjoamaan polkuja ohjelmistotestaajien urakehitykselle ja luomaan testaaajille mahdollisuuksia saada tunnustettu pätevytyminen omalla äidinkielellään. Lisäksi järjestelmällä halutaan mahdollistaa testauslaitojen vertailu ja testaaajien siirtyminen eri maiden välillä esimerkiksi työn perässä. ISTQB:n sertifiointijärjestelmä kokoaa testauksen tietopohjaa (engl. *Testing Body of Knowledge*) kehittämällä sanastoa sekä hyödyntämällä IT-alan parhaita käytäntöjä ja kaikkein innovatiivisinta tutkimusta. Järjestelmä asettaa kriteerit koulutusten tarjoajille, jotta testauksen tietopohjaa opetetaan johdonmukaisesti eri maissa, ja kehittää kaikille avointa kansainvälistä testausyhteisöä, joka on sitoutunut ohjelmistotestaukseen liittyvän tiedon, ideoiden ja innovaatioiden jakamiseen. Näiden lisäksi sertifiointijärjestelmä kansainvälisenä aloitteena pyrkii edistämään suhteita korkeakouluihin, eri maiden hallituksiin, tiedotusvälineisiin, ammattiliittoihin ja muihin asianosaisiin. (ISTQB 2020b.)

ISTQB:n kehittämä sertifiointijärjestelmä perustuu avoimesti saatavilla olevaan sertifikaattisisältöön (engl. *syllabus*), sanastoihin (engl. *glossary*) ja yhteisiin koesääntöihin (engl. *rules*). Sertifiointijärjestelmä sisältää kolme tasoa, joita ovat perustaso (engl. *Foundation level*), jatkotaso (engl. *Advanced level*) ja asiantuntijataso (engl. *Expert level*). Näiden lisäksi sertifikaatit on ryhmitelty kolmeen eri osaamispolkuun, joita ovat ketterä polku (engl. *Agile*), ydinpolku (engl. *Core*) ja erikoistumispolku (engl. *Specialist*). Tällä hetkellä jokaisella osaamispolulla voi suorittaa vähintään perus- ja jatkotason sertifikaatteja, ja näiden lisäksi ydinpolulla voi suorittaa vielä kahta erilaista asiantuntijataso sertifikaattia. (ISTQB 2020c; ISTQB 2020d.) Suoritettujen perus- ja jatkotason sekä perustason laajennusten sertifikaatit ovat voimassa ikuisesti, asiantuntijataso sertifikaatti on voimassa 5 vuotta sertifikaatin myöntämishetkestä (ISTQB 2020e).

ISTQB:n sertifiointijärjestelmässä jokaisen osaamispolun lähtöpisteenä on kaikille poluille yhteinen perustason testausertifikaatti. Perustason sertifioitu testaaja -sertifikaatti (engl. *Foundation Level, Certified Tester*) on suunnattu erityisesti testauksen pariin aikoville tai IT-alan ammattilaisille, jotka tarvitsevat työssään ohjelmistotestauksen perusteiden ymmärtämistä. Perustason sertifikaatin voi siis hankkia kuka tahansa aiheesta kiinnostunut osallistumalla maansa ISTQB:n paikallisjärjestön järjestämään koetilaisuuteen. Kokeeseen valmistava kurssi ei ole edellytys kokeen suorittamiseen, vaan henkilö voi ilmoittautua kokeeseen opiskeltuaan ISTQB-perustason sertifikaattisisällön ja sanaston. (ISTQB 2018a: 7.) Perustason sertifikaatti perehdyttää testauksen perusteisiin, testaukseen ohjelmiston elinkaaren aikana, testisuunnittelutekniikoihin ja staattisiin tekniikoihin, testauksen hallinnan perusteisiin sekä testauksen työkaluihin (ISTQB 2019b: 11). Perustason sertifikaatin suoritus vaaditaan, jotta henkilö voi osallistua eri osaamispolkujen perustason laajennusten sertifikaattikokeisiin tai jatko- ja asiantuntijatason sertifikaattikokeisiin (ISTQB 2020f).

Perustason sertifikaattisisältöjen laajennuksia on tällä hetkellä (toukokuussa 2020) kahdeksan: ketterä testaaja (engl. *Agile Tester*), mallipohjainen testaaja (engl. *Model-Based Tester*), käytettävyydestestaaja (engl. *Usability Testing*), hyväksymistestaaja (engl. *Acceptance Testing*), suorituskykytestaaja (engl. *Performance Testing*), autoteollisuuden ohjelmistojen testaaja (engl. *Automotive Software Tester*), uhkapeliteollisuuden testaaja (engl. *Gambling Industry Tester*) sekä mobiilisovellusten testaaja (engl. *Mobile Application Testing*) (ISTQB 2020d). Laajennosten englanninkielissä nimissä on käytetty sanoja *Tester* ja *Testing*, mutta tässä tutkimuksessa päädyttiin suomentamaan yhtenäisyyden vuoksi kaikki laajennokset testaaja-loppupäätteisiksi. Seuraavissa kappaleissa esitellään laajennokset lyhyesti.

Ketterä testaaja -laajennus on suunnattu ketterissä ympäristöissä työskenteleville ammattilaisille, kuten esimerkiksi testaajille, testausinsinööreille, testianalysioijille, testauskonsulenteille, testauspäälliköille ja ohjelmistokehittäjille (ISTQB 2014a: 7). Laajennus antaa perustietämyksen ketterästä ohjelmistokehityksestä, ketterän testauksen periaatteista, käytännöistä ja prosesseista sekä ketteristä testausmenetelmistä, tekniikoista ja työkaluista. (ISTQB 2014b) Mallipohjainen testaaja -laajennus on tarkoitettu testausalan ammattilaisille, ja sertifikaatin suoritettuaan testaaja tuntee mallipohjaisen testauksen terminologian, käsitteet, tek-

niikat ja prosessit, osaa luoda ja ylläpitää mallipohjaisessa testauksessa syntyviä testausmateriaaleja sekä osaa integroida mallipohjaisen testauksen testausprosessiin käyttämällä parhaita toimintatapoja (ISTQB 2015a: 6, 2015b).

Uusimpien ISTQB-sertifiointijärjestelmään lisättyjen laajennusten nimistä voi suoraan päätellä, että kukin sertifikaatti käsittelee tiettyä testauksen osa-aluetta tai teollisuuden alaa. Käytettävyydestaaja-laajennus perehdyttää käytettävyyden peruskäsitteisiin, käytettävyyden, käyttökokemuksen ja saavutettavuuden riskeihin, käytettävyyden ja saavutettavuusstandardeihin, soveltuvien menetelmien valitsemiseen käytettävyydestaauksessa sekä käytettävyydskatsauksen, käytettävyydestaauksen ja käyttäjätutkimuksen järjestämiseen. (ISTQB 2018b). Hyväksymistäaaja-laajennus puolestaan keskittyy käyttäjän hyväksymistästaukseen, sopimukselliseen ja lakisääteiseen hyväksymistästaukseen sekä alfa- ja betatestaukseen (ISTQB 2019c). Suorituskykytestaaja-laajennus sisältää suorituskyvyn testauksen pääpiirteet teknisestä, menetelmäpohjaisesta sekä organisaation näkökulmasta. Muita aihealueita ovat suorituskykytestauksen peruskäsitteet, suorituskyvyn mittauksen perusteet, suorituskyvyn testaus ohjelmiston elinkaaren aikana sekä suorituskykytestaustehtävät ja työkalut. (ISTQB 2018c.)

Eri teollisuudenaloihin suoraan keskittyviä sertifikaattilaajennuksia on kolme kappaletta. Autoteollisuuden ohjelmistojen testaaja -laajennus keskittyy ohjelmistojen testaukseen autoalalla, alan järjestelmien testausstandardeihin, testaukseen virtuaalisessa ympäristössä, sekä autospesifisiin staattisiin ja dynaamisiin testaustekniikoihin (ISTQB 2018d). Uhkapeeliteollisuuden testaaja -laajennus käsittelee uhkapeeliteollisuuden avainkäsitteitä, ja perehdyttää uhkapeelitalan ekosysteemeihin ja alan yhteisiin testityyppeihin (ISTQB 2019d). Mobiilisovellusten testaaja -laajennus tutustuttaa testaukseen mobiilissa ympäristössä, ja laajennus käsittelee mobiilisovellusten yleisiä testityyppejä ja testimenetelmiä, mobiilisovellusympäristöjä ja työkaluja sekä testin suorittamisen automatisointia mobiilissa ympäristössä (ISTQB 2019e).

Perustasoon pohjautuvat jatkotason sertifikaatit kattavat vaativien testaustehtävien ja testauksen kehittämisen osaamisalueet. Jatkotason sertifikaatit on suunnattu henkilöille, jotka ovat edenneet ohjelmistotestauksen työurallaan jo pidemmälle, tai jotka tarvitsevat työssään

syvällisempää ohjelmistotestauksen ymmärtämistä. (ISTQB 2012a: 6.) Jatkotasolla voi tällä hetkellä suorittaa kuusi erilaista sertifikaattia: ydinpolulla on tarjolla testauspäällikön (engl. *Test Manager*), testausasiantuntijan (engl. *Test Analyst*) ja teknisen testausasiantuntijan (engl. *Technical Test Analyst*) sertifikaatit. Ketterällä polulla on tarjolla Ketterä tekninen testaaja (engl. *Agile Technical Tester*) -sertifikaatti. Lisäksi erikoistumispolulla on mahdollisuus sertifioitua tietoturvatestaajaksi (engl. *Security Tester*) sekä testausautomaatioinsinööriksi (engl. *Test Automation Engineer*). (ISTQB 2020d.)

Jatkotason sertifikaattien sisällöt vaihtelevat sertifikaatin pääroolin mukaan. Testauspäällikkö-sertifikaatin sisältönä ovat testausprosessi, testauksen hallinta, katselmoinnit, vianhallinta, testausprosessien kehittäminen, testaustyökalut ja automaatio sekä vuorovaikutustaidot ja tiimin kokoaminen. Sertifikaatin suoritettuaan jatkotason testauspäälliköllä odotetaan olevan riittävät taidot, jotka mahdollistavat testaajan urapolulla kehittymisen asiantuntijatasolla testauksen hallinnan ja testausprosessin kehittämisen alueilla. (ISTQB 2012b.) Testausasiantuntija-sertifikaattiin sisältyvät testausprosessi ja testauksen hallinta testausasiantuntijan vastuiden näkökulmasta katsottuna, testaustekniikat, ohjelmiston laatuominaisuuksien testaaminen, katselmoinnit, vianhallinta ja testaustyökalut (ISTQB 2012c). Tekninen testausasiantuntija -sertifikaatti eroaa edellisestä siinä, että sertifikaattisisällön pääpaino on teknisen testauksen laatuominaisuuksissa, rakenneperustaisessa testauksessa sekä testaustyökaluissa ja automaatiossa (ISTQB 2012d).

Tietoturvatestaaja-sertifikaatin sisältönä on tietoturvatestauksen perusteet, tavoitteet, strategiat ja prosessit, tietoturvatestaus ohjelmiston elinkaareissa, turvamekanismien testaus, ihmistekijät turvatestauksessa, tietoturvatestauksen läpivienti, arviointi ja raportointi sekä tietoturvatestaustyökalut ja standardit (ISTQB 2016b). Testausautomaatioinsinööri-sertifikaatti perehdyttää testausautomaation tavoitteisiin, testausautomaation käyttöönoton valmisteluun, yleiseen testausautomaatioarkkitehtuuriin, testausautomaation käyttöönottoon liittyviin riskeihin ja epävarmuustekijöihin, testausautomaation raportointiin ja metriikkaan, manuaalisen testauksen siirtämiseen testausautomaatioympäristöön, testausautomaatioympäristön ja testien toiminnan varmistamiseen sekä jatkuvaan kehittämiseen (ISTQB 2016c). Jatkotason sertifikaateista viimeisin on ketterä tekninen testaaja -sertifikaatti, jonka sisältönä

on vaatimusten suunnittelu, testaus ketterässä ympäristössä, testiautomaatio sekä käyttöönotto ja toimitus (ISTQB 2019f: 7). Yhteistä kaikille jatkotason sertifiikaateille on se, että jatkotason sertifiikaatit tukevat ammattimaisten testaajien urapolkujen määrittelyä luomalla perustan, jonka jälkeen henkilö voi hankkia lisää taitoja ja osaamista asiantuntijatasolla.

Asiantuntijataso sertifiikaatit ovat tietyn testauksen erityisalueen osaajan eri moduuleista koostuvia sertifiikaatteja. Tällä hetkellä asiantuntijataso sertifiikaatteja on olemassa kahta tyyppiä: testausprosessin kehittämisen sertifiikaatti (engl. *Improving the Testing Process*) sekä testauksen hallinnan sertifiikaatti (engl. *Test Management*). (ISTQB 2020d.) Testausprosessin kehittäminen -sertifiikaatti koostuu kahdesta osakokonaisuudesta, joita ovat testausprosessin arviointi ja testausprosessin parannusten toteuttaminen (ISTQB 2011a). Testauksen hallinnan sertifiikaatti koostuu kolmesta osakokonaisuudesta, joita ovat strateginen testauksen hallinta, operatiivinen testauksen hallinta ja testaustiimin johtaminen (ISTQB 2011b). Asiantuntijataso sertifiikaatin saadakseen henkilön on läpäistävä monivalintakoe ja kirjoitettava essee, ja lisäksi hänellä tulee olla perustason ja sitä edeltävän jatkotason sertifiikaatin suoritusten ohella vaihteleva määrä työkokemusta testaustehtävistä. Asiantuntijataso sertifiikaatti on voimassa 5 vuotta. Asiantuntijataso sertifiikaatteihin on liitetty jatkuvan oppimisen ajatus. Jotta saavutettu asiantuntijasertifiikaatti pysyy voimassa, tulee henkilön suorittaa sertifiikaattikoe uudelleen hyväksytysti tai vaihtoehtoisesti osoittaa jatkuvaa osaamisensa kehittämistä keräämällä sertifiikaatin voimassaolon uusimiseen oikeuttavia pisteitä. Pisteitä voi hankkia esimerkiksi suorittamalla asiantuntevia testaustöitä, käymällä tai opettamalla aiheeseen liittyviä kursseja, osallistumalla konferensseihin tai kirjoittamalla kirjoja tai artikkeleita. (ISTQB 2020e.)

Yhteistä kaikille perus-, jatko- ja asiantuntijataso sekä perustason laajennusten ISTQB-sertifiikaateille on se, että jokaisessa sertifiikaattisisällössä on määritelty selkeästi kyseisen sertifiikaatin oppimistavoitteet sekä liiketoiminnan kannalta, että aihealueiden näkökulmasta. Liiketoiminnallisissa tavoitteissa määritellään osaaminen, joita tietyn sertifiikaatin suorittaneelta henkilöltä voidaan olettaa. Esimerkiksi perustason sertifiikaatin suorittaneen henkilön tulisi hallita testauksen terminologia ja käsitteet sekä ymmärtää testausprosessin, testauksen hallinnan sekä testauksen lähestymistapojen peruseriaatteet testauksen tavoitteiden tukemiseksi; osata priorisoida toiminnalliset ja ei-toiminnalliset testit; osata suorittaa sovittujen

testivaatimusten mukaiset testit sekä analysoida ja raportoida testauksen tulokset; kirjoittaa selkeitä virheraportteja; osallistua katselmointeihin sekä testaustyökalujen valintaan ja käyttöönottoon (ISTQB 2019b). Aihealuekohtaiset oppimistavoitteet on puolestaan luokiteltu Bloomin taksonomian mukaan kuuteen eri tasoon (ISTQB 2011a: 10–11). Perus- ja jatkotasolla oppimistavoitteet on jaettu neljään tasoon seuraavasti:

1. K1: muistaa, tunnistaa, palauttaa mieleen
2. K2: ymmärtää, selittää, perustella, verrata, luokitella, ryhmitellä, antaa esimerkki, tehdä yhteenveto
3. K3: soveltaa, käyttää
4. K4: analysoida

Asiantuntijatasolla on edellisten lisäksi vielä kaksi uutta tavoitetasoa:

5. K5: arvioida
6. K6: luoda uutta

Taulukossa 1 on kuvattu kunkin ISTQB-sertifikaatin osalta kyseisen sertifikaatin sisällön opettamiseen suositeltu kuluva minimiaika päivinä, yhden päivän vastatessa 7 tunnin opetusta. ISTQB-sertifikaattien sisältöjen opetukseen kuluu minimissään 2–10 päivää sertifikaatista riippuen. Vertailun vuoksi taulukossa esitetään kunkin sertifikaattisisällön opetuksen kesto päivinä ja tunteina. Suositellut opetusajat on poimittu ISTQB:n sertifikaattisisällöistä tai niitä esittelevistä dokumenteista ja pyöristetty lähimpään tuntiin (ISTQB 2019c: 4–5, 2019d: 4–5, 2019e: 4–5, 2019f: 7, 2018a: 7, 2018b: 4–5, 2018c: 4–5, 2018d: 4–5, 2016a: 6, 2016c: 4–5, 2015a: 6, 2014a: 8, 2012a: 7, 2011a, 2011b). Perustason sertifikaattisisällön opettamiseen menee keskimäärin 13 tuntia, jatkotason sertifikaattisisällön opettamiseen keskimäärin 22 tuntia ja asiantuntijatasoisen sertifikaattisisällön opettamiseen keskimäärin 67 tuntia. Tässä on huomioitava, että mukana on vain opetukseen käytettävä aika, joka ei sisällä aikaa, joka opiskelijalta kuluu harjoitustehtävien ja harjoituskokeiden tekoon, sertifikaattisisällön itsenäiseen läpikäymiseen tai sertifiointikokeeseen valmistautumiseen.

Sertifikaatti	Taso	Sertifikaattisisällön opetuksen kesto päivinä (tunteina)
Hyväksymistestaaja	Perustaso	2 päivää (10 h)
Autoteollisuuden ohjelmistojen testaaja	Perustaso	2 päivää (12 h)
Ketterä testaaja	Perustaso	2 päivää (12 h)
Mallipohjainen testaaja	Perustaso	2 päivää (12 h)
Uhkapeliteollisuuden testaaja	Perustaso	2 päivää (12 h)
Mobiilisovellusten testaaja	Perustaso	2 päivää (13 h)
Käytettävyydestestaaja	Perustaso	2 päivää (14 h)
Suorituskykytestaaja	Perustaso	2 päivää (15 h)
Sertifioitu testaaja	Perustaso	3 päivää (17 h)
Ketterä tekninen testaaja	Jatkotaso	3 päivää (16 h)
Tietoturvatestaaja	Jatkotaso	3 päivää (19 h)
Testausautomaatioinsinööri	Jatkotaso	3 päivää (20 h)
Tekninen testausasiantuntija	Jatkotaso	3 päivää (21 h)
Testausasiantuntija	Jatkotaso	4 päivää (21 h)
Testauspäällikkö	Jatkotaso	5 päivää (33 h)
Testausprosessin kehittäminen	Asiantuntijataso	9 päivää (63 h)
Testauksen hallinta	Asiantuntijataso	10 päivää (70 h)

Taulukko 1. ISTQB-sertifikaattisisältöjen opetukseen käytettävä suositeltu minimiaika

ISTQB:n sertifikaattisisältö valikoitui yhdeksi tämän tutkimuksen teoriapohjaksi monesta syystä. ISTQB:n sertifikaattisisällöt ovat ilmaisia, avoimesti saatavilla olevia testausosamista kuvaavia dokumentteja. Sertifikaattisisällöt ovat kenen tahansa hyödynnettävissä esimerkiksi testauskurssimateriaalina, kunhan kirjoittajat ja ISTQB mainitaan lähteinä ja sertifikaattisisällön tekijänoikeuksien omistajina (ISTQB 2019b: 9). Lisäksi jokaisen ISTQB:n sertifikaatin oppimistavoitteet on yksityiskohtaisesti kirjattu ja sertifikaatin sisältö on selkeästi määritelty. Sertifikaattisisällöt on määritelty ohjelmistotestauksen parhaiden asiantuntijoiden kansainvälisenä yhteistyönä, joten ne edustavat laajaa näkemystä ohjelmistotestauksesta työelämän osaamisvaatimukset mukaan lukien. Sertifiointijärjestelmä on myös riippumaton tietyistä teknologioista sekä luotettava, sillä ISTQB-sertifikaattisisältöjen kehittäminen, kokeisiin valmentava koulutus ja varsinainen sertifiointi eli kokeiden järjestäminen on

selkeästi erotettu toisistaan. (ISTQB 2020d.) Lisäksi ISTQB:n sertifikaatteja pyritään kehittämään jatkuvasti ohjelmistoalan muutosten myötä esimerkiksi erilaisten ohjelmistoalan trendejä arvioivien kyselytutkimusten avulla (ISTQB 2017). Sertifikaatit tukevat myös ammattimaisten testaajien urapolkujen määrittelyä, ja työnantajien arvostama sertifiointitodistus on hyvä apu osaamisen osoittamisena ja tunnustamisena. Lisäksi perustason sertifikaatti, perustason ketterä laajennus sekä kolme jatkotason sertifikaattisisältöä on suomennettu, joten ne ovat hyvä lisä muuten vähäiseen suomenkieliseen kirjallisuuteen ohjelmistotestauksen saralla. ISTQB sertifikaattisisällöt soveltuvat myös itsenäiseen etäopiskeluun, sillä materiaalit ovat saatavilla pdf-muodossa verkossa.

2.3 Testaus ohjelmistotekniikan opetussuosituksissa

Tietoteknisellä alalla kansainvälisessä korkeakoulujen opetussuunnitelmien kehittämissä uranuurtajia ovat tiedeyhteisöt ACM ja IEEE. Association for Computing Machinery (jatkossa ACM) on maailmanlaajuinen tieteellinen yhteisö, jonka tarkoituksena on edistää tietotekniikan tutkimusta sekä -opetusta. ACM julkaisi ensimmäiset tietojenkäsittelytieteen opetussuosituksensa vuonna 1968. Institute of Electrical and Electronics Engineers (jatkossa IEEE) on kansainvälinen tekniikan järjestö, jonka tehtävänä on muun muassa koulutuksen edistäminen sekä monien alojen keskeisten standardien määrittely. Sen alaisuudessa toimiva IEEE Computer Society (jatkossa IEEE-CS) on asiantuntijaryhmä, jonka tarkoituksena on edistää tietojenkäsittelytieteen ja -teknologian teoriaa, käytäntöä ja soveltamista sekä jäsenien ammattitaitoa. IEEE-CS on julkaissut opetussuosituksia vuodesta 1977 lähtien. (ACM, AIS & IEEE-CS 2005: 5.)

1990-luvun alussa ACM ja IEEE yhdistivät voimansa muiden asiantuntijaryhmittymien kanssa ns. Joint Task Force -työryhmäksi, jonka tehtävänä oli uudistaa tietoteknisen alan opetussuosituksia. Työryhmän työn tuloksena syntyi yhteinen tietojenkäsittelyn opetussuositus vuonna 2001, josta sittemmin kehitettiin 4 muuta opetussuositusta eri tieteenaloille. Tällä hetkellä ACM/IEEE tarjoaa sarjan sisällöltään erilaisia opetussuosituksia (engl. *Computing Curricula Series*), joita ovat tietokonetekniikan (engl. *Computer Engineering*), tietojenkäsittelytieteen (engl. *Computer Science*), tietojärjestelmätieteen (engl. *Information Sys-*

tems), tietotekniikan (engl. *Information Technology*) sekä ohjelmistotekniikan (engl. *Software Engineering*) opetussuositukset. (ACM ym. 2005: 5.) ACM/IEEE:n opetussuosituksia kehitetään edelleen, koska tietotekniikan ala on nopeasti kehittyvää, ja yhteiskunnan ja työelämän tarpeet muuttuvat jatkuvasti. Uusimmat ammattikorkeakouluille ja yliopistoille suunnatut kansainväliset opetussuositukset on laadittu 2000-luvun aikana (ACM 2018).

2.3.1 Alemman korkeakoulututkinnon opetussuositukset

ACM/IEEE:n opetussuositukset on suunnattu pääosin alemman korkeakoulututkinnon suunnitteluun ja kehittämiseen. Kaikissa ACM/IEEE:n alemman korkeakoulututkinnon sarjan opetussuosituksissa käytetään pohjana SWEBOK:in tietoa alueita, ja siten kaikkien alojen opetussuosituksissa kuvataan tiedot ja taidot, joita valmistuvan opiskelijan tulee hallita. Nämä tiedot ja taidot kuvataan SWEBOK:in tapaan erilaisissa osaamisalueissa, mutta osaamisalueet eroavat jonkin verran SWEBOK:sta ja toisistaan opetussuosituksesta riippuen. Esimerkiksi juuri testauksen painotus vaihtelee eri opetussuosituksien välillä. Tämän tutkimuksen yhdeksi teoriataustaksi valittiin ohjelmistotekniikan alemman korkeakoulututkinnon SE2014-opetussuositus, jonka ensimmäinen versio julkaistiin vuonna 2004 ja päivitetty versio vuonna 2014. (ACM 2018; ACM & IEEE-CS 2015, 2004.) Ohjelmistotestauksen opetussuositus SE2014 valittiin tämän tutkimuksen yhdeksi teoreettiseksi viitekehikseksi, koska siinä painotetaan testausta kaikkein eniten verrattuna muihin ACM/IEEE:n opetussuosituksiin (ACM ym. 2005: 24).

SE2014-opetussuositus perustuu tietämyksen runkoon, jota dokumentissa kutsutaan nimellä SEEK (engl. *Software Engineering Education Knowledge*). SEEK edustaa kokonaisuutta, joka minimissään pitäisi kuulua jokaiseen ohjelmistotekniikan opetussuunnitelmaan. SEEK sisältää 10 erilaista ohjelmistotekniikan osaamisaluetta, jotka ovat ohjelmistojen ydinkohdat (engl. *Computing Essentials*), matemaattinen ja tekninen perusta (engl. *Mathematical and Engineering Fundamentals*), ammatilliset käytännöt (engl. *Professional Practice*), ohjelmistojen mallinnus ja analyysi (engl. *Software Modeling and Analysis*), vaatimusmäärittely ja -analyysi (engl. *Requirements Analysis and Specification*), ohjelmistojen suunnittelu (engl.

Software Design), ohjelmistojen verifiointi ja validointi (engl. *Software Verification and Validation*), ohjelmistojen kehityskaari (engl. *Software Process*), ohjelmistojen laatu (engl. *Software Quality*) sekä ohjelmistoturvallisuus (engl. *Security*) (ACM & IEEE-CS 2015).

SE2014-opetussuosituksessa testaukseen läheisesti liittyviä aiheita löytyy useammasta osaamisalueesta. Esimerkiksi Ohjelmistojen kehityskaari -osaamisalueesta löytyy riskien hallintaan, metriikoihin, testausautomaatioon ja jatkuvaan integraatioon liittyvät aiheet. Vaatimusmäärittely ja analyysi -osaamisalueesta puolestaan löytyy esimerkiksi testattavuuteen, jäljitettävyyteen ja hyväksymistestien suunnitteluun liittyvät aiheet. Varsinainen ohjelmistotestaus kuuluu Ohjelmistojen verifiointi ja validointi -osaamisalueeseen, joka sisältää erilaisia tekniikoita sen varmistamiseksi, että ohjelmistokomponentti tai koko järjestelmä täyttää vaatimukset ja sidosjärjestelmien odotukset. (ACM & IEEE-CS 2015.)

SE2014-opetussuosituksessa tietämyksen rungon eri osaamisalueet (engl. *knowledge areas*) jaetaan edelleen pienempiin yksiköihin eli aihealueisiin (engl. *units*), jotka sisältävät kunkin osaamisalueen keskeisimmät aiheisällöt eli aiheet (engl. *topics*). Ohjelmistojen verifiointi ja validointi jaetaan neljään aihealueeseen, joita ovat Osaamisalueeseen liittyvä terminologia ja perusteet (engl. *V & V terminology and foundations*), Katselmoinnit ja staattinen analyysi (engl. *Reviews and static analysis*), Testaus (engl. *Testing*) sekä Poikkeamien analyysi ja raportointi (engl. *Problem analysis and reporting*). Nämä aihealueet jaetaan edelleen 26 aiheeseen, jotka on esitetty Kuviossa 2.

Reference		k,c,a	E,D	Hours
VAV	Software verification and validation			37
VAV.fnd	V&V terminology and foundations			5
VAV.fnd.1	V&V objectives and constraints	k	E	
VAV.fnd.2	Planning the V&V effort	k	E	
VAV.fnd.3	Documenting V&V strategy, including tests and other artifacts	a	E	
VAV.fnd.4	Metrics and measurement (e.g., reliability, usability, and performance)	k	E	
VAV.fnd.5	V&V involvement at different points in the life cycle	k	E	
VAV.rev	Reviews and static analysis			9
VAV.rev.1	Personal reviews (design, code, etc.)	a	E	
VAV.rev.2	Peer reviews (inspections, walkthroughs, etc.)	a	E	
VAV.rev.3	Static analysis (common defect detection, checking against formal specifications, etc.)	a	E	
VAV.tst	Testing			18
VAV.tst.1	Unit testing and test-driven development	a	E	
VAV.tst.2	Exception handling (testing edge cases and boundary conditions)	a	E	
VAV.tst.3	Coverage analysis and structure-based testing	a	E	
VAV.tst.4	Black-box functional testing techniques	a	E	
VAV.tst.5	Integration testing	c	E	
VAV.tst.6	Developing test cases based on use cases and/or user stories	a	E	
VAV.tst.7	Testing based on operational profiles (e.g., most-used operations first)	k	E	
VAV.tst.8	System and acceptance testing	a	E	
VAV.tst.9	Testing across quality attributes (e.g., usability, security, compatibility, and accessibility)	a	E	
VAV.tst.10	Regression testing	c	E	
VAV.tst.11	Testing tools and automation	a	E	
VAV.tst.12	User interface testing	k	E	
VAV.tst.13	Usability testing	a	E	
VAV.tst.14	Performance testing	k	E	
VAV.par	Problem analysis and reporting			5
VAV.par.1	Analyzing failure reports	c	E	
VAV.par.2	Debugging and fault isolation techniques	a	E	
VAV.par.3	Defect analysis (e.g., identifying product or process root cause for critical defect injection or late detection)	k	E	
VAV.par.4	Problem tracking	c	E	

Kuvio 2. Ohjelmistojen verifiointi ja validointi SE2014-opetussuosituksessa (ACM & IEEE-CS 2015)

Kuviossa 2 esitetystä Osaamisalueeseen liittyvä terminologia ja perusteet -aihealueessa aiheita ovat ohjelmistojen verifiointin ja validoinnin tavoitteet, ohjelmiston verifiointin ja validoinnin suunnittelu ja dokumentointi, metriikat ja mittaaminen sekä ohjelmistojen verifiointi ja validointi ohjelmistokehityksen eri vaiheissa. Katselmoinnit, vertaisarvioinnit ja staattinen analyysi kuuluvat puolestaan Katselmoinnit ja staattinen analyysi -aihealueeseen.

Poikkeamien analyysi ja raportointi -aihealueen sisältöjä ovat vikaraporttien analysointi, debuggaus ja virheiden eristämisen tekniikat, virheiden analyysi ja juurisyiden selvittäminen sekä virheiden seuranta.

Ohjelmistojen verifioinnin ja validoinnin osaamisalueen suurin aihealue on testaus. Kuviossa 2 esitetty Testauksen aihealue sisältää seuraavat 14 aihetta: yksikkötestaus ja testivetoisen kehitys, poikkeusten käsittely, kattavuusanalyysi ja rakenteeseen perustuvat eli lasilaa-tikkotestaustekniikat, toiminnalliset mustalaatikkotestaustekniikat, integrointitestausta, testitapausten suunnittelu käyttäjätarinoiden pohjalta, käyttöprofiilin (mukainen) testaus, järjestelmä- ja hyväksymistestaus, laatuominaisuuksien testaaminen, regressiotestaus, testausyökalut ja testausautomaatio, käyttöliittymän testaus, käytettävyydestä sekä suorituskykytestaus.

Tämän tutkimuksen aineiston analyysia varten SE2014-opetussuosituksesta haluttiin tarkastella myös Ohjelmistoturvallisuuden ja Ohjelmistojen laadun osaamisalueita. Näiden osaamisalueiden sisältö on esitetty Kuvioissa 3 ja 4. Molemmat osaamisalueet liittyvät ohjelmistotestaukseen, sillä ohjelmistotestauksella pyritään parantamaan ohjelmistojen laatua. Ohjelmistojen testattaessa on myös syytä tuntea tietoturvaan liittyvät seikat sekä osata tunnistaa ja etsiä ohjelmistoissa olevia haavoittuvuuksia ja tietoturva-aukkoja. Myös SE2014-opetussuosituksessa tuodaan esiin Ohjelmistojen laadun ja Ohjelmistoturvallisuuden osaamisalueiden aiheiden poikkeuksellinen luonne muihin osaamisalueisiin liittyvinä aiheina (ACM & IEEE-CS 2015: 26).

Reference		k,c,a	E,D	Hours
SEC	Security			20
SEC.sfd	Security fundamentals			4
SEC.sfd.1	Information assurance concepts (confidentiality, integrity, and availability)	k	E	
SEC.sfd.2	Nature of threats (e.g., natural, intentional, and accidental)	k	E	
SEC.sfd.3	Encryption, digital signatures, message authentication, and hash functions	c	E	
SEC.sfd.4	Common cryptographic protocols (applications, strengths, and weaknesses)	c	E	
SEC.sfd.5	Nontechnical security issues (e.g., social engineering)	c	E	
SEC.net	Computer and network security			8
SEC.net.1	Network security threats and attacks	k	E	
SEC.net.2	Use of cryptography for network security	k	E	
SEC.net.3	Protection and defense mechanisms and tools	c	E	
SEC.dev	Developing secure software			8
SEC.dev.1	Building security into the software development life cycle	c	E	
SEC.dev.2	Security in requirements analysis and specification	a	E	
SEC.dev.3	Secure design principles and patterns	a	E	
SEC.dev.4	Secure software construction techniques	a	E	
SEC.dev.5	Security-related verification and validation	a	E	

Kuvio 3. Ohjelmistoturvallisuus SE2014-opetussuosituksessa (ACM & IEEE-CS 2015)

Kuviossa 3 esitetty SE2014-opetussuosituksen Ohjelmistoturvallisuuden osaamisalue jaetaan kolmeen aihealueeseen, joita ovat Tietoturvan perusteet (engl. *Security fundamentals*), Laitteistojen ja verkkojen tietoturva (engl. *Computer and network security*) ja Tietoturvallisen ohjelmiston kehittäminen (engl. *Developing secure software*). Vaikka kaikki kolme osaluuetta ovat ohjelmistotuotannon kannalta tärkeitä, näistä testauksen kannalta merkittävin on viimeinen eli Tietoturvallisen ohjelmiston kehittäminen -aihealue ja siitä erityisesti Ohjelmistoturvallisuuteen liittyvä verifiointi ja validointi -aihe. SE2014-opetussuositusta ja SWE-BOK-mallin uusinta versiota keskenään verrattaessa huomataan, että Tietoturvallisen ohjelmiston kehittäminen -aihealue pohjautuu mitä ilmeisimmin SWEBOK-mallin Tietojenkäsittelyn perusta (engl. *Computing foundations*) -osaamisalueeseen, koska käsiteltävät aiheet ovat lähes samat (Bourque ym. 2014).

Reference		k,c,a	E,D	Hours
QUA	Software quality			10
QUA.cc	Software quality concepts and culture			2
QUA.cc.1	Definitions of quality	k	E	
QUA.cc.2	Society's concern for quality	k	E	
QUA.cc.3	The costs and impacts of bad quality	k	E	
QUA.cc.4	A cost of quality model	c	E	
QUA.cc.5	Quality attributes for software (e.g., dependability, usability, and safety)	k	E	
QUA.cc.6	Roles of people, processes, methods, tools, and technology	k	E	
QUA.pca	Process assurance			4
QUA.pca.1	The nature of process assurance	k	E	
QUA.pca.2	Quality planning	k	E	
QUA.pca.3	Process assurance techniques	k	E	
QUA.pda	Product assurance			4
QUA.pda.1	The nature of product assurance	k	E	
QUA.pda.2	Distinctions between assurance and V&V	k	E	
QUA.pda.3	Quality product models	k	E	
QUA.pda.4	Root cause analysis and defect prevention	c	E	
QUA.pda.5	Quality product metrics and measurement	c	E	
QUA.pda.6	Assessment of product quality attributes (e.g., usability, reliability, and availability)	c	E	

Kuvio 4. Ohjelmistojen laatu SE2014-opetussuosituksessa (ACM & IEEE-CS 2015)

Kuviossa 4 esitetty SE2014-opetussuosituksen Ohjelmistojen laadun osaamisalue jaetaan kolmeen aihealueeseen, joita ovat Ohjelmistojen laadun perusteet (engl. *Software quality concepts and culture*), Prosessien laadunvarmistus (engl. *Process assurance*) ja Tuotteiden laadunvarmistus (engl. *Product assurance*). SE2014-opetussuosituksen Ohjelmistojen laadun osaamisalue pohjautuu SWEBOK-mallin Ohjelmistojen laatu (engl. *Software Quality*) -osaamisalueeseen, koska käsiteltävät aiheet ovat lähes samat (Bourque ym. 2014).

SE2014-opetussuosituksessa osaamisalueita täydennetään kognitiivisilla taitotasoina (engl. *cognitive skill level*), jotka ovat yhteisiä kaikille ohjelmistotekniikan tietämyksen rungon osaamisalueille. Kognitiiviset taitotasot kuvataan ns. Bloomin taksonomian avulla. Bloomin taksonomiassa tiedolliset tavoitteet jaetaan yleensä kuuteen tasoon, joita ovat mieleenpalauttaminen, ymmärtäminen, soveltaminen, analysoiminen, syntetisoiminen sekä arvioiminen. (Bloom 1959). SE2014-opetussuosituksessa opiskelijalta edellytettävät taidot kuvataan seuraavilla taitotasoina (ACM & IEEE-CS 2015):

- mieleenpalauttaminen (engl. k=*knowledge*)
- ymmärtäminen (engl. c=*comprehension*)
- soveltaminen (engl. a=*application*)

Näiden lisäksi jokaisen aiheen tärkeys suhteessa koko ohjelmistotekniikan tietämyksen runkoon määritellään seuraavasti:

- Oleellinen (engl. *E=Essential*): Aihe kuuluu ohjelmistotekniikan tietämyksen runkoon
- Vaihtoehtoinen (engl. *D=Desirable*): Aihe ei kuulu ohjelmistotekniikan tietämyksen runkoon, ja se on valinnainen, mutta aihe tulee mahdollisuuksien mukaan huomioida opetussuunnitelmassa

Ohjelmistojen verifiointi ja validointi -osaamisalueen aiheiden taitotasoksi on pääosin merkitty soveltaminen. Ohjelmistoturvallisuuden osaamisalueessa muut aihealueet tulee osata joko mieleenpalauttamisen tai ymmärtämisen taitotasolla, mutta Tietoturvallisen ohjelmiston kehittäminen -aihealueeseen liittyvä osaaminen tulee olla soveltamisen taitotasoa. Ohjelmistojen laatu -osaamisalueessa aiheet on määritelty pääosin mieleenpalauttamisen taitotasolle. Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuuden sekä Ohjelmistojen laatu -osaamisalueiden kaikki aiheet on määritelty ohjelmistotekniikan ydinosaamisen aiheiksi (E). Osaamisalueissa ei ole lainkaan vaihtoehtoisia (D) opetusaiheita. (ACM & IEEE-CS 2015.)

SE2014-opetussuosituksessa opintojen mitoitus perustuu ns. lähiopetustunnin (engl. *lecture hour*) aikayksikköön. Yksi lähiopetustunti vastaa siis aikaa, joka vaaditaan aiheen sisällön opettamiseen luennolla. Lähiopetustunti ei sisällä itsenäiseen opiskeluun, tehtävien tekoon tai demoihin osallistumiseen kuluvaan aikaan. SE2014-opetussuosituksessa opiskelijan odotetaan tekevän 2–3 kertaa opetukseen kuluvaan tuntimäärää vastaavan määrän itsenäistä työtä. (ACM & IEEE-CS 2015.)

Taulukkoon 2 on koottu Kuvioissa 2, 3, ja 4 esitetty SE2014-opetussuosituksen Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuuden sekä Ohjelmistojen laatu -osaamisalueiden sisällön opettamiseen suositeltu minimiaika. Mukaan on otettu myös opiskelijalta edellytettävä itsenäinen työ, jotta SE2014-opetussuosituksen mukaisten testaukseen liittyvien osaamisalueiden sisältöjen opiskeluun kuluva vähimmäisaikaa voitaisiin arvioida.

Aihealue	Lähiopetus (h)	Opiskelijan itsenäinen työ (h)	Yhteensä (h)
Ohjelmistojen verifiointi ja validointi			
Terminologia ja perusteet	5	10–15	15–20
Katselmoinnit ja staattinen analyysi	9	18–27	27–36
Testaus	18	36–54	54–72
Poikkeamien analyysi ja raportointi	5	10–15	15–20
Yhteensä	37	74–111	111–148
Ohjelmistoturvallisuus			
Tietoturvan perusteet	4	8–12	12–16
Laitteistojen ja verkkojen tietoturva	8	16–24	24–32
Tietoturvallisen ohjelmiston kehittäminen	8	16–24	24–32
Yhteensä	20	40–60	60–80
Ohjelmistojen laatu			
Ohjelmistojen laadun perusteet	2	4–6	6–8
Prosessien laadunvarmistus	4	8–12	12–16
Tuotteiden laadunvarmistus	4	8–12	12–16
Yhteensä	10	20–30	30–40
Kaikki osaamisalueet yhteensä	67	134–201	201–268

Taulukko 2. Testaukseen sekä siihen liittyvien sisältöjen opiskeluun kuuluva minimiaika SE2014-opetussuosituksessa

Kuten Taulukosta 2 nähdään, Ohjelmistojen verifiointi ja validointi -osaamisalueeseen liittyvän terminologian ja perusteiden lähiopetukseen on suositeltu käytettävän 5 tuntia, jonka lisäksi opiskelijan tulee tehdä 10–15 tuntia itsenäistä työtä. Näin ollen osaamisalueen terminologian ja perusteiden opiskeluun kuluu minimissään 15–20 tuntia. Poikkeamien analyysi ja raportointi -aihealueen opiskeluun kuluu aikaa saman verran. Katselmoinnit ja staattinen analyysi -aihealueen opiskeluun kuluu lähiopetus ja opiskelijan itsenäinen työ mukaan lukien minimissään 27–36 tuntia. Huomattavaa on, että Ohjelmistojen verifiointi ja validointi -osaamisalueen Testaus-aihealueen opetuksen määrä on 18 tuntia, mikä kattaa noin 50 % koko osaamisalueen 37 lähiopetustunneista. Testauksen aiheiden itsenäiseen opiskeluun menee opiskelijalta minimissään 36–54 tuntia ja koko testauksen aiheiden opiskeluun minimissään siten 54–72 tuntia. Koko Ohjelmistojen verifiointi ja validointi -osaamisalueen aiheiden opiskeluun kuluu lähiopetus ja opiskelijan itsenäinen työ mukaan lukien 111–148 tuntia.

Taulukon 2 mukaan Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuuden sekä Ohjelmistojen laatu -osaamisalueiden kaikkien aiheiden opetukseen kuuluu SE2014-opetussuositusten mukaan minimissään 67 tuntia. Kun tähän lisätään opiskelijalta edellytettävä itsenäisen työn minimimäärä 134–201 tuntia, menee näiden kolmen osaamisalueen oppimiseen minimissään 201–268 tuntia. Vertailun vuoksi vastaava aika on noin 7,4–9,9 suomalaista opintopistettä. Näin ollen ohjelmistotestausta sekä siihen liittyviä ohjelmistojen laadun tai ohjelmistoturvallisuuden aihealueita tulisi SE2014-opetussuositusten mukaan opettaa karkeasti arvioiden vähintään yhteensä 8 opintopistettä, eli esimerkiksi noin yhdellä 5 opintopisteen ja yhdellä 3 opintopisteen luentokursseilla. Tätä arviota käytetään tässä tutkimuksessa vertailtaessa korkeakoulujen ohjelmistotestauksen opetuksen määrää ja laatua minimisuosituksiin.

2.3.2 Ylemmän korkeakoulututkinnon opetussuositukset

ACM/IEEE:n ylempien korkeakoulututkintojen opetussuosituksia on kehitetty ainoastaan tietojärjestelmätieteen ja ohjelmistotekniikan tieteenaloilla, ja näissä testauksen osuus vaihtelee. Tietojärjestelmätieteen ylemmän korkeakoulututkinnon MSIS2016-opetussuosituksissa testauksen osuus on varsin pieni. MSIS2016-opetussuosituksissa opetettavat aiheet kuvataan ns. kompetensseina, joita opiskelijan tulee omata valmistumisensa jälkeen. Testaus kuuluu järjestelmien kehittämisen ja käyttöönoton kompetenssiin, jonka mukaan tietojärjestelmätieteen maisteriohjelmasta valmistuneen tulee osata ohjelmistotestauksesta seuraavat asiat (ACM & AIS 2017):

- ohjelmiston toiminnallisten ja ei-toiminnallisten testien suunnitleminen ja toteutus,
- ohjelmiston yksikkö-, integraatio- ja muunnostestaus,
- ohjelmiston hyväksymistestaus sekä
- ohjelmiston asennus- ja kokoonpanotestaus.

MSIS2016-opetussuositusten heikkous on siinä, ettei MSIS2016-opetussuosituksissa oteta huomioon testauksen erikoisosa-alueita, kuten tietoturvatestausta tai testausautomaatiota, joita käsitellään esimerkiksi ISTQB:n sertifikaattisisällöissä. Nämä testauksen osa-alueet

voisi mieltää ylempään korkeakoulututkintoon sopiviksi, koska ISTQB:n sertifiointijärjestelmässä ko. aiheisiin liittyvät sertifikaatit ovat edistyneempää jatkotasoja. Testauksen kompetenssien saavuttamiseksi kuluva aika ei myöskään avata MSIS2016-opetussuosituksessa tarkemmin.

Ohjelmistotekniikan ylempään korkeakoulututkinnon GSwE2009-opetussuosituksissa kuvattu testauksen osaamisalue kattaa suurelta osin täysin samat osa-alueet, jotka on jo esitelty SWEBOK:in yhteydessä (iSSEc 2009: 40) ja jotka ovat myös pohjana alempien korkeakoulututkintojen opetussuosituksissa. GSwE2009-opetussuositukset perustuvat kuitenkin SWEBOK-mallin vanhempaan versioon, josta puuttuu uusimpaan SWEBOK-malliin lisätty testaustyökalujen tietalue, sekä testaustekniikoiden tietalueeseen lisätyt mallipohjaiset testaustekniikat (Abran ym. 2004). Tämän takia nämä aihealueet puuttuvat myös GSwE2009-opetussuosituksista. MSIS2016-opetussuositusten heikkouksien sekä GSwE2009-opetussuosituksissa olevien päivitysten puutteiden sekä SWEBOK-mallin päällekkäisyyksien takia tässä tutkimuksessa ei erikseen käsitellä ylempien korkeakoulututkintojen opetussuosituksia ohjelmistotestauksen perusopintojen suunnittelussa.

Tässä luvussa kuvatasta teoreettisesta viitekehystä eli testauksen aihealueita ja sisältöjä kuvaavista dokumenteista SE2014-opetussuositusta käytetään tässä tutkimuksessa korkeakoulujen testausopetuksen aiheiden luokitteluun sekä testauskurssien sisältöjen laadulliseen arvioimiseen. SWEBOK-mallia sekä ISTQB:n sertifikaattisisältöä käytetään testauskurssien sisällön arvioimiseen sekä uuden ohjelmistotestauksen perusopintokokonaisuuden kehittämiseen. Seuraavissa luvuissa tarkastellaan testausta suomalaisissa opetussuunnitelmissa. Lopuksi kuvataan lyhyesti Suomessa 2010-luvulla tehtyjä ohjelmistotestauksen opetukseen liittyviä tutkimuksia.

2.4 Testaus suomalaisten korkeakoulujen opetussuunnitelmissa

Suomen korkeakoulujärjestelmä koostuu ammattikorkeakouluista ja yliopistoista. Ammattikorkeakouluissa ohjelmistotestausta opetetaan alemmissa ammattikorkeakoulututkinnoissa ammattikorkeakoulusta riippuen joko tietojenkäsittelyn, tietotekniikan tai tieto- ja viestintätekniikan koulutusohjelmissa. Näistä tietojenkäsittelyn koulutusohjelman laajuus on 210

opintopistettä ja tutkintonimikkeenä on tradenomi, AMK (engl. *Bachelor of Business Administration*); tietotekniikan tai tieto- ja viestintätekniiikan koulutusohjelmien laajuus on 240 opintopistettä, ja tutkintonimikkeenä on insinööri, AMK (engl. *Bachelor of Engineering*). Ammattikorkeakoulututkintoon sisältyy perus- ja ammattiopintoja, vapaasti valittavia opintoja, ammattitaitoa edistävää harjoittelua sekä opinnäytetyö. Ylempi ammattikorkeakoulututkinto on työelämälähtöinen toisen vaiheen korkeakoulututkinto, johon vaaditaan vähintään kolmen vuoden työkokemus alalta ammattikorkeakoulututkinnon tai muun soveltuvan korkeakoulututkinnon suorittamisen jälkeen. Ylempien ammattikorkeakoulututkintojen laajuus on teknisellä alalla 60 opintopistettä, ja tutkintoon johtava koulutus kestää päätoimisina opintoina noin 1,5 vuotta. Tutkintonimikkeenä on insinööri, ylempi AMK (engl. *Master of Engineering*). (Opetushallitus 2018a, Valtioneuvoston asetus ammattikorkeakouluista 1129/2014.)

Ohjelmistotestauksen opetus käynnistyi Suomessa yliopistoissa 1990-luvun lopussa, kun Teknillinen korkeakoulu ja Helsingin yliopisto järjestivät yhteistyössä testauskurssin (Pohjolainen 2007: 278). Ohjelmistotestausta opetetaan nykyään yliopistoissa tietotekniikan, tietojärjestelmätieteen, tietojenkäsittelyn tai tekniikan alan koulutusohjelmissa. Näillä aloilla yliopistossa suoritettava alempi korkeakoulututkinto on laajuudeltaan 180 opintopistettä, ja se voidaan suorittaa päätoimisesti opiskellen kolmessa vuodessa. Tutkintonimikkeenä on luonnontieteiden kandidaatti tai tekniikan kandidaatti (engl. *Bachelor of Science, Bachelor of Science in Technology*). Edellä mainituilla aloilla ylempi korkeakoulututkinto on laajuudeltaan 120 opintopistettä, ja se voidaan suorittaa päätoimisesti opiskellen kahdessa vuodessa. Tutkintonimikkeenä on filosofian maisteri tai diplomi-insinööri (engl. *Master of Science, Master of Science in Technology*). (Opetushallitus 2018b; Valtioneuvoston asetus yliopistojen tutkinnoista 794/2004.)

Yliopistossa suoritettaviin alempiin ja ylempiin korkeakoulututkintoihin johtavat koulutukset järjestetään pääasiassa pää- ja sivuaineopintoina tai monitieteisinä koulutusohjelmina. Tutkintoon kuuluvat opinnot jakaantuvat kieliopintoihin, perus- ja aineopintoihin sekä syventäviin opintoihin. Näistä oppiaineen tai siihen rinnastettavan kokonaisuuden perusopintojen laajuus on aina vähintään 25 opintopistettä. (Opetushallitus 2018b; Valtioneuvoston

asetus yliopistojen tutkinnoista 794/2004.) Tästä syystä tässä tutkimuksessa kehitetyn ohjelmistotestauksen perusopintojen kokonaislaajuus on vähintään 25 opintopistettä.

Suomen korkeakouluissa opintojen laajuutta ja edistymistä mitataan opintopisteillä. Opintojaksot pisteytetään niiden edellyttämän työmäärän mukaan siten, että yhden lukuvuoden opintojen suorittamiseen vaadittava keskimääräinen 1600 tunnin työmäärä vastaa 60 opintopistettä. (Valtioneuvoston asetus yliopistojen tutkinnoista 794/2004; Valtioneuvoston asetus ammattikorkeakouluista 1129/2014.) Yksi opintopiste vastaa siis noin 27 tunnin työmäärää. Opiskelijalta vaadittavaan työhön lasketaan kaikki kurssin suorittamiseen vaadittava työ, johon voi kuulua esimerkiksi luentojen seuraamista, harjoituksiin osallistumista tai itsenäistä opiskelua. Myös tässä tutkimuksessa käytetään suunniteltujen ohjelmistotestauksen kurssien laajuuden mittarina opintopisteitä.

Finnish Software Testing Boardin puheenjohtajan Kari Kakkosen mukaan yliopistojen ja ammattikorkeakoulujen tulisi lisätä testauksen osuutta kurssitarjontaansa, ja paras ohjelmistotestauksen kehittämisen vaihtoehto olisi saada ammattikorkeakouluihin ohjelmistotestauksen suuntautumisvaihtoehto tai yliopistoihin alan maisteriohjelma (Tivi 2010, Tivi 2013). Lisäksi eräs keino koko alan arvostuksen lisäämiseen voisi olla it-alan yritysten sponsoiman ohjelmistojen laadun lahjoitusprofessorin perustaminen, kuten on tehty Yhdysvalloissa ja Australiassa (Tivi 2013).

Valtakunnallisella korkeakoulujärjestelmän tasolla valtioneuvoston asetukset luovat kehykset suomalaisten korkeakoulujen opetussuunnitelmille. Tutkintoasetuksissa säädetään yleisellä tasolla ammattikorkeakoulu- ja yliopistotutkintojen rakenteet, tieteelliset tavoitteet sekä opintojen mitoitus (Valtioneuvoston asetus ammattikorkeakouluista 1129/2014; Valtioneuvoston asetus yliopistojen tutkinnoista 794/2004). Suomessa ammattikorkeakoulut ja yliopistot kuitenkin suunnittelevat kukin omat opetussuunnitelmansa itsenäisesti. Esimerkiksi yliopistoissa opetussuunnitelmat koostuvat yleensä ainelaitoksen tai laitosten laatimista koulutusohjelmista, joiden tavoitteena on tuottaa tietäntyyppistä osaamista esimerkiksi työelämän asiantuntijatehtäviin. Opetussuunnitelmien tarkistamiseen ja päivittämiseen vaikuttaa se, missä hallintoyksiköissä opetussuunnitelmat käsitellään ja kuinka niiden laati-

minen on organisoitu. Yliopistojen opetussuunnitelmien muotoutumista ohjaa lisäksi laitoksen tai yksikön perinne ja sinne kertynyt erityisosaaminen, sillä asiantuntemus on usein sidottu professuureihin. (Karjalainen, Lapinlampi, Jaakkola & Alha 2003.) Suomalaisilla ammattikorkeakouluilla ja yliopistoilla ei siten ole yhtenäistä opetussuunnitelmaa tai yleistä runkoa, joka kuvaisi kunkin alan opetuksen painopisteet ja sisällöt. Tämän tutkimuksen teoreettiseksi lähtökohdiksi on valittu kansainväliset ohjelmistotekniikan opetussuositukset sekä SWEBOK ja ISTQB:n sertifikaattisisältö, jotta Suomen korkeakoulujen ohjelmistotestauksen opetuksen mahdollisten puutteiden selvittäminen olisi mahdollista.

2.5 Ohjelmistotestauksen opetukseen liittyvät aiemmat tutkimukset Suomessa

Suomessa ohjelmistotestaukseen liittyvät väitöskirja- ja pro gradu -tutkimukset sekä muunlaiset opinnäytteet ovat tällä vuosituohannella keskittyneet etenkin yrityksissä tehtävään ohjelmistotestaukseen, kuten ohjelmistoalan yritysten testausprosesseihin ja niiden parantamiseen (Kasurinen 2011), yritysten testauskäytänteisiin (Iivonen 2015), testauksen metriikoihin (Heikkinen 2015), yritystason testauspolitiikan luomiseen (Vainio 2017) sekä pelitestaukseen (Lahti 2014). Lisäksi on tutkittu itse ohjelmistotestausta ja sen menetelmiä, kuten esimerkiksi tutkivaa testausta (Itkonen 2011), suorituskykytestausta (Ruokojä 2012), käytettävyydestausta (Koskela 2014; Riihiäho 2015) ja mallipohjaista testausta (Abbors 2015); testaustyökaluja (Järviö 2011), testausautomaatiota (Salonen 2012; Tuominen 2014; Lempiäinen 2015) sekä testausta ohjelmistotuotannon osana (Palomäki 2016). Testauksen opettamista sen sijaan on Suomessa tutkittu varsin vähän. Testauksen opettamiseen liittyvät tutkimukset ovat käsitelleet joko testauksen opetuksen laajuutta tietyissä suomalaisissa korkeakouluissa tai testauskurssien oppimateriaaleja.

Testauksen opetuksen laajuutta suomalaisissa korkeakouluissa on tutkittu osana ohjelmistotekniikan opetusta (Mynttinen 2017) sekä käytännön ohjelmointi -kurssia suunniteltaessa (Kyttälä 2011). Mynttinen (2017) selvitti pro gradu -tutkielmassaan miten hyvin ohjelmistotekniikan opetus Suomessa vastaa kansainvälisiä opetussuosituksia. Tutkimus toteutettiin vertailemalla kolmen suomalaisen korkeakoulun lukuvuoden 2016–2017 opinto-oppaiden

sisältöä opetussuosituksiin. Testauksen osalta tutkimuksessa todettiin, että tutkimuksessa tarkastellun kolmen korkeakoulun alemmissa korkeakoulututkinnoissa ohjelmiston laadunvarmistamisen opetus jää vähälle huomiolle. Mynttisen tutkimuksessa ei kuitenkaan tutkittu muiden suomalaisten korkeakoulujen testausopetusta.

Kyttälä (2011) selvitti diplomityössään opintojensa alussa olevien opiskelijoiden vaikeimpina pitämiä käytännön ohjelmoinnin aihealueita. Diplomityönsä teoriaosuudessa Kyttälä kartoitti käytännön ohjelmointikurssien ja ohjelmistotestauksen opetusta Suomen teknisen alan yliopistoissa. Kartoituksen perusteella ilmeni, että vain kahdessa yliopistossa kolmestatoista opetettiin ohjelmistotestausta kandidaatin tutkintoon kuuluvilla kursseilla lukuvuonna 2010–2011. Erillisiä ohjelmistojen testaukseen liittyviä kursseja puolestaan järjestettiin kyseisenä lukuvuonna pääsääntöisesti kaikissa maasteritason Ohjelmistotekniikan tai diplomi-insinöörin tutkinnon mukaisissa opinnoissa kahta yliopistoa lukuun ottamatta. (Kyttälä 2011.) Kyttälän tutkimuksen teoriaosassa tarkasteltiin kuitenkin vain yliopistojen testauskurssien tarjontaa lukumäärinä, mutta ei tarkasteltu tai vertailtu kurssien sisältöjä. Lisäksi ammattikorkeakoulujen testausopetusta ei kartoitettu millään tavalla. Mynttisen (2017) ja Kyttälän (2011) tutkimusten tulokset ovat kuitenkin yhteneväiset sen suhteen, että suomalaisissa korkeakouluissa erityisesti kandidaatin tason tutkinto-ohjelmien testausopetuksessa on ollut puutteita tällä vuosituohannella.

Testauksen oppimateriaaleja tai työkaluja on arvioitu tai tuotettu viidessä eri tutkimuksessa. Kasurinen (2015) arvioi erilaisia työkaluja ja lähestymistapoja, joita voi hyödyntää ohjelmistotestauksen verkko-opetuksessa. Tutkimuksessa kävi ilmi, että autenttisten työelämätaustan käyttäminen ohjelmistotestauksen verkko-opetuksessa on yksi tekijä verkossa tapahtuvan oppimisen onnistumisessa ja samalla se pitää yllä opiskelijan motivaatiota. Kasurisen mukaan ohjelmistotestausta voidaan opettaa itseopiskelumuotoisena, ja perinteiset luennot voidaan korvata kokonaan tarkoitukseen rakennetulla verkkovideoarkistolla, kunhan opiskelijoilla on mahdollisuus lähiopetukseen ja -ohjaukseen. (Kasurinen 2015.)

Korjus (2015) selvitti ja määritteli diplomityössään ohjelmistotestauksen perusteiden aiheisällön, ja laati määrittelyn pohjalta suomenkielisen oppimateriaalin Lappeenrannan tek-

nillisen yliopiston järjestämälle Ohjelmistotestauksen perusteet -opintojaksolle. Ennen Korjuksen diplomityötä ohjelmistotestaukseen keskittyvää erillistä opintojaksoa ei vielä löytenyt Lappeenrannan teknillisen yliopiston kurssitarjonnasta. Diplomityön tuloksena syntynyt oppimateriaali otettiin käyttöön keväällä 2015 järjestetyllä ohjelmistotestauksen perusteita käsittelevällä opintojaksolla. (Korjus 2015.) Sainio (2010) puolestaan tutki opinnäytetyössään ohjelmistotestauksen merkitystä, työmenetelmiä ja työvälineitä, ja kirjoitti tehdyn kartoituksen pohjalta testauksen perusteet -opintojaksolle suunnatun oppimateriaalin ja diasarjan luentoja varten Saimaan ammattikorkeakoulun tarpeisiin. Opinnäytetyön kirjoittamisen aikaan ohjelmistotestauksesta ei ollut tarjolla ammattikorkeakouluun soveltuvaa suomenkielistä oppikirjaa.

Lindroos (2014) rakensi opinnäytetyössään murtautumisympäristön tietoturvatestauksen opetuskäyttöön Turun ammattikorkeakoulussa. Opinnäytetyön lopputuloksena syntyi testausympäristö, johon voitiin ottaa etäyhteys ja tehdä erilaisia murtautumistestaukseen liittyviä laboratoriotehtäviä. Kaleva (2011) kehitti opinnäytetyössään virheitä sisältävän testauskohteen (tietokantapohjainen web-sovellus) käytettäväksi opetuksessa Oulun seudun ammattikorkeakoulun Liiketalouden yksikössä toteutettavalla valinnaisella Ohjelmistotestausopintojaksolla. Korjuksen (2015), Sainion (2010), Lindroosin (2014) ja Kalevan (2011) tutkimuksissa yhteistä oli se, että niissä pyrittiin kehittämään oppimateriaaleja yksittäisille testauksen kursseille. Näissä tutkimuksissa ei kuitenkaan pyritty laajentamaan ohjelmistotestauksen opetusta kansallisella tasolla.

Vaikka aiemmat Suomessa tehdyt tutkimukset ovat käsitelleet ohjelmistotestausta varsin monesta eri näkökulmasta, ei yhdessäkään ole keskitytty ohjelmistotestauksen opintokokonaisuuden tai yksittäisen ohjelmistotestauksen kurssin sisällön suunnittelemiseen. Missään aiemmassa tutkimuksessa ei myöskään ole kartoitettu ohjelmistotestauksen opetuksen laajuutta kaikissa Suomen korkeakouluissa. Tämän tutkimuksen yhtenä tavoitteena onkin paikata näitä aukkoja ohjelmistotestauksesta tehdyssä tutkimuksessa. Seuraavassa luvussa 3 kuvataan tämän tutkimuksen tavoitteet, tutkimusaineisto, tutkimusmenetelmät ja muut tutkimukseen liittyvät seikat.

3 Tutkimusmenetelmä

Tässä luvussa kuvataan tutkimuksen tarkoitus, tutkimuskysymykset, tutkimuksessa käytettävät käsitteet, tutkimuskohde sekä tutkimuksen eteneminen. Lisäksi luvussa kuvataan, miten tutkimusaineisto koottiin, ja esitellään lyhyesti tutkimusmenetelmät. Luvun loppuun kuvataan tutkimuksen rajaamiseen vaikuttaneet seikat.

3.1 Tutkimuksen tarkoitus ja tutkimuskysymykset

Tämän laadullisen tutkimuksen tarkoituksena on selvittää, mitä ohjelmistotestauksesta opetetaan suomalaisissa ammattikorkeakouluissa ja yliopistoissa. Lisäksi tutkimuksessa selvitetään, miten hyvin tai huonosti testauksen opetus vastaa kansainvälisiä opetussuosituksia, ja miten ohjelmistotestauksen korkeakouluopetusta voitaisiin kehittää opetuksen sisältöjen näkökulmasta.

Tutkimuksessa tehtävän ohjelmistotestauksen opetuksen selvitystyön tavoitteena on kartoittaa ohjelmistotestauksen opetuksessa olevia mahdollisia puutteita. Tehdyn selvityksen sekä lähdekirjallisuuden pohjalta luodaan kokonaan uusi testausalalle aikoville opiskelijoille tarkoitettu ohjelmistotestauksen perusopintokokonaisuus. Perusopintokokonaisuuteen suunnitellaan opintokokonaisuuden sisältö, osaamistavoitteet sekä opintokokonaisuuden sisältämien kurssien kurssikohtainen sisältö ja osaamistavoitteet.

Tutkimuksessa pyritään vastaamaan seuraaviin tutkimuskysymyksiin:

1. Mitä ohjelmistotestauksesta opetetaan suomalaisissa korkeakouluissa?
2. Miten ohjelmistotestauksen opetus Suomessa vastaa kansainvälisiä opetussuosituksia?
3. Miten ohjelmistotestauksen korkeakouluopetuksen sisältöä voitaisiin kehittää?

3.2 Tutkimuksen käsitteet, tutkimuskohde ja tutkimuksen eteneminen

Ohjelmistotestauksen käsite voidaan määritellä eri tavoin. Suppeassa merkityksessä ohjelmistotestaus nähdään useimmiten prosessina, jonka tavoitteena on löytää virheitä järjestelmästä (Myers, Badgett & Sandler, 2012). Tässä tutkimuksessa ohjelmistotestaus ymmärretään suhteellisen laajassa merkityksessä. **Ohjelmistotestaus** (engl. *software testing*) määritellään ohjelmistojen suunnitteluun, toteutukseen ja arviointiin liittyvistä ohjelmiston elinkaaren toiminnoista muodostuvaksi prosessiksi, jonka tavoitteena on

- i. arvioida, vastaako ohjelmisto sille asetettuja vaatimuksia
- ii. osoittaa, että ohjelmiston vaatimukset sopivat suunniteltuun käyttöön
- iii. todentaa, että ohjelmisto on tietoturvallinen
- iv. löytää virheitä ohjelmistosta.

Määritelmä vastaa osittain ISTQB:n suomenkielisessä sanastossa esitettyä testauksen määritelmää (FiSTB 2015: 58), mutta tähän määritelmään mukaan on korostetusti otettu lisäksi tietoturvan käsite painottamaan sitä, että tietoturvallisuuteen liittyvien piirteiden testaus on nykypäivänä tärkeää. Tietoturvallisuudella tarkoitetaan tässä sitä, että ohjelmisto on eheä, saatavilla oleva ja luottamuksellinen, eikä sen käyttämisestä aiheudu käyttäjälle harmia tai haittaa.

Tässä tutkimuksessa **osaamistavoite** (engl. *learning outcome*) määritellään ymmärrykseksi siitä, mitä opiskelijan odotetaan osaavan suorittuaan ohjelmistotestaukseen liittyvän kurssin. Osaamistavoitteessa kuvataan testauskurssin tavoiteltu tulos, eli mitä opiskelija osaa, tietää, ymmärtää tai kykenee tekemään opittuaan kurssilla käsiteltävät testaukseen liittyvät asiat. Määritelmä vastaa W5W2-hankkeessa esitettyä osaamistavoitteen määritelmää (W5W2-hanke 2009: 4).

Tämän tutkimuksen tutkimuskohteena ovat IT-alaa opettavien ammattikorkeakoulujen ja yliopistojen lukuvuoden 2016–2017 opinto-oppaat. Tutkimus eteni kokonaisuudessaan seuraavasti: Ensin etsittiin testauksen aihealueita, oppisisältöjä ja testausosaamista esitteleviä määritteleviä dokumentteja Internetistä. Seuraavaksi tehtiin testauksen aihesisältöjä kuvaavien dokumenttien valinta, jonka jälkeen dokumentteihin tutustuttiin päällisin puolin

testausosaamisen, testauksen osa-alueiden ja testauksen erilaisten aiheisältöjen ymmärtämiseksi. Seuraavaksi rajattiin tutkimuskohteet ja koottiin tutkimusaineisto, joka luokiteltiin ja analysoitiin mahdollisten testausopetuksessa ilmenevien heikkouksien ja puutteiden selvittämiseksi. Lopuksi saatujen tulosten ja valitun lähdekirjallisuuden pohjalta annettiin ratkaisuehdotuksia testauksen opetuksen kehittämiseksi. Päällimmäisenä ratkaisuehdotuksena tässä tutkimuksessa luotiin seitsemän ohjelmistotestausta käsittelevää kurssia sisältävä ohjelmistotestauksen perusopin kokonaisuus, jonka avulla ohjelmistotestauksen opetusta voitaisiin kehittää.

3.3 Testauksen aiheisältöjä kuvaavien dokumenttien valinta

Testausosaamisen keskeisiä sisältöjä ja aihealueita määrittelevistä dokumenteista tähän tutkimukseen valittiin SWEBOK, ISTQB:n sertifikaattisisältö sekä ohjelmistotekniikan kansainväliset opetussuositukset. Nämä dokumentit valittiin, koska ne ovat Internetissä avoimesti saatavilla ja soveltuvat hyvin korkeakouluopetuksen kehittämiseen. Lisäksi kyseisissä dokumenteissa esitellään suhteellisen kattavasti testauksen eri aihealueita.

Tämän tutkimuksen yhdeksi teoreettiseksi viitekehukseksi valittiin ISTQB:n sertifikaattisisältö, mutta seuraavia kaupallisia testausertifikaatteja ei otettu selvitystyön pohjaksi: SSBB (Six Sigma Black Belt), CSTP, CASTP, CSTA, CSTL, CMSTP, CSQM, CTM (International Institute for Software Testing -organisaation sertifikaatit), CSTE, CSQA (www.softwarecertifications.org -organisaation sertifikaatit), BCS (bcs.org) sekä CQIA (asq.org). Näitä sertifikaatteja ei valittu, koska ne eivät ole ISTQB:n tavoin yleishyödyllisen yhteisön ylläpitämiä vaan kaupallisia, eikä niiden materiaali ole vapaasti saatavilla.

Tietojärjestelmätieteen ylemmän korkeakoulututkinnon MSIS2016-opetussuosituksissa testauksen osuus rajoittuu ohjelmiston tuotannon aikaiseen testaukseen, hyväksymistestaukseen sekä asennus- ja kokoonpanotestaukseen. Muissa dokumenteissa käsiteltyjä testaukseen kuuluvia aihealueita kuten testauksen hallintaa, käytettävyydestausta tai testausautomaatiota ei mainita MSIS2016-opetussuosituksessa. Ohjelmistotekniikan ylemmän korkea-

koulututkinnon GSwE2009-opetussuosituksissa kuvattu testauksen osaamisalue kattaa täysin samat osa-alueet, jotka on jo esitelty SWEBOK:in yhteydessä. Näistä syistä tässä tutkimuksessa ei käsitellä ylempien korkeakoulututkintojen opetussuosituksia.

ISO/IEC 29119 -testausstandardia ei myöskään otettu mukaan testauskurssien sisällön arvioimiseen eikä ohjelmistotestauksen perusopintokokonaisuuden kehittämiseen, koska testausstandardiin liittyy useita ongelmia. Esimerkiksi testausstandardin maksullisuus (Reid 2014) on nähty yhtenä isona ongelmana. Lisäksi testausalan ammattilaiset ovat esittäneet standardia vastaan suurta kritiikkiä (Marvin 2014; Pyhäjärvi 2014; Reid 2014) eivätkä kaikki testausalan ammattilaiset kannata tämän standardin kuvaamaa testaustoimintaa (ipetitions 2014). Lisäksi suurin osa suomalaisten yritysten testausorganisaatioista ei virallisesti käytä mitään testaukseen liittyvää standardia testaustoiminnassaan (Kasurinen 2013: 155–156). Myös erilaiset laatumallit kuten ISO/IEC 25010 sekä SWEBOK:ssa esitetty ohjelmiston laadun aihealue jätettiin tämän tutkimuksen ulkopuolelle, koska tässä tutkimuksessa haluttiin keskittyä ohjelmistotestaukseen muiden laadunvarmistamisen keinojen sijaan. Lisäksi esimerkiksi uusinta alemman korkeakoulututkinnon ohjelmistotekniikan kansainvälistä opetussuosituksia aiempaan versioon verrattaessa huomattiin, että laatumallit ja -standardit on jätetty pois ydintietämyksen rungosta uusimmassa vuoden 2014 ohjelmistotekniikan opetussuosituksen versiossa.

3.4 Tutkimusaineisto ja tutkimusmenetelmät

Tässä luvussa kuvataan tarkemmin tutkimuksessa käytetyn tutkimusaineiston kokoaminen, tutkimusaineiston luokittelu sekä analyysi. Tutkimusaineiston analyysi perustuu aineistolähtöiseen sisällönanalyysiin, jonka peruspiirteet kuvataan lyhyesti.

3.4.1 Tutkimusaineiston kokoaminen

Laadullisen tutkimuksen yleisimpiä aineistonkeruumenetelmiä ovat haastattelu, kysely, havainnointi ja erilaisista dokumenteista koottu tieto (Tuomi & Sarajärvi 2018: 83). Tämän tutkimuksen aineisto kerättiin Internetissä olevista dokumenteista, eli suomalaisten korkeakoulujen opinto-oppaista. Aineisto kerättiin toukokuussa 2017, ja aineistonkeruu toteutettiin

käymällä läpi kaikkien IT-alaa opettavien suomalaisten korkeakoulujen Internetistä löytyvät lukuvuoden 2016–2017 opinto-oppaat. Opinto-oppaista tunnistettiin potentiaalisia ohjelmistotestausta käsitteleviä kursseja. Kurssin tunnistamisessa hakusanoina käytettiin sanoja *testaus*, *testi*, *testaaminen*, *laatu*, *käytettävyys* ja *turvallisuus*, näiden sanojen alkuosia sekä sanojen ja alkuosien englanninkielisiä vastineita.

Varsinainen tutkimusaineisto muodostettiin seuraavasti. Jokaisesta hakusanoilla löydetystä kurssista tarkasteltiin ensin tarkemmin opinto-oppaassa saatavilla olevia kurssin tavoitteita ja kurssikuvausta. Mikäli opinto-oppaassa ei mainittu kurssin tarkempaa sisältöä, etsittiin kyseisen kurssin sisältökuvaus ja oppimistavoitteet korkeakoulun verkossa saatavilla olevasta kurssikirjanpitojärjestelmästä (esim. SoleOPS, Weboodi). Kurssi tulkittiin testausta käsitteleväksi kurssiksi ja otettiin mukaan aineistoon, jos kurssikuvauksessa tai kurssin oppimistavoitteissa mainittiin testaus, jokin testauksen menetelmä tai testaustyökalu. Löytyneistä kursseista osa (6 kpl) käsitteli enemmän ohjelmiston laatua kuin ohjelmistotestausta, mutta myös tällaiset kurssit otettiin mukaan aineistoon, koska pelkän kurssikuvauksen perusteella ei voitu tietää käsiteltiinkö kurssilla ohjelmistotestausta laadunvarmistamisen menetelmänä vai ei. Sisältökuvauksen perusteella myös osa ohjelmistoturvallisuuden kursseista käsitteli enemmän tietoturvallista ohjelmointia kuin tietoturvatestausta. Myös tällaiset kurssit otettiin mukaan aineistoon, koska ei voitu sanoa varmasti kuinka paljon kurssilla käsiteltiin testausta. Myös ohjelmiston haavoittuvuuksiin ja niiden havaitsemiseen tai ehkäisemiseen keskittyvät kurssit luettiin testausta käsitteleviksi kursseiksi.

Jokaisen aineistoon poimitun testausta käsittelevän kurssin tiedot kirjattiin taulukkomuotoon Excel-*taulukkolaskentadokumenttiin*. Jokaisesta kurssista kirjattiin ylös kurssin järjestäjä, nimi, laajuus, taso (alempaan vai ylempään korkeakoulututkintoon kuuluva), pakollisuus suhteessa opetusohjelmaan, vaaditut esitiedot, kurssikuvaus sekä linkki kurssikuvaukseen. Jos kurssin pystyi liittämään valinnaisena tai pakollisena tutkintoon useammassa kuin yhdessä tutkinto-ohjelmassa, kurssi tulkittiin pakolliseksi, mikäli se oli pakollinen jossakin oppilaitoksen tutkinto-ohjelmassa. Aineistoon valittuja ohjelmistotestausta käsitteleviä kursseja oli loppujen lopuksi yhteensä 62 kappaletta.

3.4.2 Tutkimusaineiston analyysi ja luokittelu

Tämän tutkimuksen aineisto analysoitiin aineistolähtöisen sisällönanalyysin keinoin. Aineistolähtöinen sisällönanalyysi on analyysimenetelmä, jossa pyritään luomaan tutkimusaineistosta teoreettinen kokonaisuus siten, että aineistosta valitaan analyysiyksiköt tutkimuksen tarkoitus ja tutkimuskysymykset huomioiden (Tuomi & Sarajärvi 2018: 108).

Aineistolähtöinen sisällönanalyysi sopii tämän tutkimuksen analyysimenetelmäksi useista syistä. Ensinnäkin aineistolähtöinen sisällönanalyysi on perusanalyysimenetelmä, jota voidaan hyödyntää kaikissa laadullisen tutkimuksen tyypeissä. Aineistolähtöisen sisällönanalyysin avulla voidaan myös analysoida dokumentteja systemaattisesti ja objektiivisesti. Lisäksi menetelmän tavoitteena on etsiä tekstistä merkityksiä, ja sillä pyritään saamaan tutkitavasta ilmiöstä kuvaus tiivistetyssä muodossa. (Tuomi & Sarajärvi 2018: 103, 117.) Tässä laadullisessa tutkimuksessa tutkimusaineistona on nimenomaan dokumentit (opinto-oppaat), ja merkityksiä etsitään opinto-oppaiden sisältävistä kurssikuvauksista.

Aineistolähtöisellä sisällönanalyysilla aineisto saadaan järjestettyä johtopäätösten tekoa varten. Aineisto voidaan järjestää esimerkiksi luokittelemalla, teemoittamalla tai tyypittelemällä. Luokittelussa aineistosta määritellään luokkia ja lasketaan, kuinka monta kertaa jokainen luokka esiintyy aineistossa. Teemoittelussa laadullinen aineisto pilkotaan ja ryhmitellään erilaisten aihepiirien mukaan. Tyypittelystä aineisto ryhmitetään tietyiksi tyypeiksi, joihin tiettyä teemaa koskevia näkemyksiä yleistetään. (Tuomi & Sarajärvi 2018: 105–107.) Tässä tutkimuksessa hyödynnetään kaikkia kolmea edellä mainittua aineiston järjestystapaa.

Tämän tutkimuksen tutkimusaineistoon valitut testauksia käsittelevät kurssit jaettiin ensin kahteen yläluokkaan järjestävän oppilaitoksen (ammattikorkeakoulu vai yliopisto) perusteella. Tämän jälkeen jokainen kurssi luokiteltiin eri teemojen mukaisiin kategorioihin, jotta voitiin selvittää, mihin testauksen opetus painottuu, ja mitä aihealueita ammattikorkeakouluissa ja yliopistoissa opetetaan vähiten. Tässä tutkimuksessa aineisto luokiteltiin siten, että aineistosta pyrittiin kokoamaan samaan luokkaan kurssit, joilla käsiteltiin samaa testauksen osa-aluetta tai aihetta, esimerkiksi käytettävyydestä. Ammattikorkeakouluissa ja yliopistoissa opettavat testauksia käsittelevät kurssit luokiteltiin aineiston analyysin aikana seuraaviin yhdeksään kategoriaan: ohjelmistotestauksen perusteet, ohjelmistoturvallisuus,

käytettävyydestä, ohjelmistojen laatu, ketterä testaus, testaustekniikat, pelitestaus, vaatimuksiin perustuva testaus sekä testaus ja ohjelmistokehitys. Näiden kategorioiden muodostus ja kurssien luokittelussa ilmenneet ongelmat kuvataan tarkemmin tulosten esittelyn yhteydessä Luvussa 4.

Luokittelun jälkeen aineistosta pyrittiin löytämään aiheita, joita testauksen kursseilla ei käsitelty, jotta testausopetuksessa ilmenevät puutteet voitiin paremmin tunnistaa. Lisäksi puutteiden esiintuomisessa hyödynnettiin valittua lähdekirjallisuutta siten, että testauskurssien sisältöjä verrattiin SE2014-opetussuositukseen, SWEBOK:in aihealueisiin sekä ISTQB:n sertifikaattisisältöön. Luokittelun lisäksi testausta käsittelevät kurssit tyypiteltiin niiden pakollisuuden tai valinnaisuuden mukaan, jotta voitiin selvittää, miten suuri osa testauksen opetuksesta on ns. pakollista ydinainesta.

3.5 Tutkimuksen rajaus

Ensimmäisenä tutkittavista kohteista rajattiin pois ne ammattikorkeakoulut ja yliopistot, joissa ei opeteta lainkaan tietojärjestelmätieteen, tietojenkäsittelytieteen tai tietotekniikan aloja. Tutkimuksen ulkopuolelle jäivät siten seuraavat 7 korkeakoulua: Diakonia-ammattikorkeakoulu, Humanistinen ammattikorkeakoulu, Poliisiammattikorkeakoulu, Saimaan ammattikorkeakoulu, Lapin yliopisto, Svenska handelshögskolan sekä Taideyliopisto.

Ohjelmistotestaukseen liittyviä kursseja järjestetään Suomessa useimmissa IT-alaa opettavissa ammattikorkeakouluissa, yliopistoissa sekä yritysten tarjoamina. Yritysten tarjoamat ohjelmistotestaukseen liittyvät kurssit ja rekrytointiohjelmat rajattiin kokonaan pois tutkimusaineistosta, koska suurin osa yritysten tarjoamista maksullisista kursseista keskittyy ISTQB-sertifikaattisisällön opettamiseen. Lisäksi yritysten tarjoamista lähinnä testausautomaatioon liittyvistä rekrytointiohjelmissa oli tutkimushetkellä saatavilla varsin vähän tietoa.

Tässä empiirisessä tutkimuksessa keskityttiin kartoittamaan pelkästään ohjelmistotestauksen opetusta. Ohjelmistotekniikan alan opetusta järjestävissä korkeakouluissa tarjotaan yleensä myös ohjelmistotuotannon perustason kursseja, joilla ohjelmistotestausta käsitellään suppeasti osana ohjelmistojen tuotantoprosessia. Esimerkiksi Jyväskylän yliopistossa järjestettiin lukuvuonna 2016–2017 viiden opintopisteen Ohjelmistotuotanto-kurssi, ja kurssilla

käytiin läpi ohjelmiston elinkaari määrittelystä käyttöönottoon ja ylläpitoon. Kyseisellä Ohjelmistotuotanto-kurssilla ohjelmistotestauksen käytänteiden ymmärtäminen on vain yksi yhdeksästä muusta ohjelmistotuotantoon liittyvästä osaamistavoitteesta. (Jyväskylän yliopisto 2016.) Tällä ohjelmistotuotannon kurssilla ja yleensäkin muiden korkeakoulujen vastaavilla kursseilla testauksen osuus on varsin pieni, ja siksi kyseisiä kursseja ei tässä tutkimuksessa luettu varsinaisiksi ohjelmistotestausta käsitteleviksi kursseiksi, eikä kyseisiä kursseja siten otettu mukaan aineistoon. Lisäksi tässä tutkimuksessa korkeakoulu tulkittiin ohjelmistotestausta opettavaksi oppilaitokseksi vain, mikäli se järjesti yhdenkin sellaisen kurssin, jolla ohjelmistotestausta käsiteltiin muusta kuin ohjelmistotuotannon näkökulmasta.

Tutkimuksessa kehitettyyn ohjelmistotestauksen perusopintokokonaisuuteen ei tämän tutkimuksen puitteissa luoda kurssien arviointikäytänteitä eikä kehitetä opintojaksoilla käytettävää oppimateriaalia. Toteutettavalle opintokokonaisuudelle ei myöskään tehdä itsearviointia eikä opintokokonaisuutta opeteta käytännössä. Tässä tutkimuksessa ei myöskään oteta kantaa siihen, miten opintokokonaisuuteen liitetyt kurssit tulisi opettaa. Nämä opintokokonaisuuden toteutukseen liittyvät asiat jätetään jatkokehitysideoiksi.

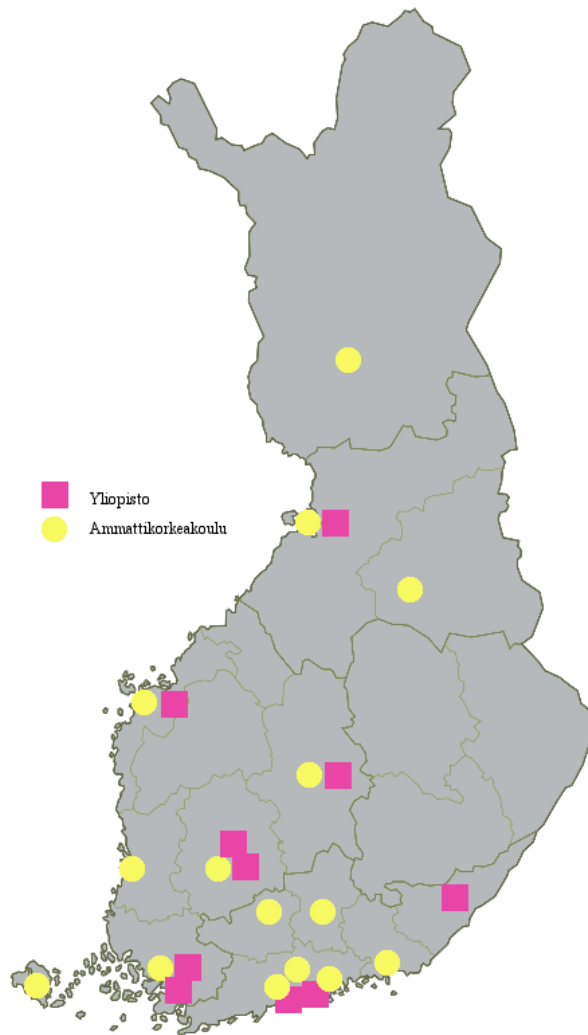
4 Tutkimuksen tulokset

Tässä luvussa kuvataan tutkimuksen tulokset. Ensin esitellään ohjelmistotestauksen opetusta Suomen korkeakouluissa lukuvuonna 2016–2017. Testausopetus kuvataan erikseen ammattikorkeakoulujen ja yliopistojen osalta. Sitten esitellään korkeakoulujen testauskursseilla käsiteltäviä aiheita, ja kuvataan joitakin testausopetuksen heikkouksia ja puutteita sekä pohditaan niiden mahdollisia syitä.

Luvun lopussa esitetään joitakin testausopetuksen kehittämideoita, sekä esitellään tämän tutkimuksen tuloksena suunniteltu ohjelmistotestauksen perusopintokokonaisuus, sen kuvaus ja osaamistavoitteet sekä opintokokonaisuuden sisältämien kurssien sisältö, osaamistavoitteet ja kurssien AHOT-esimerkit.

4.1 Ohjelmistotestauksen opetus Suomen korkeakouluissa

Tutkimuskysymysten kannalta oli tärkeää hahmottaa ensimmäiseksi korkeakoulut, joissa järjestetään ohjelmistotestaukseen liittyviä kursseja. Suomessa on 32 korkeakoulua, joissa järjestetään IT-alan koulutusta. Näistä 25 korkeakoulussa järjestettiin ohjelmistotestauksen opetusta lukuvuonna 2016–2017. Tässä tutkimuksessa tehdyn testausopetuksen kartoituksen perusteella ohjelmistotestauksen kursseja järjestetään Suomessa 15 ammattikorkeakoulussa sekä kymmenessä yliopistossa. Nämä korkeakoulut on merkattu kartalle Kuvioon 5.



Kuvio 5. Ohjelmistotestauksen opetus Suomen korkeakouluissa 2016–
2017

Suomalaisissa korkeakouluissa järjestettiin lukuvuonna 2016–2017 yhteensä 62 ohjelmistotestaukseen liittyvää kurssia 25 eri korkeakoulussa. Jokaisessa testausta opettavassa korkeakoulussa järjestettiin siis keskimäärin 2,4 testauskurssia. Jokaista IT-alaa opettavaa korkeakoulua kohti järjestettiin keskimäärin 1,9 testausta käsittelevää kurssia.

Seuraavissa luvuissa 4.1.1 ja 4.1.2 on kuvattu Kuviossa 5 esitettyjen suomalaisten ammattikorkeakoulujen ja yliopistojen testauksen opetusta lukuvuonna 2016–2017. Testauksen kurseista kaksi kurssia oli ns. yhteistyökurseja, joissa sama kurssi järjestettiin kahdessa eri korkeakoulussa (Aalto-yliopisto ja Helsingin yliopisto, Vaasan yliopisto ja Vaasan ammattikorkeakoulu).

4.1.1 Ohjelmistotestauksen kurssit ammattikorkeakouluissa

Suomen 25:sta ammattikorkeakoulusta 21 ammattikorkeakoulua järjesti tietojärjestelmätieteen, tietojenkäsittelytieteen tai tietotekniikan alaan liittyviä koulutusohjelmia lukuvuonna 2016–2017. Näistä 15 ammattikorkeakoulussa järjestettiin testausta käsitteleviä kursseja, jotka kuuluvat pakollisina tai vapaavalintaisina erilaisiin IT-alan koulutusohjelmiin. Ohjelmistotestaukseen liittyviä kursseja ei järjestetty seuraavissa kuudessa IT-alaa opettavassa ammattikorkeakoulussa: ammattikorkeakoulu Arcadassa, ammattikorkeakoulu Noviassa, Centria ammattikorkeakoulussa, Karelia-ammattikorkeakoulussa, Savonia-ammattikorkeakoulussa ja Seinäjoen ammattikorkeakoulussa.

Ensimmäisen ja toisen tutkimuskysymyksen kannalta oli oleellista selvittää, millaisia testauksen kursseja ammattikorkeakouluissa opetetaan. Alla olevassa Taulukossa 3 on kuvattu ammattikorkeakoulujen järjestämät testauksen kurssit lukuvuonna 2016–2017. Testaukseen liittyviä kursseja järjestettiin 15 eri ammattikorkeakoulussa yhteensä 36 kappaletta. Testauksen kursseja järjestettiin siten keskimäärin 2,4 kurssia jokaista ohjelmistotestausta opettavaa ammattikorkeakoulua kohti, ja keskimäärin 1,7 kurssia jokaista IT-alaa opettavaa ammattikorkeakoulua kohti. Jyväskylän ammattikorkeakoulu erottui selvimmin joukosta, sillä se tarjosi yhteensä 9 ohjelmistotestaukseen jollakin tavalla liittyvää kurssia.

Kuten Taulukosta 3 käy ilmi, suomalaisissa ammattikorkeakouluissa ohjelmistotestausta opetetaan useimmiten kolmen opintopisteen (12 kurssia) tai viiden opintopisteen (21 kurssia) laajuisella kurssilla alemmissa korkeakoulututkinnoissa. Metropolia ammattikorkeakoulu ja Jyväskylän ammattikorkeakoulu järjestivät ainoina ammattikorkeakouluina testauksen kursseja, jotka voi liittää ylempään ammattikorkeakoulututkintoon. Testauksen opetus näyttäisi siis puuttuvan lähes kokonaan ylemmistä ammattikorkeakoulututkinnoista. Syynä tähän voi olla se, että ammattikorkeakouluissa on haluttu profiloitua ylemmissä korkeakoulututkinnoissa kokonaan johonkin muuhun kuin ohjelmistotekniikkaan.

Järjestäjä	Kurssin nimi	Op	Taso	Pakollinen
Haaga-Helia	Vaatimusmääritys ja vaatimuslähtöinen testaus	5	alempi	Ei
HAMK	Testausprosessit	3	alempi	Kyllä
JAMK	Ohjelmistosuunnittelu ja testaus	5	alempi	Kyllä
JAMK	Järjestelmätestaus	4	alempi	Kyllä
JAMK	Käyttöliittymä ja käytettävyys	4	alempi	Kyllä
JAMK	Data Security Testing	3	alempi	Kyllä
JAMK	Testing	3	alempi	Ei
JAMK	Agile Testing	3	alempi	Ei
JAMK	Auditing and Testing Technical Security	5	ylempi	Kyllä
JAMK	Web Application Security	5	ylempi	Ei
JAMK	Software Exploitation	5	ylempi	Ei
KAJAK	Ohjelmistokehitysprosessi, versionhallinta ja testaus	3	alempi	Kyllä
LAMK	Software Testing and Maintenance	4	alempi	Kyllä
LAMK	Usability Engineering	3	alempi	Kyllä
Lapin AMK	Testauksen perustaidot	5	alempi	Kyllä
Laurea	Ohjelmistotestaus	5	alempi	Ei
Laurea	Käytettävyyden arviointi ja testaus	5	alempi	Ei
Metropolia	Ohjelmistojen laatu	5	alempi	Ei
Metropolia	Tietoturvallinen ohjelmistokehitys	5	alempi	Ei
Metropolia	Pelaajakeskeinen suunnittelu ja pelitestaus	5	alempi	Ei
Metropolia	Software Quality	5	ylempi	Ei
Metropolia	Fundamental Practices for Secure Software Development	5	ylempi	Ei
OAMK	Ohjelmistotestaus	3	alempi	Kyllä
OAMK	Laatujärjestelmät, testaus ja katselmointi	3	alempi	Ei
OAMK	Tietoturvaohjelmointi	3	alempi	Kyllä
OAMK	Usability Testing	3	alempi	Kyllä
SAMK	Laadunhallinta	5	alempi	Kyllä
TAMK	Ohjelmistotestaus	3	alempi	Kyllä
TAMK	Playtesting and Usability	5	alempi	Kyllä
TURKUAMK	Pelitestaus	5	alempi	Ei

TURKUAMK	Information Security Testing and Assessment	5	alempi	Ei
VAMK	Ohjelmistotestaus	5	alempi	Kyllä
XAMK	Pelitestaus	5	alempi	Kyllä
XAMK	Game Testing II	5	alempi	Kyllä
XAMK	Pentesting	5	alempi	Kyllä
ÅLAND	Testauslähtöinen ohjelmistokehitys	3	alempi	Ei

Taulukko 3. Testauksen kurssit ammattikorkeakouluissa lukuvuonna
2016–2017

Toisen tutkimuskysymyksen kannalta oli oleellista selvittää, miten hyvin tai huonosti ammattikorkeakoulujen opetus vastasi annettuja opetussuosituksia. Ammattikorkeakoulujen testausopetus ei vastannut määrällisesti kovinkaan hyvin Luvussa 2 esitettyä Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuus ja Ohjelmistojen laatu -osaamisalueita käsittelevän testausopetuksen noin 8 opintopisteen suositeltavaa minimimäärää. Vain 7 ammattikorkeakoulussa opetettiin testaukseen liittyviä aiheita yhtä paljon tai enemmän kuin 8 opintopistettä. Näistä vain seuraavat 5 ammattikorkeakoulua opetti Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuus ja Ohjelmistojen laatu -osaamisalueita: Jyväskylän ammattikorkeakoulu, Metropolia ammattikorkeakoulu, Oulun ammattikorkeakoulu, Turun ammattikorkeakoulu ja Kaakkois-Suomen ammattikorkeakoulu.

Jyväskylän ammattikorkeakoulu ja Oulun ammattikorkeakoulu ovat esimerkkejä ammattikorkeakouluista, joissa testausopetus oli määrällisesti (opintopisteissä sekä testauksen kursien lukumäärissä arvioituna) varsin hyvällä tasolla muihin ammattikorkeakouluihin sekä SE2014-opetussuositukseen verrattuna. Sen sijaan esimerkiksi Ahvenanmaan, Kajaanin ja Hämeen ammattikorkeakouluissa testauksen opetuksen määrässä oli muihin ammattikorkeakouluihin sekä SE2014-opetussuositukseen nähden selviä puutteita. Useampien testauksen kurssien puuttuminen näistä ammattikorkeakouluista voi johtua siitä, kyseiset oppilaitokset ovat halunneet profiloitua jollekin muulle ohjelmistotekniikan osa-alueelle. Toisaalta testauskurssien puuttuminen voi johtua myös siitä, että kyseisten ammattikorkeakoulujen opetussuunnitelmat pohjautuvat SE2014-opetussuosituksen sijaan joihinkin muihin ACM:n/IEEE:n opetussuosituksiin, joissa testausta painotetaan selkeästi vähemmän. Seu-

raavissa kappaleissa on kuvattu Jyväskylän ja Oulun ammattikorkeakoulun sekä Ahvenanmaan, Kajaanin ja Hämeen ammattikorkeakoulujen testausopetusta esimerkkeinä kattavasta ja puutteellisesta testausopetuksen määrästä.

Jyväskylän ammattikorkeakoulu erottui testauksen opetuksessa edukseen muiden ammattikorkeakoulujen joukosta, sillä se järjesti poikkeuksellisesti yhteensä yhdeksän ohjelmistotestaukseen liittyvää kurssia: *Ohjelmistosuunnittelu ja testaus*, *Järjestelmätestaus*, *Käyttöliittymä ja käytettävyys*, *Data Security Testing*, *Testing*, *Agile Testing*, *Auditing and Testing Technical Security*, *Web Application Security* sekä *Software Exploitation*. Näistä neljä ensimmäistä kurssia kuuluvat Tieto- ja viestintäteknikan koulutusohjelman mukaisiin opintoihin suuntautumisvaihtoehdosta riippuen. *Ohjelmistosuunnittelu ja testaus* sekä *Järjestelmätestaus* -kurssit ovat pakollisia Tieto- ja viestintäteknikan tutkinto-ohjelman Ohjelmistotekniikan suuntautumisvaihtoehdon ydinopinnoissa. Viiden opintopisteen laajuisen *Ohjelmistosuunnittelu ja testaus* -kurssin testauksen osiossa käsitellään testauksen merkitystä osana tuotekehitysprosessia, ja tuotetaan testitapauksia vaatimusmäärittelyiden pohjalta. Neljän opintopisteen laajuisen *Järjestelmätestaus*-kurssin aiheita ovat hyväksyntätestaus, järjestelmätestaus, testauksen hallinta, vianhallinta ja testausautomaatio. *Käyttöliittymä ja käytettävyys* -kurssilla opiskellaan käyttöliittymän määrittelyn ja suunnittelun perusteita, käytettävyydestestauksen menetelmiä sekä www-sivuston käytettävyydestausta. Kurssi on neljän opintopisteen laajuinen ja se kuuluu pakollisena Tieto- ja viestintäteknikan tutkinto-ohjelman Mediateknikan suuntautumisvaihtoehdon ydinopintoihin. *Data Security Testing* -kurssilla aiheena ovat tietoturvatestausten menetelmät, -prosessit ja -tekniikat, tietoturvatestaukseen liittyvä terminologia, lainsäädäntö ja määräykset, standardit ja käytännöt, haavoittuvuuskien tunnistustekniikat sekä testaustyökalut. Kurssi on viiden opintopisteen laajuinen ja se on pakollinen Tieto- ja viestintäteknikan koulutuksen Kyberturvallisuuden suuntautumisvaihtoehdossa. (Jyväskylän ammattikorkeakoulu 2016a.)

Testing-kurssin sisältönä on testauksen periaatteet, testaus ohjelmiston elinkaareissa, staattinen testaus, dynaaminen testaus, testauksen hallinta sekä testaustyökalut. *Agile Testing* -kurssilla käsitellään ketterän kehityksen perusteita, ketterän testauksen perusteita sekä ketterän testauksen työkaluja ja tekniikoita. Nämä kolmen opintopisteen laajuiset *Testing* ja *Agile Testing* -kurssit ovat valittavissa Tietojenkäsittelyn koulutusohjelman vaihtoehtoisissa

ammattiopinnoissa. *Auditing and Testing Technical Security* -kurssin aiheita ovat lainsäädännön ja määräysten rajoitukset, tietoturvatestausta ja tietoturvatestaustekniikat, haavoittuvuuksien havaitseminen sekä testausprosessi, testauksen aktiviteetit ja raportointi. Kurssin laajuus on viisi opintopistettä, ja se on pakollinen ylemmän ammattikorkeakoulututkinnon Master's Degree Programme in Information Technology -koulutusohjelman Cyber Security -pääaineessa. (Jyväskylän ammattikorkeakoulu 2016b, 2016c.)

Tutkinto-ohjelmiin kuuluvien ohjelmistotestausta käsittelevien kurssien lisäksi Jyväskylän ammattikorkeakoulu järjesti 15 opintopisteen laajuisen *Ethical hacking* -opintokokonaisuuden, jossa opiskeltiin tietojärjestelmien koventamista testaus toiminnan ja auditoinnin kautta. Opintokokonaisuus keskittyy yleisimpien haavoittuvuuksien löytämiseen sekä niiden estämiseen. Opintokokonaisuuteen kuuluu seuraavat kaksi viiden opintopisteen laajuista testaus käsittelevää kurssia: *Web Application Security* ja *Software Exploitation*. *Web Application Security* -kurssilla käsitellään yleisimpiä verkkosovellusten haavoittuvuuksia sekä niiden havaitsemista, korjaamista ja lieventämistä. *Software Exploitation* -kurssilla aiheena on yleisemmin sovellusten haavoittuvuuksien etsiminen ja niiden lieventäminen. (Jyväskylän ammattikorkeakoulu 2016a.)

Oulun ammattikorkeakoulu tarjosi neljä ohjelmistotestaukseen liittyvää kurssia: *Laatujärjestelmät, testaus ja katselmointi*, *Ohjelmistotestaus*, *Tietoturvaohjelmointi* sekä *Usability Testing*. Kukin kurssi on laajuudeltaan kolme opintopistettä. *Laatujärjestelmät, testaus ja katselmointi* -kurssilla käydään läpi laadun peruskäsitteet, laatujärjestelmä ja sen auditointi, standardit, ISO-9000, CMM-kypsyysmalli, benchmarking, testausmenetelmät sekä katselmoinnit. Kurssin voi sisällyttää vapaasti valittaviin opintoihin Tietojenkäsittelyn tutkinto-ohjelmassa Internet-palveluiden ja digitaalisen median, Järjestelmä-asiantuntemuksen sekä Web-sovelluskehityksen suuntautumisvaihtoehdoissa. *Ohjelmistotestaus*-kurssin sisältönä on testauksen rooli ohjelmistokehityksessä, manuaalinen testaus, automatisoitu testaus, testaustyökalut sekä testauksen suunnittelu, toteutus ja raportointi. Kurssi kuuluu pakollisiin ammattiopintoihin Tietojenkäsittelyn tutkinto-ohjelman Web-sovelluskehityksen suuntautumisvaihtoehdossa. Lisäksi kurssin voi liittää valinnaisiin opintoihin Degree Programme in Business Information Technology -tutkinto-ohjelmassa. *Tietoturvaohjelmointi*-kurssin ai-

heita ovat tavallisimmat haittaohjelmat, kräkkeröinti ja web-hyökkäykset, tietoturva-auditoinnit, eettinen hakkerointi, verkkoliikenteen suodatus ja salausten menetelmien perusteet. Kurssi kuuluu pakollisena Tietotekniikan tutkinto-ohjelman Ohjelmistokehityksen suuntautumisvaihtoehdon mukaisiin opintoihin. *Usability Testing* -kurssin aiheita ovat käytettävyydestä testauksen tarkoitus, käytettävyydestä suunnittelu ja toteutus sekä testitulosten raportointi, ja kurssi kuuluu pakollisiin opintoihin Degree Programme in Information Technology -tutkinto-ohjelmassa. (Oulun ammattikorkeakoulu 2016.)

Testauksen opetus ei vastannut SE2014-opetussuosituksissa mainittua noin 8 opintopisteen minimimäärää 10 ammattikorkeakoulussa. Näistä Ahvenanmaan ammattikorkeakoulussa, Kajaanin ammattikorkeakoulussa ja Hämeen ammattikorkeakoulussa testauksen opetus oli erityisen vähäistä, sillä testausta opetettiin vain yhdellä 3 opintopisteen kurssilla. Ahvenanmaan ammattikorkeakoulu järjesti ainoastaan yhden ruotsinkielisen testausta jollakin muotoa käsittelevän kurssin nimeltä *Testauslähtöinen ohjelmistokehitys*. Kurssi on laajuudeltaan kolme opintopistettä, ja sen voi sisällyttää Tietotekniikan koulutusohjelmassa vapaavalintaisiin opintoihin. Kurssin aiheita ovat testausmenetelmät, kehittämismenetelmät, ketterä kehitys, testivetoinen kehitys ja jatkuva integraatio. (Högskolan på Åland 2016.)

Kajaanin ammattikorkeakoulu tarjosi yhden ohjelmistotestaukseen liittyvän *Ohjelmistokehitysprosessi, versionhallinta ja testaus* -kurssin, joka on laajuudeltaan kolme opintopistettä. Kurssi on pakollinen Tieto- ja viestintäteknikan koulutuksessa Älykkäiden järjestelmien sekä Peliteknologian linjoilla. *Ohjelmistokehitysprosessi, versionhallinta ja testaus* -kurssin aiheita ovat ”ketterät ohjelmistonkehitysmenetelmät, vaatimusmäärittely ja testaus, dokumenttien ja koodin katselmointi, versionhallinnan tarkoitus ja perusperiaatteet, versionhallintaohjelmistot, projektin versionhallinta ja moduulien versiointi sekä projektin muutosten hallinta” (Kajaanin ammattikorkeakoulu 2016a, 2016b). Tieto- ja viestintäteknikan koulutuksen mukaiseen insinööritutkintoon sisältyy lisäksi kaksi muuta testaukseen liittyvää kurssia (*Testaustekniikka* sekä *Testaus ja testausuunnittelu*), mutta kyseisiä kursseja ei tarjottu ainakaan lukuvuonna 2016–2017, joten näitä kursseja ei siksi otettu mukaan aineistoon (Kajaanin ammattikorkeakoulu 2016a, 2016b).

Hämeen ammattikorkeakoulussa ohjelmistotestausta opetettiin Tietojenkäsittelyn koulutusohjelmassa osana pakollisen ydinosaamisen Liiketoiminta, suunnittelu ja testaus-moduulia, jossa testauksen osuus on 3 opintopistettä. Moduuliin sisältyvällä *Testausprosessit*-kurssilla testaukseen liittyviä aiheita ovat yksikkö- ja järjestelmätestaus sekä asiakastestisuunnitelman laatiminen. (Hämeen ammattikorkeakoulu 2016.)

Toisen tutkimuskysymyksen kannalta oli myös oleellista selvittää, pidettiinkö testausopetusta ammattikorkeakoulujen ydinopintoihin kuuluvana tutkinnon pakollisena osana. Kuten Luvussa 2 kävi ilmi, sekä SWEBOK:ssa että SE2014-opetussuosituksessa testaus nähdään ohjelmistokehittäjän ydintietämykseen kuuluvana osana. Ammattikorkeakouluissa 20 kurssia eli 56 % testausta käsittelevistä kursseista oli pakollisia, ja 16 kurssia eli 44 % valinnaisia. Testaukseen liittyvän kurssin pakollisuus vaihteli eri ammattikorkeakoulujen välillä: joissakin koulutusohjelmissa testaukseen liittyvät kurssit kuuluivat pakollisina tutkinto-ohjelmaan, toisissa koulutusohjelmissa niiden suorittamista suositeltiin, jolloin ne voitiin sisällyttää valinnaisina opintoina tutkintoon. Esimerkiksi Kaakkois-Suomen ammattikorkeakoulussa testaukseen liittyvät kurssit ovat pääosin pakollisia tutkintoon kuuluvia kursseja, kun taas Metropolia ammattikorkeakoulussa kaikki testaukseen liittyvät kurssit tutkintotasosta riippumatta ovat vapaaehtoisia opintoja. Vaihtelu johtuu todennäköisesti siitä, että oppilaitokset voivat itse suunnitella opetussuunnitelmansa ja valita aiheiden painotusalueet vapaasti. Koko Suomessa ammattikorkeakouluissa testauskursseja järjestettiin eniten pakollisina alempaan ammattikorkeakoulututkintoon kuuluvina opintoina. Ammattikorkeakoulujen testausopetus näyttäisi siis pakollisuuden suhteen kuitenkin melko hyvin vastaavan annettuja SE2014-opetussuosituksia.

4.1.2 Ohjelmistotestauksen kurssit yliopistoissa

Suomessa on 14 yliopistoa, joista 10 yliopistoa tarjosi testaukseen liittyviä kursseja luvulla vuonna 2016–2017. Lapin yliopistossa ja Taideyliopistossa ei opeteta ollenkaan tietojärjestelmätieteen, tietojenkäsittelytieteen tai tietotekniikan aloja, eikä kyseisissä yliopistoissa siis järjestetty minkäänlaisia testaukseen liittyviä kursseja. Hanken Svenska handelshögskolanissa tietojenkäsittelyä voi opiskella ainoastaan suppeana sivuaineena, mutta sivuainekoko-

naisuus ei sisällä testaukseen liittyviä kursseja (Hanken 2017). Itä-Suomen yliopiston Luonnontieteiden ja metsätieteiden tiedekunnan lukuvuoden 2015–2016 tutkintovaatimuksissa esiintyi tietojenkäsittelytieteen valinnaisiin syventäviin opintojaksoihin lukeutuva viiden opintopisteen laajuinen *Tietojärjestelmän testaus* -kurssi, mutta kyseinen kurssi on poistunut uusimmista tutkintovaatimuksista (Itä-Suomen yliopisto 2015, Itä-Suomen yliopisto 2016).

Ensimmäisen ja toisen tutkimuskysymyksen kannalta oli oleellista selvittää, millaisia testauksen kursseja yliopistot järjestävät. Alla olevassa Taulukossa 4 on kuvattu suomalaisten yliopistojen järjestämät testauksen kurssit lukuvuonna 2016–2017. Testaukseen liittyviä kursseja järjestettiin 10 eri yliopistossa kaiken kaikkiaan 26 kappaletta. Testauksen kursseja järjestettiin siten keskimäärin 2,3 kurssia jokaista ohjelmistotestausta opettavaa yliopistoa kohti, ja keskimäärin 2,6 kurssia jokaista IT-alaa opettavaa yliopistoa kohti. Aalto-yliopisto erottui selvimmin joukosta, sillä se tarjosi yhteensä 4 ohjelmistotestaukseen jollakin tavalla liittyvää kurssia.

Kuten taulukosta 4 käy ilmi, suomalaisissa yliopistoissa ohjelmistotestauksen opetus on keskittetty pääosin maisteritasoisille valinnaisille viiden opintopisteen kursseille. Ainoastaan Aalto-yliopiston ja Helsingin yliopiston *Software Security* -yhteistyökurssi sekä Lappeenrannan teknillisen korkeakoulun *C-ohjelmoinnin ja testauksen perusteet* -kurssin testaukseen keskittyvä osa (noin 3 op) olivat laajuudeltaan alle viiden opintopisteen kursseja. Kurssin tason suhteen poikkeuksen muodostivat Lappeenrannan teknillinen korkeakoulu sekä Vaasan yliopisto, joissa testausta käsittelevän kurssin voi liittää alempaan korkeakoulututkintoon (Vaasan yliopisto 2016a). Myös Aalto-yliopistossa yhden testausta käsittelevän kurssin (*Information Security*) voi liittää valinnaisena alempaan korkeakoulututkintoon. Koska sama kurssi on pakollinen useammassa Aalto-yliopiston maisteriohjelmassa, kyseinen kurssi tulkittiin tässä tutkimuksessa kuitenkin pelkästään ylemmän korkeakoulututkinnon tasoiseksi pakolliseksi kurssiksi.

Järjestäjä	Kurssin nimi	Op	Taso	Pakollinen
Aalto	Software Testing and Quality Assurance	5	ylempi	Ei
Aalto	Information Security	5	ylempi	Kyllä
Aalto	Usability Evaluation	5	ylempi	Kyllä

Aalto/HY	Software Security	4	ylempi	Ei
HY	Ohjelmistoprosessit ja ohjelmistojen laatu	5	ylempi	Ei
HY	Ohjelmistojen testaus	5	ylempi	Ei
HY/Aalto	Software Security	4	ylempi	Ei
JYU	Ohjelmistotestaus	5	ylempi	Ei
JYU	Ohjelmistoturvallisuus	5	ylempi	Kyllä
JYU	Järjestelmähaavoittuvuudet	5	ylempi	Ei
LUT	C-ohjelmoinnin ja testauksen periaatteet	6	alempi	Kyllä
OY	Software Quality and Testing	5	ylempi	Kyllä
OY	Emerging Trends in Software Testing	5	ylempi	Kyllä
OY	Usability Testing	5	ylempi	Ei
TTY	Ohjelmistojen testaus	5	ylempi	Kyllä
TTY	Eettinen hakkerointi	5	ylempi	Kyllä
TaY	Usability Evaluation Methods	5	ylempi	Ei
TaY	Open Source and Software Quality	5	ylempi	Ei
TaY	Testing, Security and Trust	5	ylempi	Ei
TY	Practical Agile Software Testing	5	ylempi	Kyllä
TY	Usability Testing	5	ylempi	Ei
TY	System and Application Security	5	ylempi	Kyllä
VY/VAMK	Ohjelmistotestaus	5	alempi	Ei
ÅA	Software Testing	5	ylempi	Ei
ÅA	Software Safety	5	ylempi	Ei
ÅA	Software Quality	5	ylempi	Ei

Taulukko 4. Testauksen kurssit yliopistoissa lukuvuonna 2016–2017

Toisen tutkimuskysymyksen kannalta oli oleellista selvittää, vastasiko yliopistojen testausopetus annettuja suosituksia. Taulukosta 4 nähdään, että yliopistojen testausopetus vastasi määrällisesti varsin hyvin SE2014-opetussuosituksen Ohjelmistojen verifiointi ja validointi, Ohjelmistoturvallisuus ja Ohjelmistojen laatu -osaamisalueita käsittelevää 8 opintopisteen minimimäärää. Suurimmassa osassa yliopistoja testausta, ohjelmistoturvallisuutta ja ohjelmistojen laatua käsitteleviä aiheita opetettiin vähintään 10 opintopisteen verran. Aalto-yliopisto, Oulun yliopisto ja Turun yliopisto ovat esimerkkejä yliopistoista, joissa testausopetus oli varsin hyvällä tasolla määrällisesti muihin yliopistoihin sekä SE2014-opetussuosituksen verrattuna. Testauksen opetuksen määrässä oli muihin yliopistoihin sekä SE2014-opetussuosituksen nähden selviä puutteita ainoastaan Lappeenrannan teknillisessä

yliopistossa ja Vaasan yliopistossa. Seuraavissa kappaleissa on kuvattu joidenkin Taulukossa 4 esitettyjen yliopistojen testausopetusta.

Aalto-yliopistossa ohjelmistotestausta opetetaan ainoastaan Perustieteiden korkeakoulussa maisteritasolla englanninkielisessä tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelmassa. (Aalto-yliopisto 2017a.) Testausta käsiteltiin kolmella viiden opintopisteen laajuisella kurssilla: *Software Testing and Quality Assurance*, *Usability Evaluation* sekä *Information Security*. Näistä viimeisin käy valinnaiseksi opinnoksi myös Tekniikan kandidaatin tutkintoon tietotekniikan pääaineessa. Varsinainen testauksen ja laadunvarmistuksen perusteita käsittelevä *Software Testing and Quality Assurance* -kurssi on merkattu opinto-oppaassa suositelluksi valinnaiseksi kurssiksi ainoastaan Software and Service Engineering -pääaineessa. *Usability Evaluation* sekä *Information Security* -kurseilla käsitellään ohjelmistotestausta kurssin varsinaisen aihepiirin näkökulmasta testauksen peruskurssia huomattavasti suppeammin. Molemmat kurssit ovat pakollisia Tietojenkäsittelytieteen maisteriohjelman pääaineissa, joiden ohjelmaan ne on sisällytetty. Lisäksi *Usability Evaluation* -kurssi on pakollisena Aalto-yliopiston ja EIT Digital Master Schoolin yhteisen tietotekniikan alan kaksioistutkinto-ohjelman (ICT Innovation) Human Computer Interaction and Design -pääaineen mukaisissa opinnoissa. (Aalto-yliopisto 2017b.) Näiden kurssien lisäksi Aalto-yliopisto on vuodesta 2014 lähtien järjestänyt Helsingin yliopiston kanssa yhteistyössä neljän opintopisteen laajuinen ohjelmistoturvallisuutta ja ohjelmistojen tietoturvatestausta käsittelevän *Software Security* -kurssin, joka käy valinnaiseksi kurssiksi Tieto-, tietoliikenne- ja informaatiotekniikan maisteriohjelmassa.

Oulun yliopiston Tieto- ja sähkötekniikan tiedekunta järjesti kolme maisteritasoista ohjelmistotestaukseen liittyvää kurssia: *Software Quality and Testing*, *Emerging Trends in Software Testing* ja *Usability Testing*. Kukin kurssi on viiden opintopisteen laajuinen. *Software Quality and Testing* -kurssilla aiheina ovat ohjelmistojen laatu ja laadunvarmistus, ohjelmiston laadunhallinta, metriikat, toiminnallinen testaus, yksikkö-, integraatio-, järjestelmä-, regressio- ja hyväksymistestaus, testauslähtöinen ohjelmistokehitys sekä testausautomaatio. Kurssi on pakollinen Information Processing Science -maisteriohjelman Ohjelmistotuotannon suuntautumisvaihtoehdossa sekä European Masters in Software Engineering -maiste-

riohjelmassa, jonka Oulun yliopisto järjestää yhdessä kolmen muun eurooppalaisen yliopiston kanssa. *Emerging Trends in Software Testing* -kurssilla opetetaan edistyneempiä testaus-tekniikoita (malli- ja hakupohjainen testaus, tutkiva testaus, mutaatiotestaus, kombinatorinen testaus, staattinen testaus) sekä virtualisointia ja testausautomaatiota. *Usability Testing* -kurssin sisältönä on käytettävyydestauksen perusteet ja käytettävyydestit, käytettävyydestausprosessit, käytettävyydestiskenaariot ja -tehtävät, käytettävyydestin läpivienti, testausmateriaalin analysointi sekä käytettävyydestitulosten raportointi. Molemmat kurssit ovat valinnaisia Information Processing Science -maisteriohjelman Ohjelmistotuotannon tai Tietojärjestelmien suuntautumisvaihtoehdon mukaisissa opinnoissa. *Emerging Trends in Software Testing* -kurssi on lisäksi pakollinen myös European Masters in Software Engineering -maisteriohjelmassa. (Oulun yliopisto 2016a.)

Turun yliopiston Matemaattis-luonnontieteellinen tiedekunnan tulevaisuuden teknologioiden laitos järjesti kolme testausa käsittelevää kurssia: *Practical Agile Software Testing*, *Usability Testing* sekä *System and Application Security*. Jokainen testausa käsittelevä kurssi on maisteritasoinen ja laajuudeltaan viisi opintopistettä. *Practical Agile Software Testing* -kurssi tarjoaa yleiskatsauksen ohjelmistotestaukseen erityisesti ketterän kehityksen näkökulmasta. Kurssilla käsitellään testauksen terminologiaa, tavoitteita, haasteita, käytäntöjä ja erilaisia testauksen koulukuntia. Lisäksi tarkastellaan testauksen integroimista asiakasviestintään ja kehitysohjelmaan. Kurssi on pakollinen Tietotekniikan tutkinto-ohjelmassa Ohjelmistotekniikan pääaineessa. *Usability Testing* -kurssi tarjoaa tarvittavat tiedot ja taidot suunnitella, valmistaa, suorittaa ja raportoida käytettävyydestejä laboratoriossa. Molemmat edellä mainitut kurssit voi sisällyttää valinnaisiksi opinnoiksi Tietojenkäsittelytieteiden tutkinto-ohjelmassa Vuorovaikutusmuotoilun pääaineessa. *System and Application Security* -kurssin aiheina ovat haittaohjelmat, palvelimen heikkoudet ja ohjelmistoturvallisuus sekä niiden testaaminen. Kurssi on pakollinen Master's Degree Programme in Information Security and Cryptography -maisteriohjelman mukaisissa opinnoissa. (Turun yliopisto 2016.)

Testauksen kursseja järjestävistä yliopistoista Lappeenrannan teknillisessä yliopistossa ja Vaasan yliopistossa testauksen opetus ei vastannut määrällisesti SE2014-opetussuosituksen 8 opintopisteen minimimäärää. Lappeenrannan teknillisen yliopiston LUT School of Busi-

ness and Management -yksikkö järjesti vain yhden testaukseen liittyvän kurssin, *C-ohjelmoinnin ja testauksen periaatteet*. Ohjelmistotestauksen johdanto-osuus kattaa puolet eli noin 3 opintopisteen verran kyseisestä kuuden opintopisteen laajuisesta kurssista. Testauksen osuudessa opiskellaan ohjelmistotestauksen työkaluja, tavallisimpia testausdokumenteja, yksikkö-, integrointi- ja järjestelmätestausta sekä testaamista käytännössä (LUT 2017a). *C-ohjelmoinnin ja testauksen periaatteet* -kurssi kuuluu aineopintojen pakollisena kurssina tekniikan kandidaatin tutkintoon (LUT 2017b).

Vaasan yliopiston Teknillinen tiedekunta tarjosi yhden ohjelmistotestausta käsittelevän viiden opintopisteen laajuisen *Ohjelmistotestausta*-kurssin. Kurssi toteutettiin yhteistyössä Vaasan ammattikorkeakoulun kanssa. (Vaasan yliopisto 2016a.) Kurssin voi sisällyttää valinnaisena Energia- ja informaatiotekniikan kandidaattiohjelman Informaatiotekniikan suunnan opintoihin (Vaasan yliopisto 2016b) tai diplomi-insinöörin opintojen maisteriohjelmassa Ohjelmistotekniikan linjan mukaisiin opintoihin (Vaasan yliopisto 2016c).

Toisen tutkimuskysymyksen kannalta oli myös oleellista selvittää, pidettiinkö testausopetusta yliopistoissa ydinopintoihin kuuluvana tutkinnon pakollisena osana. Yliopistoissa 10 kurssia eli 38 % kaikista testausta käsittelevistä kursseista oli pakollisia, ja 16 kurssia eli 62 % valinnaisia. Testaukseen liittyvien kurssien pakollisuus vaihtelee siis myös yliopistoissa ammattikorkeakoulujen tapaan. Esimerkiksi Helsingin yliopistossa järjestettävät testauksen kurssit olivat valinnaisia opintoja; Tampereen teknillisessä yliopistossa puolestaan kaikki testaukseen liittyvät kurssit olivat tutkintoon kuuluvia pakollisia opintoja. Ammattikorkeakouluihin verrattuna yliopistoissa testausta opetettiin enemmän valinnaisilla kursseilla. Koko Suomessa yliopistoissa testauskursseja järjestettiin pääosin ylemmissä korkeakoulututkinnoissa, ja eniten valinnaisina tutkintoon kuuluvina opintoina. SWEBOK-malliin ja SE2014-opetussuositukseen verrattuna alemmien korkeakoulututkintojen testausopetus ei vastannut suositusta siinä mielessä, että pelkästään testauksen opetusta alemmalla tasolla on liian vähän. Syynä voi olla se, että yliopistoissa käsitellään testausta yleensä kandidaattitutkintoon kuuluvien ohjelmistotuotannon kurssien osana, ja on voitu ajatella, että se on kandidaattitasolla riittävä määrä testausopetusta. Testaukseen ”erikoistuminen” maisteriopinnoissa on voitu jättää opiskelijan oman mielenkiinnon varaan.

Valtaosa kaikista alempiin korkeakoulututkintoihin sisältyvistä testauksen kursseista on pakollisia, kun taas ylempiin korkeakoulututkintoihin sisältyvä testausopetus on pääosin valinnaista. Erot johtuvat siitä, että testausta opetetaan yliopistoissa pääosin maisteriohjelmassa, kun taas ammattikorkeakouluissa testausta opetetaan lähinnä alemmissä korkeakoulututkinnoissa. Kokonaisuutena testausta opetetaan hiukan enemmän valinnaisena (52 %) kuin pakollisena (48 %). Testausta opetetaan myös hiukan enemmän (33 kurssia) alemmissä korkeakoulututkinnoissa kuin ylemmissä korkeakoulututkinnoissa (29 kurssia).

4.1.3 Testauskurssien aihealueet

Jokaisen kolmen tutkimuskysymyksen kannalta oli oleellista selvittää, mitä aihesisältöjä testausta käsittelevillä kursseilla opetetaan korkeakouluissa. Tämän tutkimuksen tutkimusaineistoon valitut testausta käsittelevät kurssit luokiteltiin ensin aineistolähtöisen sisältöanalyysin avulla kahteen pääkategoriaan (ammattikorkeakoulun järjestämä kurssi ja yliopiston järjestämä kurssi). Sitten aineiston analyysia jatkettiin aineistolähtöisen sisältöanalyysin avulla siten, että kurssit jaettiin niiden sisältökuvauksista tai osaamistavoitteista nostetun teeman mukaan yhdeksään kategoriaan, joita ovat ohjelmistotestauksen perusteet, ohjelmistoturvallisuus, käytettävyydestaus, ohjelmistojen laatu, ketterä testaus, testaustekniikat, pelitestaus, vaatimuksiin perustuva testaus sekä testaus ja ohjelmistokehitys. Luokittelu tehtiin, jotta voitiin selvittää, mihin testauksen opetus painottuu, ja mitä aihealueita suomalaisissa yliopistoissa ja ammattikorkeakouluissa opetetaan vähiten.

Eri kategorioihin luokiteltujen kurssien määrät on esitetty Taulukossa 5. Seuraavissa kappaleissa on kuvattu näiden kategorioiden muodostamisen vaiheet ja kurssien luokittelu kategorioihin. Tekstissä viitataan suluissa olevalla koodilla Luvussa 2 kuvattuihin SE2014-opetus-suosituksen osaamisalueisiin ja niiden aihealueisiin. Kurssisisällön tai osaamistavoitteiden autenttiset esimerkit annetaan neljästä suurimmasta testauskurssien kategoriasta sekä pienimmästä kategoriasta. Yksityiskohtaiset tiedot ammattikorkeakoulujen ja yliopistojen testauskurssien luokitteluista löytyvät Liitteestä A.

Testauksen aihealue	Kurseja yliopistoissa	Kurseja ammattikorkeakouluissa	Kaikki yhteensä
Ohjelmistotestauksen perusteet	8	9	17
Ohjelmistoturvallisuus	9	9	18
Käytettävyystestaus	4	4	8
Ohjelmistojen laatu	3	3	6
Ketterä testaus	1	1	2
Testaustekniikat	1	0	1
Pelitestaus	0	5	5
Vaatumuksiin perustuva testaus	0	3	3
Testaus ja ohjelmistokehitys	0	2	2
Yhteensä	26	36	62

Taulukko 5. Testauskurssien luokittelu testauksen aihealueisiin

Ensimmäisen kategoria nimettiin Ohjelmistotestauksen perusteet -kategoriaksi, ja siihen luokiteltiin kaikki kurssit, joilla käsiteltiin yleisellä tapaa ohjelmistotestauksen peruskäsitteitä kuten testaustasoja, testaustekniikoita, testauksen tavoitteita, testauksen merkitystä ohjelmistokehityksessä, testauksen välineitä, testausdokumentaatiota, virheiden raportointia tai testauksen terminologiaa. Tähän kategoriaan kuuluvat kurssit vastaavat suurin piirtein ISTQB:n perustason sertifioitu testaaja -sertifikaatin sisältöjä. Kun ensimmäisen kategorian kurssisisältöjä verrattiin SE2014-opetussuositukseen, huomattiin, että kurssit käsitelivät yleisimmin Ohjelmistojen verifiointi ja validointi -osaamisalueeseen liittyvää terminologiaa ja perusteita (VAV.fnd, tässä VAV-koodit viittaavat SE2014-opetussuosituksen osaamis- ja aihealueisiin). Tämän kategorian kurssit näyttivät käsittelevän myös osin Katselmoinnit ja staattinen analyysi -aihealuetta (VAV.rev) sekä joitakin SE2014-opetussuosituksen Testauksen aihealueeseen kuuluvia aiheita, joita on esitelty Luvussa 2 (VAV.tst.3, VAV.tst.6, VAV.tst.9, VAV.tst.10, VAV.tst.12). Kuten Taulukosta 5 voidaan nähdä, yliopistoissa ja ammattikorkeakouluissa oli yhteensä 17 kurssia, joilla opetettiin ohjelmistotestauksen perusteita. Näistä yksi kurssi oli Vaasan ammattikorkeakoulun ja Vaasan yliopiston yhteistyökurssi, joten kurssi on laskettu sekä yliopistossa että ammattikorkeakoulussa opettavaksi ohjelmistotestauksen perusteita käsitteleväksi kurssiksi. Kyseinen viiden opintopisteen laajuinen *Ohjelmistojen testaus* -yhteistyökurssi on hyvä esimerkki (1) SWEBOK-mallin mukaisesta testauksen perusteita käsittelevästä kurssista. Kurssin sisältöä ovat:

(1) Testauksen periaatteet (mm. staattinen ja dynaaminen testaus), testauksen menetelmät (lasi-, harmaa- ja mustalaatikkotestaus), testauksen tasot (yksikkö-, integraatio-, järjestelmä- ja hyväksymistestaus), testauksen merkitys ohjelmistokehityksessä, testauksen välineet, testausdokumentaatio, testauksen suunnittelu ja hallinta (Vaasan yliopisto 2016b).

Kyseinen kurssi on tosin valinnainen Vaasan yliopiston Energia- ja informaatiotekniikan kandidaatin tutkinnossa (Vaasan yliopisto 2016a), mutta kuitenkin pakollinen eli ydinopin-tojen osa Vaasan ammattikorkeakoulun Tietotekniikan koulutusohjelmassa (Vaasan ammattikorkeakoulu 2016), joten se vastaa osittain SE2014-opetussuositusten ja SWEBOK-mallin näkemystä, jossa testaus kuuluu ydinosaamiseen. Lisäksi kurssi vastaa laajuudeltaan SE2014-opetussuosituksia. Kuten esimerkiksi (2) nähdään, kurssin osaamistavoitteet ovat:

(2) Opiskelija ymmärtää testauksen peruseriaatteet ja tunnistaa testauksen roolin osana ohjelmistotuotantoprosessia. Opiskelija osaa listata testauksessa tarvittavan dokumentaation ja ymmärtää sen merkityksen onnistuneelle testaukselle. Opiskelija ymmärtää miten erilaiset testauksen työkalut tukevat testausta ja tunnistaa testausprosessin kulun. Opiskelija osaa soveltaa oppimaansa itsenäisesti toteutettavaan yksikkötestaukseen ja omaa valmiudet toimia osana testautiimiä. Opintojakso kehittää yhteistyötaitoja, ongelmanratkaisu- ja päätöksentekotaitoja sekä tuotekehityksen ja markkinoinnin tuntemista. (Vaasan yliopisto 2016a.)

Kurssiluokittelun toiseen eli Ohjelmistoturvallisuus-kategoriaan luokiteltiin sisältökuvauksiensa perusteella kaikki kurssit, joilla käsiteltiin tietoturva ja siihen liittyviä osa-alueita, tietoturvatestausta sekä ohjelmiston haavoittuvuuksia tai niiden havaitsemista. Tähän kategoriaan mukaan otettiin myös kurssit, joilla käsiteltiin tietoturva ohjelmistokehittäjän näkökulmasta, koska tällaisten kurssien sisältökuvausten perusteella ei voitu sanoa, missä määrin kurssilla käsiteltiin kuvauksessa mainittua testausta. Metropolia ammattikorkeakoulun viiden opintopisteen *Tietoturvallinen ohjelmistokehitys* -kurssi (3) ja Åbo Akademi:n viiden opintopisteen *Software Safety* -kurssi ovat hyviä esimerkkejä (4) tästä ongelmasta. Näiden kurssien sisältökuvausten perusteella kyseisellä kurssilla käsiteltävä testauksen osuus ja sen laajuus ovat epäselviä:

(3) Open Web Applications Security Project OWASP, SAMM (Software Assurance Maturity Model), ASVS (OWASP Application Security Verification Standard), OWASP TOP 10 -lista, verkkoon liittyvä turvallisuus, Apache www-palvelin ja sen asentaminen, turvallisuus ja sertifikaatit, ohjelmointikielten ominaisuudet, hyvä ohjelmointitapa sekä turvallisuuteen liittyvät asiat, tietokantaliittymät (esim. MySQL), käyttäjä-

hallinta tietokannoissa, Google Hacking - mitä kaikkea onkaan löydettävissä huonosti suojatulta ja lähtökohtaisesti virheellisesti konfiguroidulta sivustolta ja miten nämä voidaan piilottaa, **penetraatiotestaus ja testaustyökalut**, palvelinten koven-taminen (Metropolia 2016, korostus lisätty).

(4) Programming control systems at an application level, simulation of the behavior of controlled processes, simulating behavior of faulty hardware, techniques for safety analysis: FMEA, FTA, FFA, HAZOP, deriving software requirements from safety analysis, modelling requirements using use cases, state diagrams, allocating safety requirements, architecting safety critical systems, system partitioning, safety kernel, layered approach to architecting, verification: **overview of static and dynamic test-ing**, safety-critical systems development life-cycle, safety integrity levels, brief intro-duction to formal methods (Åbo Akademi 2016, korostus lisätty).

SE2014-opetussuositukseen verrattaessa huomattiin, että Ohjelmistoturvallisuus-kategori-aan luokitellut kurssit käsittelivät pääosin Ohjelmistoturvallisuuden osaamisalueen Tietotur-vallisen ohjelmiston kehittäminen -aihealuetta (SEC.dev) kokonaisuutena. Kuten Taulu-kosta 5 käy ilmi, Ohjelmistoturvallisuus-luokkaan kuului eniten kursseja. Ohjelmistoturval-lisuus-kategoriaan luokiteltiin testauskurssien sisältökuvausten perusteella yhteensä 18 kurs-sia, joista 9 kurssia opetettiin yliopistoissa ja 9 ammattikorkeakouluissa. Yliopistoissa ope-tettavista kursseista yksi kurssi oli Aalto-yliopiston ja Helsingin yliopiston yhteistyökurssi, joten kurssi on laskettu mukaan kahteen kertaan yliopistossa opetettavaksi ohjelmistoturval-lisuutta käsitteleväksi kurssiksi.

Kolmanteen Käytettävyytestaus-kategoriaan luokiteltiin kurssit, joiden aiheita olivat esi-merkiksi käytettävyyden arviointimenetelmät, käytettävyytestarkastukset, loppukäyttäjätēs-taus sekä käytettävyytestausten suorittaminen, analysointi ja raportointi. Nämä kurssit kä-sittelevät sisältökuvausten perusteella mitä ilmeisimmin SE2014-opetussuosituksessa mai-nittua Ohjelmistojen verifiointi ja validointi osaamisalueen Käytettävyytestauksen aihetta (VAV.tst.13). Kuten Taulukosta 5 käy ilmi, Käytettävyytestaus-luokkaan kuuluvia kursseja oli molemmissa korkeakoulutyypeissä neljä, yhteensä 8 kappaletta. Oulun yliopiston viiden opintopisteen *Usability Testing* -kurssi on hyvä esimerkki käytettävyytestauksen opetuk-sesta. Kuten esimerkistä (5) nähdään, kyseisen kurssin aiheita ovat käytettävyytestauksen perusteet, käytettävyytestausprosessit, käytettävyytestit ja -skenaariot, käytettävyystes-tauksen läpivienti, käytettävyytestausmateriaalin analysointi, käytettävyytestien tulosten raportointi:

(5) Basic terms and types of usability testing, usability tests process, usability test tasks and scenarios, test subjects, following through a usability test, analysing usability test material, reporting the findings from usability tests (Oulun yliopisto 2016b).

Neljännän kurssikategorian muodosti Ohjelmistojen laatu. Tähän kategoriaan luokiteltiin kurssit, joilla opetettiin kurssien sisältökuvauksen perusteella laatuun liittyviä käsitteitä testausta enemmän. Neljännän kategorian kurssit käsittelivät sisältökuvausten perusteella hyvin vaihtelevasti SE2014-opetussuosituksen Ohjelmistojen laadun (QUA) osaamisaluetta. Kuten Taulukosta 5 voidaan nähdä, sekä yliopistoissa että ammattikorkeakouluissa opetettiin yhtä montaa laatua käsittelevää kurssia. Yhteensä ohjelmiston laatuun liittyviä kursseja oli 6. Ohjelmistojen laatua käsittelevät kurssit otettiin mukaan aineistoon, sillä kyseisten kurssien sisältökuvauksen perusteella ei voitu sanoa missä määrin kurssilla käsiteltiin testausta laatuun liittyvien käsitteiden ja teorioiden ohella. Metropolian kurssi *Ohjelmistojen laatu* on hyvä esimerkki (6) tällaisesta kurssista. Kurssilla käsitellään sen sisältökuvauksen mukaan seuraavia asioita:

(6) Mitä laatu on, mitä sillä ymmärretään, virheettömyys vs. asiakkaiden odotukset, ohjelmiston laadun näkökulmat (funktionaalinen laatu, rakenteellinen laatu), ohjelmistotuotantoprosessin laatu, koodauskäytännöt, refaktorointi ja **testaus laadun varmistamisessa**, ketterät prosessit: saavutetaanko niillä entistä tyydyttävämmät järjestelmät ja käyttöliittymät? (Metropolia 2016, korostus lisätty.)

Viidennen kategorian muodosti Ketterä testaus. Tämä kategoria muodostettiin, koska ketteryys mainittiin paitsi kurssin nimessä, etenkin kurssikuvauksessa. Kuvauksen perusteella tähän kategoriaan luokitelluilla kursseilla testausta käsiteltiin erityisesti ketterästä näkökulmasta, sillä aiheita olivat ketterän kehityksen perusteet, ketterän testauksen perusteet sekä ketterän testauksen työkalut ja tekniikat. Nämä kurssit olisi voitu luokitella myös Ohjelmistotestauksen perusteet -kategoriaan, mutta ne haluttiin pitää erillään samalla tavoin kuin ISTQB:n sertifikaattisällöissä on erotettu toisistaan perustason sertifioitu testaaaja ja ketterä testaaaja -sertifikaattisällöt. SE2014-opetussuositukseen verrattaessa huomattiin, että Ohjelmistojen verifiointi ja validointi, Ohjelmistojen laatu tai Ohjelmistoturvallisuus -osaamisalueista ei suoraan löytynyt vastinetta tälle kategorialle. Syy tähän voi olla se, että ketterää ohjelmistokehitystä ei ole haluttu painottaa SWEBOK-mallissa. Kuten Taulukosta 5 voidaan

nähdä, Ketterä testaus -kategoriaan luokiteltuja kursseja oli kaksi kappaletta. On hyvä huomata, että jos ketterän testauksen kurssit olisi luokiteltu ohjelmistotestauksen perusteet -luokkaan, olisi kyseisessä luokassa ollut eniten eli 19 kappaletta testauksen kursseja. Näin ollen Ohjelmistotestauksen perusteet -kategoria olisi ollut suurin Ohjelmistoturvallisuus -kategorian sijaan.

Kuudennen kategorian muodostivat erilaisiin testaustekniikoihin syventyvät kurssit. Tähän kategoriaan luokiteltuja kursseja oli aineistossa kuitenkin vain yksi kurssi: Oulun yliopistossa opetettava 5 opintopisteen *Emerging Trends in Software Testing* -kurssi. Kyseinen kurssi olisi voitu luokitella ketterien kurssien tapaan Ohjelmistotestauksen perusteet -kategoriaan, koska testaustekniikoiden tuntemus on testaajan perusosaamista. Näin ei kuitenkaan tehty, koska kuten esimerkiksi (7) nähdään, kyseisen kurssin aiheita ovat nimenomaan kehittyneet testaustekniikat, kuten mallipohjainen testaus, hakupohjainen testaus, mutaatiotestaus, tutkiva testaus, kombinatorinen testaus, staattinen testaus sekä staattisen testauksen työkalut:

(7) Advanced testing techniques: Model-based testing, search-based testing, mutation, exploratory testing, combinatorial testing, static testing, static analyzers, test environments, virtualization, OS system containers, test automation (Oulun yliopisto 2016c).

Lisäksi kurssikuvauksessa Ohjelmistotestauksen perusteet vaadittiin kurssin esitietoina, joten tämän vaatimuksen takia kurssia ei voitu luokitella ensimmäiseen kategoriaan. Kyseistä kurssia ei voitu sen sisällön poikkeuksellisuuden vuoksi helposti luokitella mihinkään muuhunkaan kategoriaan. Tästä syystä kurssia varten luotiin oma Testaustekniikat-kategoria. SE2014-opetussuosituksessa tämän kategorian kurssi näyttäisi käsittelevän Ohjelmistojen verifiointi ja validointi -osaamisalueen Testaus-aihealueeseen sisältyvää Mustalaatikkotestaustekniikat (VAV.tst.4) -aihetta.

Pelitestaukseen liittyvät kurssit muodostivat seitsemännen kategorian. Tähän kategoriaan luokiteltiin kurssit, joilla käsiteltiin pelien testausta yleisesti sekä pelinkehityksen eri vaiheissa. Kuten Taulukosta 5 käy ilmi, pelitestausta käsitteleviä kursseja opetettiin ainoastaan ammattikorkeakouluissa. Tähän kategoriaan ei löytynyt vastinetta SE2014-opetussuosituk-

sista, koska pelitestausta ei mainita missään testausta käsittelevässä osaamisalueessa. Pelitestausta ei myöskään mainita SWEBOK-mallissa tai ISTQB:n sertifikaattisällöissä. Peli-teollisuus on kuitenkin erityisesti Suomessa merkittävä ohjelmistoalaan kuuluva toimiala, joten pelitestausta käsitteleville kursseille haluttiin tässä tutkimuksessa siksi luoda oma erityiskategoriansa. Pelitestausta testauksen erikoisosaamisalueena näyttäisi siis kuitenkin puuttuvan opetussuosituksista.

Kahdeksanteen eli Vaatimukseen perustuva testaus -kategoriaan luokiteltiin kurssit, jotka käsittelevät jollakin tapaa järjestelmän vaatimuksia ja vaatimusten testausta hyväksymistestauksen näkökulmasta. Näitä kursseja ei luokiteltu ohjelmistotestauksen perusteita käsitteleviksi kursseiksi, koska hyväksymistestaus itsessään on laaja osa-alue, joka edellyttää testauksen perusteiden hyvää hallintaa. Tähän kategoriaan liittyvät kurssit käsittelevät myös jonkin verran ohjelmiston suunnittelua, joka luonnollisesti on kytköksissä vaatimusten määrittelyyn. Tähän kategoriaan kuuluvia kursseja ei kuitenkaan jätetty pois aineistosta, koska ne eivät käsitelleet pääosin ohjelmistotuotantoakaan. Kuten Taulukosta 5 voidaan nähdä, vaatimukseen perustuvaa testausta käsiteltiin vain ammattikorkeakouluissa kolmella kursilla. Kurssikuvausten perusteella näillä kursseilla käsitellään vaihtelevasti SE2014-opetussuosituksen Ohjelmistojen verifiointi ja validointi -osaamisalueen Käyttöprofiilin mukaista testausta (VAV.tst.7) ja Järjestelmä- ja hyväksymistestausta (VAV.tst.8).

Viimeinen eli yhdeksäs kategoria nimettiin Testaus ja ohjelmistokehitys -kategoriaksi. Tähän kategoriaan luokiteltiin kurssit, joita ei sujuvasti voitu luokitella mihinkään edelliseen kategoriaan. Näillä kursseilla käsiteltiin testivetoista kehitystä tai yksikkö- ja järjestelmätestausta enemmän kuin muita testaukseen liittyviä aiheita. SE2014-opetussuosituksen verrattuna kurssien sisältö vastasi siten kutakuinkin Ohjelmistojen verifiointi ja validointi -osaamisalueen Testauksen aihealueeseen sisältyvien Yksikkötestaus ja testivetoinen kehitys (VAV.tst.1), Poikkeamien käsittely (VAV.tst.2) ja Integraatiotestaus (VAV.tst.5) -aiheiden sekä Poikkeamien analyysi ja raportointi (VAV.par) -aihealueita. Kuten Taulukosta 5 käy ilmi, Testaus ja ohjelmistokehitys -kursseja oli ainoastaan kaksi, ja kursseja opetettiin vain kahdessa eri ammattikorkeakoulussa.

Tutkimuskysymysten kannalta oli oleellista selvittää, erosiko ammattikorkeakoulujen ja yliopistojen testausopetus merkittävästi toisistaan ja SE2014-opetussuosituksista. Ammattikorkeakouluissa testauksen kurssien sisällöt vaihtelivat suuresti, sillä koulutusohjelmasta tai testauksen näkökulmasta riippuen testauskurssin sisältönä on testauksen perusteet, testauslähtöinen/tietoturvallinen ohjelmointi, vaatimuslähtöinen tai ketterä testaus, järjestelmättestaus, peli- tai käytettävyydestestaus, ohjelmiston haavoittuvuudet ja penetraatiotestaus sekä sosiaalinen hakkerointi.

Ammattikorkeakouluissa 50 % testauksen opetuksesta käsitteli joko ohjelmistotestauksen perusteita tai ohjelmistoturvallisuutta. Ohjelmistoturvallisuutta ja ohjelmistotestaamisen perusteita käsiteltiin testaukseen liittyvillä kursseilla yhtä paljon. Kolmanneksi eniten opetettiin pelitestausta, joka kattoi 14 % testauksen opetuksesta. Pelitestausta opetettiin ainoastaan ammattikorkeakouluissa, mikä voi johtua siitä, että pelien kehitys ja sen teoria on vasta viimeisen kymmenen vuoden aikana tullut yliopistojen tarjontaan, kun taas varsinainen pelien testaus on ammattikorkeakouluihin paremmin sopivaa käytännönläheistä testauksen ammatitilaan kuuluvaa toimintaa. Neljänneksi eniten ammattikorkeakouluissa opetettiin käytettävyydestestausta, jonka osuus oli 11 % koko testauksen opetuksesta.

Muut aiheet (ohjelmiston laatu, ketterä testaus, vaatimukseen perustuva testaus sekä testaus ja ohjelmistokehitys) kattoivat yhteensä 25 % koko testauksen opetuksesta. Näitä aiheita opetettiin suhteessa muihin aiheisiin kaikkein vähiten. Missään ammattikorkeakoulussa ei opetettu erillisellä kurssilla testaustekniikoita, mikä voi johtua siitä, että SWEBOK-mallissa erikoistestaustekniikoiden hallintaa ei ole erityisemmin korostettu. Lisäksi erikoistestaustekniikat mielletään ISTQB:n sertifikaattisisällössäkkin jatkotasolla käsiteltäviksi aiheiksi. Kuten aiemmin nähtiin, ammattikorkeakouluissa testausta opetettiin lähinnä alemmalla korkeakoulutasolla, joten opetuksen järjestäjät eivät välttämättä ole halunneet sisällyttää edistyneempiä testaustekniikoita tutkinto-opetukseen.

Yliopistoissa testauksen kurssien sisällöt vaihtelivat ammattikorkeakouluihin verrattuna vähemmän. Koulutusohjelmasta riippuen testausta opetetaan pääasiassa ohjelmistoturvallisuuden osana tai erillisellä ohjelmistotestauksen perusteita käsittelevällä kurssilla. Yliopistoissa opetettiin eniten ohjelmistoturvallisuutta, joka kattoi 35 % testauksen opetuksesta. Toiseksi

eniten opetettiin ohjelmistotestauksen perusteita, joka kattoi 31 %. Yhteensä nämä kaksi aihealuetta kattoivat 66 % kaikesta yliopistojen testausopetuksesta. Muut ohjelmistotestaukseen liittyvät yliopistotason kurssit käsittelivät lähinnä käytettävyydestä tai ohjelmistojen laatua sekä vähissä määrin ketterää testausta ja testaustekniikoita. Muut testausalueet kattoivat noin yhden kolmasosan koko yliopistojen testausopetuksesta. Yliopistoissa ei järjestetty ollenkaan pelitestaukseen, vaatimuksiin perustuvaan testaukseen tai testaus ja ohjelmistokehitys -luokkiin kuuluvia kursseja. Syynä tähän voi olla se, että kyseiset testauksen osa-alueet ovat melko käytännönläheisiä, joten niiden opetus on siksi voitu jättää yliopistoissa vähemmälle huomiolle. Toinen syy erikoistestausalueiden puuttumiselle voi olla se, että yliopistoissa testaus on keskittynyt ylempiin korkeakoulututkintoihin, ja ylemmän tason tutkinnoissa ei ole välttämättä aina tilaa kovin monelle erikoiskurssille.

Yhteenvedon Suomen korkeakouluissa opetettiin yhteensä 56 % ohjelmistotestauksen perusteita ja ohjelmistoturvallisuutta. Ohjelmiston laadun ja käytettävyydestä osuus koko opetuksesta oli yhteensä 23 %. Muiden aiheiden opetus kattoi 21 % korkeakoulujen testausopetuksesta. Alla olevassa Taulukossa 6 kuvataan erityisesti ohjelmistotestauksen perusteiden opetuksen määrää suomalaisissa IT-alaa opettavissa ammattikorkeakouluissa (Amk) ja yliopistoissa (Yo) suhteessa IT-alaa opettavien yliopistojen ja ammattikorkeakoulujen kokonaismääriin. Tiedot on koottu tässä tutkimuksessa tehdyn kartoituksen perusteella.

Korkeakoulu	Amk luku- määrä	Amk (%)	Yo luku- määrä	Yo (%)	Yht.	Yht. (%)
IT-alaa voi opiskella pääaineena	21		11		32	
ohjelmistotestauksen opetusta järjestetty	15	71 %	10	91 %	25	78 %
ohjelmistotestauksen peruskurssi järjestetty	9	43 %	8	73 %	17	53 %
ohjelmistotestauksen peruskurssi pakollinen	7	33 %	3	27 %	10	31 %

Taulukko 6. Ohjelmistotestauksen perusteiden opetus IT-alan korkeakouluissa

Taulukosta 6 käy ilmi, että Suomessa ohjelmistotestausta opetetaan 71 % kaikista IT-alaa opettavista ammattikorkeakouluista ja 91 % kaikista yliopistoista, eli yhteensä 78 % kaikista korkeakouluista. Yliopistoissa testausopetusta järjestetään siis suhteessa enemmän kuin ammattikorkeakouluissa. Syynä tähän voi olla se, että ammattikorkeakoulut ovat voineet profiloitua IT-alan opetuksessaan johonkin muuhun kuin ohjelmistotekniikkaan, kun taas yliopistoissa on voitu käsitellä tasaisemmin kaikkia ohjelmistotekniikan osa-alueita.

Ohjelmistotestauksen perusteita käsittelevää kurssia järjestetään 43 % kaikista IT-alaa opettavista ammattikorkeakouluista ja 73 % kaikista IT-alaa opettavista yliopistoista, eli yhteensä vain 53 % kaikista IT-alaa opettavista korkeakouluista. Pelkästään ohjelmistotestauksen perusteita käsittelevää kurssia tarjottiin 9 ammattikorkeakoulussa, ja näissä testauksen peruskurssi oli IT-alan tutkinto-ohjelmaan kuuluva pakollinen opintojakso 7 ammattikorkeakoulussa (Jyväskylän, Lahden, Lapin, Oulun, Satakunnan, Tampereen ja Vaasan ammattikorkeakouluissa). Ohjelmistotestauksen perusteita jollakin tavalla käsittelevää kurssia tarjottiin 8 yliopistossa, ja näissä jonkinlainen testauksen peruskurssi oli IT-alan tutkinto-ohjelmaan kuuluva pakollinen opintojakso ainoastaan kolmessa yliopistossa: Oulun yliopistossa, Tampereen teknillisessä yliopistossa sekä Lappeenrannan teknillisessä yliopistossa. Yliopistoissa keskitytään testauksen opetuksessa siis selvästi ammattikorkeakouluja enemmän ohjelmistotestauksen perusteisiin. Tämä voi johtua siitä, että yliopistojen opetus on teoriapainotteisempaa kuin ammattikorkeakouluissa.

Testauksen perusteiden osaamisen kannalta huolestuttavaa on, että testauksen peruskurssi oli pakollinen opintojakso vain noin 33 % IT-alaa opettavista ammattikorkeakouluista ja noin 27 % IT-alaa opettavista yliopistoista. Testauksen peruskurssi oli siis pakollinen vain 31 % kaikista IT-alaa opettavista korkeakouluista. Ero oli suurempi yliopistoissa: testauksen perusteita järjestettiin useimmissa IT-alaa opettavissa yliopistoissa, mutta vain noin joka kolmannessa IT-alaa opettavassa yliopistossa testauksen peruskurssi oli pakollinen. Syynä tähän on voi olla se, että testausta opetetaan useimmiten ylemmissä korkeakoulututkinnoissa, joissa opiskelijat voivat valita mihin osa-alueeseen haluavat suuntautua.

Kolmea suurinta luokkaa vertailtaessa huomataan, että ohjelmistotestauksen perusteita käsitteleviä kursseja oli yhteensä 17, ja aihetta opetettiin eniten pakollisena aiheena alemmissa

korkeakoulututkinnoissa. Ohjelmistoturvallisuutta käsitteleviä kursseja oli yhteensä 18, ja ohjelmistoturvallisuuteen liittyviä testauksen kursseja opetettiin valinnaisina tai pakollisina aiheina ylemmissä korkeakoulututkinnoissa. Käytettävyyttä opetettiin 8 kurssilla pakollisena tai valinnaisena. Käytettävyyden opetus jakaantui muista testauksen aiheista poiketen tasaisemmin alempien ja ylempien korkeakoulututkintojen välillä. Syynä tähän voi olla korkeakoulujen profiloituminen: käytettävyydestä ei välttämättä ole useimmissa korkeakouluissa koettu tärkeäksi testauksen osa-alueeksi.

4.1.4 Testausopetuksen heikkoudet ja puutteet

Kolmannen tutkimuskysymyksen kannalta oleellista oli kartoittaa korkeakoulujen testauksen opetuksessa olevia puutteita. Tässä tutkimuksessa tehdyn testausopetuksen kartoituksen perusteella suomalaisten korkeakoulujen testausopetuksen yksi heikkous on se, että suurin osa testauskursseista on vapaavalintaisia opintoja. Ammattikorkeakoulujen 36 testauskursista 16 kurssia on valinnaisia, ja samoin myös yliopistojen 26 testauskursista 16 kurssia on valinnaisia. Yhteensä aineiston 62 kurssista 32 kurssia eli noin 51 % on valinnaisia testauskursseja. Testauskursien valinnaisuus voi johtua siitä, etteivät opetuksen suunnittelijat koe testausta varsinaisena ohjelmistotekniikan ydinosaamisena, koska SE2014-opetussuosituksissa testauksen aihealueelle on varattu minimissäänkin vain 18 luentotuntia. Näin ollen on voitu ajatella, että testaus on jotenkin muita aiheita vähäpätöisempi aihealue, eikä kaikkien ohjelmistotekniikan opiskelijoiden tarvitse sitä välttämättä osata. Opiskelijat eivät välttämättä valitse testaukseen liittyviä kursseja niiden ollessa tutkinto-ohjelmassa valinnaisina, joten testausta käsitteleviä kursseja olisi tulevaisuudessa hyvä lisätä pakollisena osana koulutusohjelmiin opiskelijoiden testausosaamisen parantamiseksi.

Suomalaisten korkeakoulujen testausopetuksen laadussa on myös puutteita ohjelmistotekniikan uusimpiin opetussuositukseen verrattuna. SE2014-opetussuosituksessa listataan seuraavat aiheet, joita käsitteleviä kursseja löytyi hyvin vähän tai ei ollenkaan korkeakoulujen lukuvuoden 2016–2017 opetusohjelmasta.

- Mustalaatikkotestaustekniikat (VAV.tst.4)
- Testaustyökalut ja testausautomaatio (VAV.tst.11)

- Suorituskykytestaus (VAV.tst.14)
- Ohjelmistoturvallisuuteen liittyvä verifiointi ja validointi (SEC.dev.5)

Edistyneempiä mustalaatikkotestaustekniikoita kuten tutkivaa testausta tai mallipohjaista testausta käsiteltiin vain Oulun yliopiston *Testaustekniikat*-kurssilla. Muissa korkeakouluissa näitä testaustekniikoita opetettiin vähäisemmässä määrin lähinnä muutamalla ohjelmistotestauksen perusteita käsittelevällä kurssilla. Yksinomaan testaustekniikoihin keskittyvien kurssien puute voi johtua siitä, ettei opetussuunnitelmassa yksinkertaisesti ole tilaa ns. erikoisille kursseille. On selvää, ettei 3 tai 5 opintopisteen peruskurssilla voi opettaa kovin syvällisesti testaustekniikoita, joten tulevaisuudessa olisi hyvä saada lisää erilaisia testaustekniikkakursseja korkeakoulujen kurssivalikoimiin.

Testausautomaatio mainittiin kurssin sisältökuvauksissa kurssilla käsiteltävänä aiheena ainoastaan kolmella kurssilla:

- Emerging Trends in Software Testing (Oulun yliopisto)
- Software Quality and Testing (Oulun yliopisto)
- Järjestelmätestaus (Jyväskylän ammattikorkeakoulu)

Näilläkään kursseilla testausautomaatiota tai siihen liittyviä työkaluja ei ilmeisesti käsitelty kovinkaan syvällisesti. Muut korkeakoulut eivät huomioineet testausautomaatiota ollenkaan opetussuunnitelmissaan. Suorituskykytestausta ei opetettu yhdessäkään korkeakoulussa. Näin ollen korkeakoulujen testausopetuksen yksi puute on selvästi pelkästään suorituskykyä sekä testausautomaatiota tai sen työkaluja ja automaatiotestitapausten skriptaamista käsittelevien kurssien puuttuminen opetusohjelmasta.

Yksinomaan tietoturvatestaukseen keskittyneitä kursseja tarjottiin ammattikorkeakouluissa ja yliopistoissa vähän, sillä tietoturvatestausta käsittelevillä kursseilla käsiteltiin usein myös muita tietoturvaan ja tietoturvalliseen ohjelmointiin liittyviä aiheita. Tietoturvatestausta -sana mainittiin ainoastaan seuraavien kurssien sisältökuvauksessa:

- Software Security (Aalto-yliopisto/Helsingin yliopisto)
- Data Security Testing (Jyväskylän ammattikorkeakoulu)
- Auditing and Testing Technical Security (Jyväskylän ammattikorkeakoulu)

Tietoturvatestaukseen erikoistuneiden kurssien puuttumisen syynä voi olla se, että Ohjelmistoturvallisuus on lisätty vasta neljä vuotta sitten uutena osaamisalueena SE2014-opetus-suositukseen. Näin olleen tietoturvatestausta ei ole välttämättä vielä edes ehditty lisätä uusimpien koulutusohjelmien sisältöihin, etenkin jos opetussuunnitelmien pääpaino on ollut tietoturvalisessä ohjelmoinnissa. Pelkästään tietoturvatestausta käsitteleviä kursseja olisi hyvä saada pikimmiten korkeakoulujen kurssivalikoimaan. Näin opiskelijoita voitaisiin kannustaa hankkimaan työelämässä tällä hetkellä kysyttyä testausosaamista.

Suomalaisten korkeakoulujen testausopetuksen sisältöjä ja ISTQB:n sertifikaattisisältöjä verrattaessa huomattiin, ettei korkeakoulujen testausopetuksessa juurikaan opeteta ISTQB:n perussertifikaatin laajennussertifikaatteihin kuuluvia sisältöä käytettävyydestä lukuun ottamatta. Myöskään testausautomaatioinsinööri, testausasiantuntija tai testauspäällikkösertifikaatteihin kuuluvia sisältöjä ei korkeakouluissa opetettu. ISTQB:n sertifikaattisisältöjä on voitu pitää liian erikoisosaamisena, jotta niitä vastaavia kursseja olisi lisätty korkeakoulujen opetussuunnitelmiin. Esimerkiksi mallipohjaista testausta, testauksen hallintaa ja testausautomaatiota käsitellään SWEBOK-mallissa lyhyesti vain muutamissa kohdissa, ja siksi näiden painotus SE2014-opetussuosituksissa on melko vähäinen (testausautomaatio, mallipohjainen testaus) tai vähäinen (testauksen hallinta). Termiä ketterä testaus (engl. *agile testing*) ei edes mainita SWEBOK-mallissa tai SE2014-opetussuosituksissa. Korkeakoulujen tulisi vähintäänkin olla tietoisia ISTQB:n toiminnasta ja sen tarjoamista testaussertifikaateista ja niiden avoimista sisällöistä. Tiiviimpi yhteistyö oppilaitosten ja FiSTB:n välillä (esimerkiksi useampina luentoina tai koko kurssin mittaisina vierailijaluentosarjoina) voisi edistää ISTQB:n sertifikaattisisällöissä mainittujen testausaiheiden tuomista suomalaisiin ammattikorkeakouluihin ja yliopistoihin.

Suomalaisten korkeakoulujen ohjelmistotestauksen opetusta voitaisiin kehittää monella tavalla. Yksinkertaisena opetuksen kehittämisideana voisi olla juuri edellä mainittuja puuttuvia aihekokonaisuuksia käsittelevien kurssien lisääminen oppilaitosten tarjontaan ja pakollisten testauskurssien lisääminen tutkintovaatimuksiin. Lisäksi ohjelmistotestauksen opetusta voitaisiin integroida jokaiseen kurssiin läpi IT-alan koulutusohjelman. Useimpia SE2014-opetussuosituksessa listattuja aiheita voitaisiin myös esimerkiksi opettaa kokonaan testauksen hallinnan näkökulmasta. Ohjelmistotestauksen opetuksen vieminen verkkoon etäopetuksena

voisi olla myös eräs vaihtoehto ohjelmistotestauksen opetuksen kehittämiseksi, sillä korkeakoulut Jyväskylän ammattikorkeakoulua lukuun ottamatta eivät hyödyntäneet verkko-oppimista tai verkko-oppimisympäristöjä ohjelmistotestauksen etäopiskelussa. Lisäksi ainoastaan Jyväskylän ammattikorkeakoulussa yhden testauskurssin (*Agile Testing*) päätteeksi oli mahdollista suorittaa ISTQB:n testausertifiointi. Testauksen sertifikaatteja arvostetaan työelämässä, joten ainakin perustestausertifikaattien suorittamiseen tähtäävät opintojaksot voisivat olla yksi testauksen opetuksen kehittämiskohde. Ainoastaan Jyväskylän ammattikorkeakoulussa ohjelmistotestauksesta voi opiskella erillisen opintokokonaisuuden (*Ethical Hacking*, Jyväskylän ammattikorkeakoulu 2016a), mutta muissa korkeakouluissa ohjelmistotestauksen opintokokonaisuuksia ei tarjottu lukuvuonna 2016–2017. Yhtenä vaihtoehtona voisi siten olla erillisen ohjelmistotestauksen perus- tai aineopintokokonaisuuden kehittäminen. Tämän tutkimuksen pohjalta päädyttiin kehittämään ohjelmistotestauksen opetusta luomalla uusi ohjelmistotestauksen perusopintokokonaisuus, jonka sisältöä kuvataan seuraavissa luvuissa.

4.2 Ohjelmistotestauksen perusopintokokonaisuus

Tässä tutkimuksessa tehdyn testauksen opetuksen nykytilan kartoituksen, tutkimuksessa esiinnousseiden havaintojen ja testauksen opetuksessa ilmenneiden puutteiden, tutkimuksen tekijän testausalan työkokemuksen sekä alan kirjallisuuskatsauksen pohjalta suunniteltiin 25 opintopisteen laajuinen ohjelmistotestauksen perusopintokokonaisuus, joka on tarkoitettu ohjelmistotestauksen opetuksen kehittämiseen suomalaisissa korkeakouluissa. Opintokokonaisuus on suunniteltu otettavaksi käyttöön esimerkiksi korkeakoulujen vapaavalintaisina opintoina osana IT-alan tutkintoa. Ohjelmistotestauksen perusopintojen tarkoitus on tutustuttaa opiskelija yleisellä tasolla ohjelmistotestauksen ammattiin, testaajan, testauspäällikön tai testauskonsultin roolissa vaadittaviin tietoihin ja taitoihin sekä ohjelmistotestauksen alana. Ohjelmistotestauksen perusopintojen tarkoituksena on myös edistää ohjelmistotestauksista ammattina.

Tämän tutkimuksen tuloksena suunniteltu ja kuvattu testauksen perusopintokokonaisuus edustaa yhden testausalan ammattilaisen näkemystä siitä, miten ohjelmistotestauksen opetusta suomalaisissa korkeakouluissa voitaisiin parhaiten kehittää vastaamaan enemmän IT-

alan työelämävaatimuksia Suomessa. Testauksen perusopintokokonaisuutta ei tämän tutkimuksen puitteissa oteta käyttöön missään suomalaisessa korkeakoulussa. Opintokokonaisuus on suunniteltu testauksen opetuksen kehittämiseen tulevia IT-alan tutkintorakenteita ajatellen, mutta se voidaan ottaa sellaisenaan käyttöön myös muussa kontekstissa. Tässä tutkimuksessa syntynyt testauksen perusopintokokonaisuus soveltuisi hyvin myös toteutettavaksi ja suoritettavaksi itsenäisenä verkko-opiskeluna.

4.2.1 Opintokokonaisuuden kuvaus ja osaamistavoitteet

Ohjelmistotestauksen perusopinnot -opintokokonaisuus perehdyttää opiskelijan ohjelmistotestauksen periaatteisiin, käytäntöihin ja prosesseihin, sekä testauksen suunnitteluun, toteutukseen, raportointiin, hallintaan ja arviointiin erilaisissa ohjelmistoprojekteissa. Opintokokonaisuus antaa valmiuksia työskennellä esimerkiksi testaajan, testauspäällikön tai testausasiantuntijan tehtävissä.

Ohjelmistotestauksen perusopintokokonaisuus koostuu seuraavista seitsemästä kurssista:

1. Ohjelmistotestaus 5 op
2. Testausautomaatio 5 op
3. Ketterä ja tutkiva testaus 3 op
4. Testauksen hallinta ja johtaminen 5 op
5. Testaustekniikat 3 op
6. Käytettävyytestaus 3 op sekä
7. Tietoturvatestaus 3 op.

Ohjelmistotestauksen perusopinnot aloitetaan viiden opintopisteen laajuisella Ohjelmistotestauksen kurssilla, johon ei periaatteessa vaadita minkäänlaisia esitietoja. Tosin ohjelmistotuotannon perusteiden hallinnasta ei olisi haittaa. Ohjelmistotestauksen kurssi vaaditaan esitietona muille ohjelmistotestauksen perusopintoihin kuuluville kursseille, joiden suositeltu suoritusjärjestys on kuvattu yllä olevassa listassa. Ohjelmistotestauksen kurssin suoritettuaan opiskelija voi kuitenkin suorittaa muut kurssit haluamassaan järjestyksessä.

Viiden opintopisteen kurssit ovat perusopintojen ydinosaamisaluetta. Kolmen opintopisteen kurssit ovat helposti vaihdettavissa johonkin toiseen kurssiin sen mukaan, mikä testauksen osa-alue on kulloinkin ajankohtainen tai millä osa-alueella kaivataan eniten osaamista työelämässä. Myöhemmin perusopintokokonaisuuteen voidaan lisätä valinnaisuutta tarjoamalla lisää vähintään kolmen opintopisteen laajuisia johonkin testauksen osa-alueeseen liittyviä kursseja. Tällaisia kursseja voisivat olla esimerkiksi suorituskykytestaus, pelitestaus tai testausprosessin kehittäminen.

SE2014-opetussuosituksissa annetaan ohjeita ja suosituksia siihen, miten ohjelmistotekniikan opetussuunnitelma tulisi rakentaa. Opetussuosituksen mukaan opetuksen suunnittelijan tulisi miettiä opetussuunnitelmaa sen mukaan, millaisia taitoja opiskelijalla on valmistumisen jälkeen. (ACM & IEEE-CS 2015). Tässä tutkimuksessa suunnitellun ohjelmistotestauksen perusopintokokonaisuuden osaamistavoitteet ovat seuraavat: Opintokokonaisuuden suorittanut opiskelija tuntee testaukseen liittyvät käsitteet, periaatteet, testaustasot ja testausmenetelmät, ja pystyy soveltamaan niitä erilaisissa ohjelmistoprojekteissa. Opiskelija tunnistaa ja osaa vertailla erilaisia testausmenetelmiä ja työkaluja. Opiskelija hallitsee monipuolisesti erilaiset testaustekniikat sekä osaa hyödyntää erilaisia työkaluja testauksessa ja testauksen hallinnassa. Opiskelija osaa analysoida testauksen asemaa osana ketterää ohjelmistoprosessia. Opiskelija osaa tunnistaa testauksessa tarvittavan dokumentaation sekä sen tarpeen erikokoisissa ohjelmistoprojekteissa, ja osaa toteuttaa dokumentaation tarpeiden mukaan. Opiskelija osaa soveltaa teoriaa testauksen suunnitteluun ja osaa itseohjautuvasti suunnitella, toteuttaa, raportoida, hallita ja arvioida testausta ohjelmistoprojektissa sekä ohjelmiston elinkaaren kaikissa vaiheissa. Opiskelija osaa arvioida projektiin liittyviä laaturiskejä sekä testauksen työmäärää. Opiskelija osaa kehittää ohjelmistoprojektin testausta ja osaa hankkia ajantasaista tietoa testauksen kehittämisen tueksi. Opiskelija osaa suunnitella ohjelmistoprojektin testausautomaation ohjelmistoprojektin eri vaiheissa, ja lisäksi opiskelija osaa suunnitella, toteuttaa ja raportoida ohjelmiston käytettävyydestestauksen sekä tietoturvatestauksen. Opintokokonaisuuden suorittanut opiskelija pystyy toimimaan esimerkiksi ohjelmistotestaa-jana, teknisenä testausasiantuntijana, testausautomaation asiantuntijana, tietoturvatestaa-jana, testauspäällikkönä tai testauskonsulttina.

4.2.2 Opintokokonaisuuden kurssien kuvaus ja osaamistavoitteet

Ohjelmistotestauksen perusopintoihin kuuluvat seitsemän kurssia, niiden laajuudet ja vaaditut esitiedot sekä esimerkit aiemmin hankitun osaamisen tunnustamisesta (AHOT) on listattu Liitteessä B. Ohjelmistotestauksen perusopintoihin kuuluvat kurssit pyrittiin nimeämään siten, että kurssin nimestä olisi helposti pääteltävissä kurssin sisältö. Seuraavassa on kuvattu kunkin kurssin sisältö ja osaamistavoitteet lyhyesti siten, kuin ne olisi kuvattu minkä tahansa suomalaisen korkeakoulun opinto-oppaassa. Lopuksi perustellaan kyseisten kurssin ottaminen mukaan ohjelmistotestauksen perusopintokokonaisuuteen.

Ohjelmistotestaus-kurssilla käsiteltäviä aiheita ovat testauksen tavoitteet, termistö ja käsitteet, testauksen periaatteet, testauksen tasot ja menetelmät, katselmoinnit, ohjelmistojen yleiset virheet, testattavien asioiden ja yleisten virheiden tunnistaminen, testauksen työkalut, testausdokumentaatio, testauksen rajoitteet, testauksen suunnittelun merkitys ja perusteet, testitapausten suunnittelu ja kirjoitus, testauksen ja virheiden raportointi sekä testauksen asema ohjelmistokehityksessä. Kurssin suoritettuaan opiskelija hallitsee ohjelmistotestauksen perusteet, ja osaa suunnitella, toteuttaa ja raportoida testauksen käytännössä.

Testausautomaatio-kurssilla perehdytään testausautomaation perusteisiin, testausautomaation hyötyihin ja mahdollisiin ongelmiin, testausautomaation käyttöönottoon ketterässä kehityksessä sekä testausautomaation edellyttämään osaamiseen ja rooleihin. Lisäksi kurssilla käsitellään testausautomaation työvälineitä ja niistä erityisesti Robot Frameworkia sekä avainsanaohjattujen testitapausten kirjoittamista. Lisäksi perehdytään Python-ohjelmointikielen perusteisiin ja omien kirjastojen luomiseen testausautomaatiota varten. Kurssin suoritettuaan opiskelija hallitsee testausautomaation perusteet. Lisäksi opiskelija osaa suunnitella, kirjoittaa, ylläpitää, ajaa ja dokumentoida testejä käyttäen Robot Framework -työkalua.

Ketterä ja tutkiva testaus -kurssilla ohjelmistotestausta ja sen terminologiaa, tavoitteita, haasteita ja käytäntöjä tarkastellaan erityisesti ketterästä näkökulmasta. Kurssin aiheita ovat ketterän kehityksen ja testauksen perusteet, testaus ja sen rooli ketterässä ohjelmistokehityksessä sekä ketterän testauksen työkalut. Lisäksi kurssilla perehdytään www-sovellusten testaukseen ja erityisesti ketterissä ohjelmistoprojekteissa hyödynnettävään tutkivaan testaukseen. Kurssin suoritettuaan opiskelija hallitsee ketterän kehityksen ja testauksen perusteet.

Lisäksi opiskelija osaa testata www-sovelluksia tutkivan testauksen keinoin ja raportoida testauksen löydökset.

Testauksen hallinta ja johtaminen -kurssi perehdyttää opiskelijan testauksen suunnitteluun, seurantaan ja hallintaan testauspäällikön näkökulmasta. Muita kurssilla käsiteltäviä aiheita ovat päätöskriteerien arviointi ja raportointi, riskipohjainen testaus, testausmetriikoiden määrittely ja käyttö, testauksen dokumentointi, havaintojen ja ohjelmistokehityksen elinkaari, testauksen kehittämisprosessi sekä testaustyökalujen ja testaustiimin valinta. Kurssin suorittuaan opiskelija osaa suunnitella, toteuttaa, arvioida ja kehittää ohjelmistoprojektiin liittyvää testausta.

Testaustekniikat-kurssin sisältönä ovat erilaiset mustalaatikkotestaustekniikat. Kurssilla perehdytään SWEBOK-mallissa esitettyihin testaustekniikoihin, joita ovat erilaiset intuitioon ja kokemukseen, syötettyihin arvoihin, lähdekoodiin, virheisiin, sovelluksen käyttöön, ohjelman luonteeseen, malliin, tekniikoiden valintaan ja yhdistelyyn perustuvat testaustekniikat. Kurssin suorittuaan opiskelija hallitsee erilaisten mustalaatikkotestaustekniikoiden peruseräatteen ja ymmärtää niiden erot. Lisäksi opiskelija osaa soveltaa mustalaatikkotestaustekniikoita erilaisten ohjelmistojen testaukseen.

Käytettävyydestaus-kurssilla perehdytään käytettävyydestauksen perusteisiin, käytettävyyden arviointimenetelmiin, käytettävyydestausmenetelmiin (heuristisen arviointi, kognitiivisen läpikäynti ja loppukäyttäjättestaus) sekä käytettävyydestauksen läpivienttiin. Muita aiheita ovat käytettävyyden psykologia, käytettävyydestaussuunnitelman tekeminen sekä käytettävyydestauksen suorittaminen, analysointi ja raportointi. Kurssin suorittuaan opiskelija osaa suunnitella, toteuttaa ja raportoida käytettävyydestauksen sekä osaa analysoida käytettävyydestauksen tuloksia.

Tietoturvatestaus-kurssi perehdyttää opiskelijan monipuolisesti ohjelmiston tietoturvaan ja tietoturvatestaukseen. Kurssi käsittää seuraavat aihealueet: Tietoturva ja tietosuojan perusteet, tiedon suojausmenetelmät, hyvä tiedonhallintatapa ja -käsittelytapa, ohjelmistovarmuus, tavallisimmat haittaohjelmat ja web-hyökkäykset, tietoturvan arviointi ja auditointi sekä eettinen hakkerointi, standardisointi ja tietoturvastandardit sekä OWASP TOP 10 -lista,

tietoturvatestaus (fuzz-testaus, penetraatiotestaus), asiakas- ja palvelinohjelmistojen tietoturvatestaus, tietoturvan testausmenetelmät ja -tekniikat, erilaiset tietoturvatestaustyökalut ja testaustulosten käsittely sekä tietoturvatestauksen dokumentointi ja raportointi. Kurssin suoritettuaan opiskelija osaa suunnitella, toteuttaa ja raportoida tietoturvatestauksen sekä osaa analysoida tietoturvatestauksen tuloksia.

Edellä mainitut kurssit ja niiden sisällöt valittiin osaksi testauksen perusopintokokonaisuutta tämän tutkimuksen tuloksissa ilmenneisiin testausopetuksen puutteisiin sekä toisaalta yleisimpiin ISTQB:n testaussertifikaattien sisältöihin nojaten. Opintokokonaisuuteen valittiin juuri nämä kurssit seuraavista syistä. *Ohjelmistotestaus*-kurssi valittiin, koska useimmissa korkeakouluissa opetettiin ainakin jonkinlaisia testauksen perusteita. Lisäksi SWEBOK-mallissa yksi testauksen keskeisistä aihealueista on juuri testauksen perusteet. Lisäksi mikä tahansa opintokokonaisuus on helppo aloittaa aihealueen ”peruskurssilla”, jossa kuvataan aiheeseen liittyvä terminologia ja osa-alueet. Ohjelmistotestauksen peruskurssi luo SWEBOK-mallin kuuden keskeisen aihealueen kaltaisen pohjan muille ohjelmistotestauksen perusopintokokonaisuuteen valituille kursseille. Kurssin suoritus vastaa kutakuinkin ISTQB:n sertifikaattisisältöjen sertifioitu testaaaja-sertifikaatin suoritusta.

Testausautomaatio, Ketterä ja tutkiva testaus ja *Testaustekniikat* -kurssit haluttiin lisätä perusopintokokonaisuuteen, koska kyseisiä aiheita käsittelevät kurssit näyttäisivät tämän tutkimuksen tulosten perusteella puuttuvan kokonaan tai osittain useimpien korkeakoulujen kurssitarjonnasta. Myös testauksen hallintaa ja johtamista käsiteltiin korkeakoulujen testauskursseilla hyvin vähän. ISTQB:n jatkotason Testauspäällikkö-sertifikaattisisältö kuitenkin käsittelee varsin kattavasti testauksen hallintaa ja läpivientiä projekteissa, joten tästä syystä koettiin, että perusopintokokonaisuuteen olisi hyvä lisätä kyseistä aihetta käsittelevä *Testauksen hallinta ja johtaminen* -kurssi. *Käytettävyydestaus* ja *Tietoturvatestaus* -kurssit valittiin, koska kyseisiä aiheita opetetaan vain muutamissa korkeakouluissa. Erilaiset www- ja mobiilisovellukset ovat kuitenkin yleistyneet räjähdysmäisesti tällä vuosituhanella, ja käytettävyydestaus ja tietoturvatestaus voivat siten olla tulevaisuudessakin tärkeitä testauksen osa-alueita. Esimerkiksi ISTQB on reagoinut tähän testaustarpeeseen lisäämällä tietoturva-testaaaja sekä käytettävyydestaaja-sertifikaatit sertifiointijärjestelmäänsä.

Tämän tutkimuksen tulosten perusteella näyttää siltä, että suorituskkytestaus puuttuu kokonaan suomalaisten ammattikorkeakoulujen ja yliopistojen kurssivalikoimista. Suorituskkytestausta ei kuitenkaan lisätty yhdeksi kurssiksi ohjelmistotestauksen perusopintokokonaisuuteen siitä syystä, että opintokokonaisuus haluttiin pitää minimissään eli 25 opintopisteen laajuisena. Lisäksi suorituskkytestaus vaatii ohjelmistotestauksen syvällistä ymmärtämistä, ja paljon teknistä osaamista sekä ohjelmistotekniikan tuntemusta, joten aihealue sopii paremmin esimerkiksi testauksen aineopintoihin. Koska perusopintokokonaisuus on suunniteltu opiskelijoille, joilla ei välttämättä ole laajaa tietämystä ohjelmistotekniikasta, päätettiin suorituskkytestaus jättämään pois perusopintokokonaisuudesta.

Seuraavassa Luvussa 5 poimitaan tämän tutkimuksen oleellisempia tuloksia, pohditaan niiden merkitystä ja vertaillaan tuloksia aiempaan tutkimukseen. Lisäksi pohditaan tämän tutkimuksen luotettavuutta.

5 Pohdinta

Tässä pro gradu -tutkielmassa pyrittiin saamaan vastauksia kolmeen tutkimuskysymykseen. Ensimmäinen tutkimuskysymys kuului ”Mitä ohjelmistotestauksesta opetetaan suomalaisissa korkeakouluissa?” Tiivistettynä tässä tutkimuksessa tehdyn kartoituksen perusteella suomalaisissa ammattikorkeakouluissa ja yliopistoissa opetetaan ohjelmistotestauksesta pääasiassa ohjelmistotestauksen perusteita ja ohjelmistoturvallisuutta. Lisäksi joissakin korkeakouluissa opetetaan erityisillä kursseilla käytettävyydestä, ohjelmistojen laatua sekä pelitestausta. Muita vähäisempiä testauskursseilla käsiteltyjä aihealueita ovat ketterä testaus, testaustekniikat, vaatimukseen perustuva testaus sekä testauslähtöinen ohjelmistokehitys. Ammattikorkeakoulujen ja yliopistojen testausopetus eroaa sisällön osalta toisistaan siinä, että yliopistoissa ei järjestetä ollenkaan pelitestaukseen, vaatimukseen perustuvaan testaukseen tai testauslähtöiseen ohjelmistokehitykseen keskittyviä kursseja. Ammattikorkeakouluissa ohjelmistotestausta ja siihen liittyviä ohjelmistoturvallisuuden ja ohjelmistojen laadun aiheita opetetaan useimmiten kolmen tai viiden opintopisteen laajuisilla pakollisilla kursseilla alemmissa korkeakoulututkinnoissa. Yliopistoissa ohjelmistotestauksen opetus on keskittynyt pääosin maisteritasoisille valinnaisille viiden opintopisteen kursseille.

Toinen tutkimuskysymys kuului: ”Miten ohjelmistotestauksen opetus Suomessa vastaa kansainvälisiä opetussuosituksia?” Vastauksena tähän tutkimuskysymykseen voidaan todeta, että IT-alaa opettavissa yliopistoissa annettu testauksen ja siihen läheisesti liittyvien aihealueiden opetus vastaa pääosin varsin hyvin kansainvälisiä suosituksia sekä määrällisesti että laadullisesti. Yliopistoissa testausta ei kuitenkaan pääsääntöisesti opeteta alemmissa korkeakoulututkinnoissa, mikä on selkeä puute. Lisäksi ainoastaan noin joka viidennessä IT-alaa opettavassa ammattikorkeakoulussa ohjelmistotestauksen ja siihen liittyvien ohjelmistoturvallisuuden ja ohjelmistojen laadun opetus vastaa määrällisesti ja laadullisesti kansainvälisiä suosituksia. Testausta ei muutamaa poikkeusta lukuun ottamatta opeteta ylemmissä ammattikorkeakoulututkinnoissa, mikä on myös yksi testausopetuksen heikkous.

Testauksen perusteiden osaamisen kannalta on huomattava, että testauksen peruskurssi on pakollinen opintojakso vain noin 33 % IT-alaa opettavista ammattikorkeakouluista ja noin 27 % IT-alaa opettavista yliopistoista. Testauksen peruskurssi on pakollinen siis vain 31 %

kaikista IT-alaa opettavista korkeakouluista. Oppisisältöjen näkökulmasta korkeakoulujen testausopetuksen heikkous on se, ettei korkeakouluissa juuri opeteta testauksen hallintaa, suorituskykytestausta, testausautomaatiota ja sen työkaluja, varsinaista tietoturvatestausta tai erityisiä mustalaatikkotestaustekniikoita kuten tutkivaa testausta tai mallipohjaista testausta.

Suomessa korkeakoulut saavat itse päättää opetuksensa oppisisällöt, mikä voi olla yksi syy ammattikorkeakoulujen ja yliopistojen testausopetuksen eri aihealueiden painotuksen eroihin sisällöissä, kurssien pakollisuudessa ja laajuudessa. Lisäksi Luvussa 2 esitettyä SWE-BOK-mallia ja kansainvälisiä opetussuosituksia on voitu tulkita eri oppilaitoksissa eri tavoin opetussuunnitelmien sisältöjä suunniteltaessa. Yliopistojen profiloituminen muille tietotekniikan aloille tai ohjelmistotekniikan tietopohjan muiden osa-alueiden painottuminen opetussuunnitelmissa voivat olla syynä siihen, ettei testausta opeteta kaikissa alemmissa korkeakoulututkinnoissa. Ammattikorkeakouluissa testauksen puuttuminen ylemmistä ammattikorkeakoulututkinnoista voi johtua siitä, että koko tutkinto on suhteellisen nuori.

Tässä tutkimuksessa kävi ilmi, että SE2014-opetussuosituksista puuttuu ketterä testaus ja pelitestausta, joihin liittyviä kursseja kuitenkin opetetaan Suomen ammattikorkeakouluissa. Oppilaitokset ovat voineet haluta painottaa pelitestausta, koska se on tällä hetkellä yksi ajan-kohtainen testauksen osa-alue Suomessa. Suomalaisten peliyhtiöiden viime vuosien menestys on voinut vaikuttaa siihen, että pelikehittämisen ja -testauksen kursseja on alettu järjestää myös ammattikorkeakouluissa, jotta työelämän kasvavaan osaajatarpeeseen voitaisiin vastata paremmin.

Kolmas tutkimuskysymys kuului ”Miten ohjelmistotestauksen korkeakouluopetuksen sisältöä voitaisiin kehittää?” Tässä tutkimuksessa koostettiin tutkimustulosten, testauksen korkeakouluopetuksessa havaittujen puutteiden ja valitun lähdemateriaalin perusteella yhdeksi testauksen opetuksen kehitysideaksi 25 opintopisteen laajuinen ohjelmistotestauksen perusopintokokonaisuus. Opintokokonaisuus sisältää 7 kurssia, jotka käsittelevät ohjelmistotestauksen perusteita, testausautomaatiota, ketterää ja tutkivaa testausta, testauksen hallintaa ja johtamista, testaustekniikoita, käytettävyydestä sekä tietoturvatestausta. Ohjelmistotes-

tauksen perusopintokokonaisuuden käytännön työelämähyöty on siinä, että IT-ala ja erityisesti ohjelmistoteollisuus on kasvava teollisuudenala Suomessa ja maailmalla. Etenevä digitalisaatio lisää ohjelmisto-osaamisen tarvetta johtajista suunnittelijoihin, ja myös ohjelmistotestauksen ammattilaisia tarvitaan tulevaisuudessa yhä enenevässä määrin varmistamaan kehitettävien ja uudistettavien ohjelmistojen laatu yrityksissä sekä julkisella sektorilla. Ohjelmistotestauksen opintokokonaisuus pyrkii testauksen edistämiseen ammattina ja opiskelijoiden testauksen taitojen kartuttamiseen.

Tämä tutkimus lisäsi tietoa siitä, mitä testauksesta opetetaan ammattikorkeakouluissa ja miten hyvin ammattikorkeakoulujen testausopetus vastaa ohjelmistotekniikan kansainvälisiä opetussuosituksia. Aihetta ei oltu aiemmin tutkittu juuri ammattikorkeakoulujen osalta. Kytälän (2011) tutkimuksessa kävi aiemmin ilmi, että yliopistoissa ei yleensä opeteta ohjelmistotestausta ennen syventäviä kursseja. Lisäksi Mynttisen (2017) tutkimuksessa kävi aiemmin ilmi, että alemmissa korkeakoulututkinnoissa on puutteita testauksen opetuksessa. Tämä tutkimus vahvisti näitä kahta näkemystä erityisesti yliopistojen mutta myös ammattikorkeakoulujen osalta.

Tämän tutkimuksen teon aikana kohdattiin myös joitakin ongelmia, jotka liittyivät aineiston rajaamiseen. Aineistoon ei otettu mukaan ohjelmistotuotannon kursseja, joilla testausta kuitenkin käsitellään vaihtelevasti osana kurssia. Jos nämä kurssit olisi otettu mukaan tutkimusaineistoon, aineisto olisi ollut isompi, ja testauksen opetus laajempaa lähes kaikissa korkeakouluissa. Toisaalta oli perusteltua jättää kurssit kokonaan pois, koska näin pystyttiin keskittymään niihin kursseihin, joilla testauksesta opetetaan SE2014-opetussuosituksissa mainittuja aiheita.

Tutkimusaineistoon liittyy myös rajoite, joka koski aineiston rajaamista pelkästään virallisiin tutkintovaatimuksiin ja kurssikuvauksiin. Tämä tutkimusaineiston rajaaminen voi mahdollisesti aiheuttaa virheellisiä johtopäätöksiä sen suhteen, mitä korkeakouluissa oikeasti opetetaan testauksesta, sillä viralliset tutkintovaatimukset eivät aina vastaa todellisuutta. Tutkimusaineistosta olisi voitu saada tarkempi, jos tutkimus olisi toteutettu esimerkiksi kyselylomakkeella, joka olisi lähetetty yliopistojen ja ammattikorkeakoulujen IT-alan opetuksen suunnittelusta vastaaville henkilöille.

Myös tutkimusaineistoa analysoitaessa kohdattiin joitakin ongelmia. Esimerkiksi kaikki tietoturvaohjelmointia ja tietoturvatestausta käsittelevät kurssit niputettiin samaan kategoriaan, koska aineistoa analysoitaessa ei voitu pelkän kurssin sisältökuvauksen perusteella tietää, miten paljon kurssilla käsiteltiin tietoturvatestausta. Samasta syystä myös jotkut ohjelmistojen laatua käsittelevät kurssit otettiin mukaan aineistoon, koska pelkkiä kurssikuvauksia tarkastelemalla ei saatu varmuutta siitä, miten paljon kyseisillä kursseilla käsiteltiin ohjelmistotestausta.

Suomalaisten korkeakoulujen testauksen opetuksen laajuuden vertaaminen annettuihin suosituksiin oli hankalaa. Ensinnäkään SWEBOK-malli ei oteta kantaa, kuinka paljon kunkin testauksen osaamisalueen opetukseen tai oppimiseen tulisi käyttää aikaa, tai kuinka kauan tietyn aiheen opiskeluun menee. Lisäksi SE2014-opetussuosituksessa osaamisalueen opettamiseen kuluva aika kuvataan luentotunteina, mutta aika ei sisällä opiskelijan tekemää itsenäistä työtä. Suomessa korkeakouluissa opintojaksot mitoitetaan opintopisteinä, mikä puolestaan sisältää luentojen lisäksi opiskelijan tekemän itsenäisen työn. Yhteensopivuusongelma ratkaistiin arvioimalla, mitä SE2014-opetussuosituksessa annettu luento-opetukseen minimissään käytettävä aika sekä opiskelijalta edellytettävä itsenäinen työ olisi yhteensä opintopisteinä. Aiheen oppimiseen kuluva kokonaistuntimäärä muutettiin opintopisteiksi käyttäen jakajana 27 tuntia, mikä vastaa suurin piirtein yhtä opintopistettä. Näin saatiin testauksen opetuksen minilajuudeksi 8 opintopistettä, mikä on siis vain karkea arvio.

Luvussa 3 esitettiin tässä tutkimuksessa käytetty ohjelmistotestauksen määritelmä, joka oli suhteellisen laaja, sillä se sisälsi laatu- ja turvallisuuden lisäksi myös tietoturvallisuuden käsitteen. Siksi opintojen vastaavuutta arvioitaessa huomioitiin myös SE2014-opetussuosituksen Ohjelmistoturvallisuuden ja Ohjelmistojen laadun osaamisalueet, mikä laajensi testauksen aihealueiden oppimiseen kuluva minimimäärää luento-opetus ja opiskelijan itsenäinen työ mukaan luettuna. Mikäli Ohjelmistoturvallisuutta ja Ohjelmistojen laatua ei oltaisi huomioitu tutkimusaineiston analyysissä, olisi testauksen minimivaatimus ollut noin 5 opintopistettä ja se olisi täytynyt useimmilla oppilaitoksilla. Toisaalta useampi kurssi olisi tällöin pitänyt jättää aineistosta pois, koska aineistonhaussa käytettiin myös hakutermejä ”laatu” ja ”turvallisuus”. Näin saatu aineisto olisi ollut huomattavasti pienempi, jolloin tutkimuksen

yleistettävyys olisi voinut kärsiä. Ohjelmistoturvallisuus ja Ohjelmistojen laatu haluttiin ottaa mukaan testausta käsitteleviksi aihealueiksi, koska SE2014-opetussuosituksissa todetaan näiden osa-alueiden läheinen suhde paitsi testaukseen, myös muihin SE2014-opetussuosituksessa esitettyihin osaamisalueisiin. Tämä saattaa lisätä myös tutkimuksen luotettavuutta.

Valitsin ohjelmistotestausta käsittelevän aiheen tämän pro gradu -tutkielmani aiheeksi, koska olen työskennellyt ohjelmistotestaajana erilaisissa testaukseen liittyvissä tehtävissä viimeiset 15 vuotta. Opintojeni aikana en kuitenkaan suorittanut yhtään kurssia ohjelmistotestauksesta, vaan olen oppinut kaikki tiedot ja taidot tekemällä käytännön töitä testajana, testausvastaavana ja testauskonsulttina. Olen jatkokouluttautunut itse mm. suorittamalla ISTQB perus- ja jatkotason testaussertifikaatit, ja lisäksi olen käynyt useita testausalan lyhytkoulutuksia. Uskon, että testausalan asiantuntemuksesta oli hyötyä kurssien sisältöanalyysin aikana, mikä myös parantaa tämän tutkimuksen luotettavuutta.

Tämän tutkimuksen luotettavuutta arvioidessa tulee lisäksi kiinnittää huomiota uskottavuuteen, siirrettävyyteen, riippuvuuteen ja vahvistettavuuteen. (Sarajärvi & Tuomi 2018: 161–163 [Lincoln & Cuba 1985]). Uskottavuudella tarkoitetaan sitä, että tulokset on kuvattu selkeästi ja ymmärrettävästi siten, että tutkimuksen kulku ja rajoitteet ovat selkeästi nähtävissä. Laadullisen tutkimuksen luotettavuutta parantaa aineiston keräämisen ja analysoinnin eri vaiheiden tarkka dokumentointi ja arviointi. Laadullisessa tutkimuksessa olennaista on luokittelujen tekeminen, ja tutkimuksen luotettavuutta parantaa myös näiden luokittelujen syntyminen ja tarkka kuvaus. (Hirsjärvi, Remes & Sajavaara 2014.) Tämän tutkimuksen osalta tutkimuksen kulkua ja rajoitteita kuvattiin Luvussa 3. Tutkimusaineiston luokittelu ja analyysi on kuvattu yksityiskohtaisesti tulosten esittelyn yhteydessä Luvussa 4. Lisäksi aineiston analyysiin liittyviä ongelmia kuvattiin edellä olevissa kappaleissa.

Siirrettävyydellä viitataan siihen, missä määrin tulokset voisivat olla siirrettävissä eri tutkimusympäristöön. Tässä tutkimuksessa käytettyä aineistolähtöistä sisällönanalyysia voitaisiin käyttää tutkimuksessa kuvatulla tavalla esimerkiksi, jos halutaan tutkia testauksen ope-
tusta jossakin muussa pohjoismaassa. Lisäksi tutkimusaineiston keruu voitaisiin toteuttaa samalla tavalla Suomessa uudestaan eri ajankohtana. Tämän tutkimuksen tutkimusaineisto koottiin suomalaisten korkeakoulujen lukuvuoden 2016–2017 opinto-oppaissa esiintyneistä

testausalan kursseista. Tutkimustulosten analyysi ja tutkielman loppuunsaattaminen kuitenkin viivästyivät kokopäiväisen työskentelyni takia. Kun nyt keväällä 2020 tätä pohdintaa kirjoittaessa tarkastelen valikoidusti uudestaan suomalaisten korkeakoulujen uusimpia opinto-oppaita näyttää siltä, ettei korkeakoulujen testausopetus ole kolmen vuoden aikana juurikaan muuttunut laajemmaksi tämän tutkimuksen aineistonkokoamisen ajankohtaan verrattuna.

Tutkimuksen riippuvuudella tarkoitetaan sitä, että tutkimus toteutetaan siten, kuin tieteellinen tutkimus vaatii. Tässä tutkimuksessa on pyritty noudattamaan laadullista tutkimusta ohjaavia periaatteita niin aineistonkeruussa kuin aineiston analyysissäkin. Tutkimuksen vahvistettavuudella tarkoitetaan sitä, että tutkimuksessa tehdyt valinnat ja ratkaisut on perusteltu, ja tutkimuksen tulokset saavat tukea muista tutkimuksista. Tässä tutkimuksessa valinnat perusteltiin tukeutuen käytettyihin lähdemateriaaleihin (SWEBOK-malli ja SE2014-opetussuositus). Lisäksi luvussa 3 perusteltiin miksi joitakin materiaaleja ei hyödynnetty. Kurssien luokittelua perusteltiin aineistosta nousseiden eri oppilaitosten kurssien välillä toistuneiden selkeiden teemojen avulla. Tehtyjä johtopäätöksiä pystyttiin tukemaan aiempien tutkimusten tutkimustuloksilla. Uskottavuuden, siirrettävyyden, riippuvuuden ja vahvistettavuuden kriteerit siis täyttyvät, joten tehtyä tutkimusta voidaan näiden kriteerien pohjalta kokonaisuudessaan pitää pääasiassa luotettavana.

6 Yhteenveto

Ohjelmointi ja ohjelmistotestaus ovat jatkuvasti kasvava toimiala, joka tarvitsee uusia tekijöitä. Siksi ohjelmistotestaajien koulutusta tulee kehittää vastaamaan työelämän muuttuneita tarpeita. Tässä pro gradu -tutkimuksessa kartoitettiin aineistolähtöisen sisällönanalyysin avulla mitä ohjelmistotestauksesta opetetaan nykyään suomalaisissa ammattikorkeakouluissa ja yliopistoissa, miten testauksen opetus vastaa kansainvälisiä ohjelmistotekniikan opetussuosituksia ja miten testauksen opetusta voitaisiin kehittää. Tutkimuksessa esille nousseista havainnoista ja ohjelmistotestauksen opetuksen puutteista koottiin lopuksi testauksen opetuksen kehittämisen ratkaisuehdotukseksi testausalasta kiinnostuneille opiskelijoille suunnattu ohjelmistotestauksen 25 opintopisteen laajuinen, korkeakoulutasoinen perusopintokokonaisuus, joka sisältää 7 ohjelmistotestausta käsittelevää kurssia.

Tämä tutkimus nosti esiin joukon erilaisia jatkotutkimusideoita. Tässä tutkimuksessa esiteltiin ohjelmistotestauksen perusopintokokonaisuus yhtenä ideana testauksen opetuksen kehittämiseksi. Perusopintokokonaisuutta ei kuitenkaan tämän tutkimuksen puitteissa toteutettu käytännössä, eikä sitä otettu käyttöön missään korkeakoulussa. Yhtenä jatkotutkimusideana voisikin olla joko yksittäisen kurssin toteuttaminen tai koko perusopintokokonaisuuden toteuttaminen käytännössä esimerkiksi kokonaan verkko-opintoina ja mahdollisuuksien mukaan yhteistyössä useamman korkeakoulun kanssa. Toteutusta voisi myös arvioida pedagogisesta näkökulmasta, ja opiskelijoilta voisi kerätä palautetta opintokokonaisuuden sisällöstä. Näin saataisiin kaksitasoinen tutkimus, jossa ensin toteutetaan ohjelmistotestauksen opetus ja sitten arvioidaan toteutusta eri näkökulmista.

Tämän tutkimuksen puitteissa ohjelmistotestauksen perusopintokokonaisuudelle ei myöskään luotu kurssien arviointikäytänteitä eikä kehitetty kursseilla käytettävää oppimateriaalia. Toisena jatkotutkimusideana voisi olla ohjelmistotestauksen perusopintokokonaisuuden kurssien arviointiperusteiden suunnitteleminen sekä erilaisten oppimateriaalien suunnitteleminen ohjelmistotestauksen perusopintokokonaisuuteen liitetyille kursseille, sekä näiden oppimateriaalien arvioiminen pedagogisesta näkökulmasta.

Tässä tutkimuksessa tutkittiin mitä ohjelmistotestauksesta opetetaan, mutta ei tutkittu, miten ohjelmistotestausta opetetaan. Neljäntenä jatkotutkimusideana voisi olla ohjelmistotestausta

opettavien opettajien haastattelu tai heidän opetuksensa seuraaminen, jotta saataisiin selvyttä siihen, miten ohjelmistotestausta opetetaan. Tässä kohtaa voisi myös tutkia opetuksen järjestämiseksi sitä, miten ohjelmistotestausta opetetaan kansainvälisesti, tai mitkä ovat testausalan työelämävaatimukset.

Yhtenä jatkotutkimusideana voisi myös olla ohjelmistotestauksen perusopintojen laajentaminen aineopinnoiksi. Aineopintoihin voisi ottaa mukaan tässä tutkimuksessa testauksen opetuksen yhdeksi puutteeksi havaitun suorituskykytestauksen, joka aihealueen edellyttämän erikoisosaamisen takia rajattiin perusopintokokonaisuuden ulkopuolelle. Lisäksi aineopintoihin voisi sisällyttää muita testauksen osa-alueita kuten erikoiskurssit hyväksymisestä ja www-sovellusten testaamisesta, tai testausmetriikoita käsittelevän kurssin. Aineopintokokonaisuus voisi sisältää myös yrittäjyyden, johtamisen ja ohjelmoinnin kurseja sekä ns. pehmeisiin taitoihin liittyviä kurseja, jotka käsittelevät esimerkiksi suullista ja kirjallista viestintää tai esiintymistaitoa.

Tässä tutkimuksessa pääasiallisena tavoitteena oli arvioida korkeakoulujen testausopetuksen nykytilaa sekä kehittää ohjelmistotestauksen opetusta. Ottamalla käyttöön tutkimuksessa kehitetty perusopintokokonaisuus sekä muita edellä mainittuja jatkotutkimusideoita toteuttamalla ohjelmistotestauksen opetusta voitaisiin kehittää lisää. Näin työelämään saataisiin paitsi motivoituneita ja osaavia ohjelmistotestaajia, myös koko testausammatti tulisi Suomessa tunnetummaksi.

Lähteet

Aalto-yliopisto 2017a. Koulutusohjelmat. <http://sci.aalto.fi/fi/studies/programmes/> [viitattu 26.3.2018]

Aalto-yliopisto 2017b. Master's Programme in Computer, Communication and Information Sciences Study Guide 2017–2018. <http://studyguides.aalto.fi/2017-ccis/> [viitattu 26.3.2018]

Abbors, F. 2015. Model-based testing of software systems: functionality and performance. Väitöskirja. Åbo Akademi University, Faculty of Science and Engineering, Software Engineering. <http://www.doria.fi/handle/10024/113081> [viitattu 2.2.2018]

Abran, A., Moore, J. W., Bourque, P., Dupuis, R. & Tripp, L. L. 2004. Guide to the Software Engineering Body of Knowledge: 2004 version. IEEE Computer Society. <http://opac.vimaru.edu.vn/edata/EBook/TNBook/TAI%20LIEU%20DIEN%20TU/1.%20A%20User%E2%80%99s%20Guide%20to%20Engineering/04425813.pdf> [viitattu 22.2.2018]

ACM 2018. Curricula Recommendations. <https://www.acm.org/education/curricula-recommendations> [viitattu 18.3.2018]

ACM & AIS 2017. MSIS 2016 Global Competency Model for Graduate Degree Programs in Information Systems. The Joint ACM/AIS MSIS 2016 Task Force. <https://www.acm.org/binaries/content/assets/education/msis2016.pdf> [viitattu 19.3.2018]

ACM & IEEE-CS 2015. Software Engineering 2014, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Joint Task Force on Computing Curricula, Computing Curricula Series. <https://www.acm.org/binaries/content/assets/education/se2014.pdf> [viitattu 18.3.2018]

ACM, AIS & IEEE-CS 2005. Computing Curricula 2005. The Overview Report. Joint Task Force for Computing Curricula, Computing Curricula Series. <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2005-march06final.pdf> [viitattu 18.3.2018]

ACM & IEEE-CS 2004. Software Engineering 2004, Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Joint Task Force on Computing Curricula, Computing Curricula Series. <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/se2004volume.pdf> [viitattu 18.3.2018]

Bloom, B. S. 1959. Taxonomy of Educational Objectives: The Classification of Educational Goals: Handbook I, Cognitive Domain. New York: McKay.

Bourque, P. & Fairley, R. E. (toim.) 2014. Guide to the Software Engineering Body of Knowledge, Version 3.0, IEEE Computer Society. <https://www.computer.org/web/swebok/v3> [viitattu 22.2.2018]

Capgemini 2011. Tutkimus: Pilvipalveluiden yhä kasvava käyttö lisää ohjelmistotestaus- ja laadunvarmistuspalveluiden kysyntää - Pohjoismaat jäljessä pilvipalveluiden testauksessa ja kehittämisessä. <https://www.fi.capgemini.com/news/tutkimus-pilvipalveluiden-yh-kasvava-kytt-lis-ohjelmistotestaus-ja-laadunvarmistuspalveluiden-kysynt> [viitattu 13.3.2018]

FiSTB 2020. Finnish Software Testing Board. <http://www.fistb.fi/fi/etusivu> [viitattu 21.5.2020]

FiSTB 2016. Uutiset. <http://www.fistb.fi/fi/uutiset> [viitattu 21.5.2020]

FiSTB 2015. ISTQB:n testaussanasto v. 2.3 Suomi - Englanti. Suomenkielinen käännös. http://www.fistb.fi/sites/fistb/files/liitteet/istqb_sanasto_2015-04-30%202.3%20FI-ENG.pdf [viitattu 27.3.2018]

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

Hanken 2017. Informationsbehandling. <https://www.hanken.fi/sv/om-hanken/organisation/institutioner-och-amnen/institutionen-foretagsledning-och-organisation-0> [viitattu 26.3.2018]

Heikkinen, E. 2015. Testing metrics in a software development company. Master of Science Thesis. Tampere University of Technology, Master's Degree Programme in Information Technology. <http://URN.fi/URN:NBN:fi:tty-201504221211> [viitattu 2.2.2018]

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2014. Tutki ja kirjoita. 19. painos. Helsinki: Tammi.

Högskolan på Åland 2016. Kursbeskrivningar HT2016.

<http://www.ha.ax/course.con?iPage=1169&m=&code=I380101> [viitattu 26.3.2018]

Hämeen ammattikorkeakoulu 2016. Opetussuunnitelma. <http://www.hamk.fi/hakijalle/Documents/TRTK16A.pdf> [viitattu 26.3.2018]

Iivonen, L. 2015. Ohjelmistotestauksen käytänteet Suomessa. Pro gradu -tutkielma. Turun yliopisto, Informaatioteknologian laitos, Tietojenkäsittelytiede. http://mars.cs.utu.fi/julkari/opetus/opinnaytteet/abstracts/2015_gradu_iivonen_lucilla.pdf [viitattu 2.2.2018]

ipetitions 2014. Stop 29119. <https://www.ipetitions.com/petition/stop29119> [viitattu 24.3.2018]

iSSEc 2009. Graduate Software Engineering 2009 (GSWE2009): Curriculum Guidelines for Graduate Degree Programs in Software Engineering. Stevens Institute of Technology Version 1.0. <https://www.acm.org/binaries/content/assets/education/gsew2009.pdf> [viitattu 19.3.2018]

ISTQB 2020a. About ISTQB®. <https://www.istqb.org/about-us.html> [viitattu 21.5.2020]

ISTQB 2020b. Vision & Mission. <https://www.istqb.org/about-us/vision-mission.html> [viitattu 21.5.2020]

ISTQB 2020c. Certifying Software Testers Worldwide. <https://www.istqb.org/> [viitattu 21.5.2020]

ISTQB 2020d. Why ISTQB® Certification? <https://www.istqb.org/certification-path-root/why-istqb-certification.html> [viitattu 21.5.2020]

ISTQB 2020e. ISTQB® Exam FAQs. Is there any validity period for this certification? <https://www.istqb.org/exam-root/exam-faqs.html?view=category&id=58> [viitattu 21.5.2020]

ISTQB 2020f. Foundation Level 2018. <https://www.istqb.org/certification-path-root/foundation-level-2018.html> [viitattu 21.5.2020]

ISTQB 2019a. Facts & Figures. <https://www.istqb.org/about-us/facts-figures.html> [viitattu 21.5.2020]

ISTQB 2019b. Certified Tester Foundation Level Syllabus. Version 2018 V3.1. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/2-foundation-level-documents.html> [viitattu 21.5.2020]

ISTQB 2019c. Certified Tester Specialist Syllabus Foundation Level Acceptance Testing. Version 2019. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/62-acceptance-testing.html> [viitattu 21.5.2020]

ISTQB 2019d. Certified Tester Specialist Syllabus Foundation Level Gambling Industry Tester. Version 2019. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/63-gambling-industry-tester.html> [viitattu 21.5.2020]

ISTQB 2019e. Certified Tester Specialist ISTQB® Mobile Application Testing Foundation Level Syllabus. Version 2019. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/61-mobile-application-testing.html> [viitattu 21.5.2020]

ISTQB 2019f. Certified Tester Advanced Level Syllabus Agile Technical Tester (ATT). Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/65-advanced-level-agile-technical-tester.html> [viitattu 21.5.2020]

ISTQB 2018a. Certified Tester Foundation Level 2018 Overview. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/2-foundation-level-documents.html> [viitattu 21.5.2020]

ISTQB 2018b. Certified Tester Foundation Level Specialist Syllabus Usability Testing. Version 2018. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/58-usability-testing-documents.html> [viitattu 21.5.2020]

ISTQB 2018c. Certified Tester Foundation Level Specialist Syllabus Performance Testing. Version 2018. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/59-performance-testing.html> [viitattu 21.5.2020]

ISTQB 2018d. Foundation Level Specialist CTFL® Automotive Software Tester. Version 2018. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/56-foundation-level-automotive-software-tester-documents.html> [viitattu 21.5.2020]

ISTQB 2017. ISTQB® Effectiveness Survey 2016-17. https://www.istqb.org/documents/istqb_survey_2016-2017.pdf [viitattu 21.5.2020]

ISTQB 2016a. Overview Advanced Level Syllabus Security Tester. Version 2016. Ladattavissa osoitteesta <https://www.istqb.org/downloads/category/46-advanced-level-security-tester.html> [viitattu 21.5.2020]

ISTQB 2016b. Certified Tester Advanced Level Syllabus Security Tester. Version 2016. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/46-advanced-level-security-tester.html> [viitattu 21.5.2020]

ISTQB 2016c. Certified Tester Advanced Level Syllabus Test Automation Engineer. Version 2016. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/48-advanced-level-test-automation-engineer-documents.html> [viitattu 21.5.2020]

ISTQB 2015a. ISTQB® Foundation Level Certified Model-Based Tester Overview. Version 2015. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/6-model-based-tester-extension-documents.html> [viitattu 21.5.2020]

ISTQB 2015b. ISTQB® Foundation Level Certified Model-Based Tester Syllabus. Version 2015. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/6-model-based-tester-extension-documents.html> [viitattu 21.5.2020]

ISTQB 2014a. Overview Foundation Level Extensions. Version 1.0. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/5-foundation-level-agile-tester.html> [viitattu 21.5.2020]

ISTQB 2014b. Certified Tester Foundation Level Extension Syllabus Agile Tester. Version 2014. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/5-foundation-level-agile-tester.html> [viitattu 21.5.2020]

ISTQB 2012a. Certified Tester Advanced Level Overview. Version 2012. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/73-2012-advanced-level-general-files.html> [viitattu 21.5.2020]

ISTQB 2012b. Certified Tester Advanced Level Syllabus Test Manager. Version 2012. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/70-2012-advanced-level-test-manager.html> [viitattu 21.5.2020]

ISTQB 2012c. Certified Tester Advanced Level Syllabus Test Analyst. Version 2012. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/72-2012-advanced-level-test-analyst.html> [viitattu 21.5.2020]

ISTQB 2012d. Certified Tester Advanced Level Syllabus Technical Test Analyst. Version 2012. Ladattavissa osoitteesta: <https://www.istqb.org/downloads/category/71-advanced-level-technical-test-analyst.html> [viitattu 21.5.2020]

ISTQB 2011a. Certified Tester Expert Level Syllabus Improving the Testing Process (Implementing Improvement and Change). Ladattavissa osoitteesta <https://www.istqb.org/downloads/category/12-expert-level-documents.html> [viitattu 21.5.2020]

ISTQB 2011b. Certified Tester Expert Level Syllabus Test Management (Managing Testing, Testers, and Test Stakeholders). Ladattavissa osoitteesta <https://www.istqb.org/downloads/category/12-expert-level-documents.html> [viitattu 21.5.2020]

Itkonen, J. 2011. Empirical studies on exploratory software testing. Väitöskirja. Aalto-yliopisto, Department of Computer Science and Engineering. Aalto University publication series, doctoral dissertations 107/2011. <http://urn.fi/URN:ISBN:978-952-60-4339-5> [viitattu 2.2.2018]

Itä-Suomen yliopisto 2015. Luonnontieteiden ja metsätieteiden tiedekunnan opinto-opas 2015–2016. http://www.uef.fi/documents/10184/67447/LuMet_Opinto-opas_2015-16_WWW.pdf/ca951a90-7f1c-4ed8-9c0e-ebe822c8a19c [viitattu 26.3.2018]

Itä-Suomen yliopisto 2016. Luonnontieteiden ja metsätieteiden tiedekunnan opinto-opas 2016–2017. http://www.uef.fi/documents/10184/1046728/LuMet_1617.pdf/d038b39fc759-4335-8a83-93c395d0b218 [viitattu 26.3.2018]

Jyväskylän ammattikorkeakoulu 2016a. Opinto-opas AMK. Insinööri (AMK), tieto- ja viestintäteknikan tutkinto-ohjelma 2016, 240 op. <https://opinto-opaat.jamk.fi/fi/opinto-opas-amk/tutkinto-ohjelmat-ja-opintotarjonta/suomenkieliset-opsit/2016-2017/tieto--ja-viestintateknikka/> [viitattu 26.3.2018]

Jyväskylän ammattikorkeakoulu 2016b. Opinto-opas AMK. Tradenomi (AMK), tietojenkäsittelyn tutkinto-ohjelma 2016, 210 op. <https://opinto-opaat.jamk.fi/fi/opinto-opas-amk/tutkinto-ohjelmat-ja-opintotarjonta/suomenkieliset-opsit/2016-2017/tietojenkäsittely/> [viitattu 26.3.2018]

Jyväskylän ammattikorkeakoulu 2016c. Opinto-opas YAMK. Master of Engineering, Degree programme in Information Technology, Cyber Security, year 2016, 60 credits. <https://opinto-opaat.jamk.fi/fi/Opinto-opas-yamk/tutkinto-ohjelmat-ja-opintotarjonta/englanninkieliset-opsit/2016-2017/information-technology/> [viitattu 26.3.2018]

Jyväskylän yliopisto 2016. Opetusohjelma – Informaatioteknologian tiedekunta – lukuvuosi 2016–2017. <https://korppi.jyu.fi/kotka/course/student/generalInfo.jsp?mainOrg1=1&department=2&beginyear=2016&buttonGet=Hae+opetusohjelma&language=2&include=4&schedule=2&statuses=2&statuses=3> [viitattu 26.3.2018]

Järviö, O. 2011. Testaustyökalujen kartoitus Ohjelmistotestaus opintojaksolle. Opinnäyte-työ. Oulun seudun ammattikorkeakoulu, Tietojenkäsittely. <http://www.theseus.fi/handle/10024/35384> [viitattu 2.2.2018]

Kajaanin ammattikorkeakoulu 2016a. Opinto-opas. Päivätoteutus. Tieto- ja viestintäteknikan koulutus, Älykkäät järjestelmät 2016. <http://opinto-opas.kamk.fi/index.php/fi/68146/fi/68097/TTV16SA/year/2016> [viitattu 26.3.2018]

Kajaanin ammattikorkeakoulu 2016b. Opinto-opas. Päivätoteutus. Tieto- ja viestintätekniikan koulutus, Peliteknologia 2016. <http://opinto-opas.kamk.fi/index.php/fi/68146/fi/68097/TTV16SP/year/2016> [viitattu 26.3.2018]

Kaleva, A. 2011. Testauskohde ohjelmistotestauksen opetukseen. Opinnäytetyö. Oulun seudun ammattikorkeakoulu, Liiketalouden yksikkö, Tietojenkäsittelyn koulutusohjelma. <https://oamk.finna.fi/Record/leevi.158035> [viitattu 2.2.2018]

Karjalainen, A., Lapinlampi, T., Jaakkola, E. & Alha, K. 2003. Opetussuunnitelman käsite. Teoksessa A. Karjalainen (toim.). Akateeminen opetussuunnitelmatyö. Oulun yliopisto, opetuksen kehittämissyksikkö. 26–55. <http://www oulu.fi/w5w/tyokalut/akatops305.pdf> [viitattu 13.3.2018]

Kasurinen, J. 2015. Designing an Infrastructure for a Web-Based Course in Software Testing: Preliminary results. Teoksessa S. Öystilä, P. Laine & J. Naukkarinen (toim.) Oppiva opettaja 14: Yliopistopedagogisen koulutuksen 2014 - 2015 opetuksen kehittämishankkeet. LUT Scientific and Expertise Publications. Lappeenrannan teknillinen yliopisto, Yliopistopalvelut. Raportit ja selvitykset – Reports 57. 100–117. <http://urn.fi/URN:ISBN:978-952-265-877-7> [viitattu 2.2.2018]

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

Kasurinen, J. 2011. Software test process development. Väitöskirja. Acta Universitatis Lappeenrantaensis 443. Lappeenranta University of Technology. <http://urn.fi/URN:ISBN:978-952-265-144-4> [viitattu 2.2.2018]

Korjus, N. 2015. Suomenkielisen oppimateriaalin koostaminen ohjelmistotestauksen perusteita käsittelevälle kurssille. Diplomityö, Lappeenrannan teknillinen yliopisto, School of Business and Management, Tietotekniikan koulutusohjelma. <http://urn.fi/URN:NBN:fi-fe201505218699> [viitattu 2.2.2018]

Koskela, A. 2014. Ketterän ohjelmistokehityksen ja kevyen käytettävyydestestauksen yhteensovittaminen: tapaustutkimus. Pro gradu -tutkielma. Jyväskylän yliopisto, Tietojenkäsittelytieteiden laitos, Tietojärjestelmätiede. <http://urn.fi/URN:NBN:fi:jyu-201406272171> [viitattu 2.2.2018]

- Kyttälä, L. 2011. Käytännön ohjelmointi -kurssin kehittäminen. Diplomityö. Lappeenranta teknillinen yliopisto, Teknistaloudellinen tiedekunta, Tietotekniikan osasto. <http://urn.fi/URN:NBN:fi-fe201110105695> [viitattu 2.2.2018]
- Lahti, M. 2014. Game Testing in Finnish Game Companies. Master's Thesis. Aalto University School of Science, Degree Programme in Computer Science and Engineering. https://aaltodoc.aalto.fi/bitstream/handle/123456789/15193/master_Lahti_Mikko_2015.pdf [viitattu 2.2.2018]
- Lempiäinen, T. 2015. Test automation on a third party mobile software. Diplomityö. Aalto University, School of Engineering, Department of Engineering Design and Production. https://aaltodoc.aalto.fi/bitstream/handle/123456789/16773/master_Lempiäinen_Tuomas_2015.pdf [viitattu 2.2.2018]
- Lindroos, S. 2014. Murtautumisympäristön rakentaminen opetuskäyttöön. Opinnäytetyö. Turun ammattikorkeakoulu, Tietojenkäsittely. https://www.theseus.fi/bitstream/handle/10024/75737/Saku_Lindroos.pdf [viitattu 2.2.2018]
- LUT 2017a. Weboodi, C-ohjelmoinnin ja testauksen periaatteet. <https://weboodi.lut.fi/oodi/opintjakstied.jsp?OpinKohd=23417172> [viitattu 26.3.2018]
- LUT 2017b. Tietotekniikan opintojen sisältö ja rakenne. <https://www.lut.fi/opiskelu/kandi-daattiohjelmat/tietotekniikka/opintojen-sisalto-ja-rakenne> [viitattu 26.3.2018]
- Marketvisio 2013. Ohjelmistotestauspalveluille löytyy kysyntää. <https://www.marketvisio.fi/fi/ajankohtaista/uutiset-marketvisio/1644-ohjelmistotestauspalveluille-l-ytyy-kysynt> [viitattu 30.5.2017]
- Marvin, R. 2014. The software testing schism. SDTimes. <https://sdtimes.com/applause/software-testing-schism/> [viitattu 24.3.2018]
- Metropolia 2016. Opinto-opas. Tieto- ja viestintäteknikka (Espoo), Koko opetussuunnitelma. <http://opinto-opas-ops.metropolia.fi/index.php/fi/88094/fi/70361/TXK16S1E/1739/year/2016> [viitattu 26.3.2018]

Myers, G. J., Badgett, T. & Sandler C. 2012. The Art of Software Testing. 3. painos. Hoboken, New Jersey: John Wiley & Sons.

Mynttinen, J. 2017. Tapaustutkimus Aalto-yliopiston, Lahden ammattikorkeakoulun ja Tampereen yliopiston ohjelmistotekniikan opintosisällöistä. Pro gradu -tutkielma. Tampereen yliopisto, Luonnontieteiden tiedekunta, Tietojenkäsittelytieteen tutkinto-ohjelma. <https://tampub.uta.fi/bitstream/handle/10024/101375/GRADU-1496834053.pdf> [viitattu 2.2.2018]

Opetushallitus 2018a. Ammattikorkeakoulut. http://www.oph.fi/koulutus_ja_tutkinnot/ammattikorkeakoulut_ja_yliopistot/ammattikorkeakoulut [viitattu 13.3.2018]

Opetushallitus 2018b. Yliopistot. http://www.oph.fi/koulutus_ja_tutkinnot/ammattikorkeakoulut_ja_yliopistot/yliopistot [viitattu 13.3.2018]

Oulun ammattikorkeakoulu 2016. Opinto-opas 2016–2017. <http://www.oamk.fi/opinto-opas/koulutusohjelmat/?opas=2016-2017> [viitattu 26.3.2018]

Oulun yliopisto 2016a. Weboodi, Tieto- ja sähkötekniikan tiedekunnan opinto-opas, Tietojenkäsittelytieteen tutkinto-ohjelma, 2016-17.

https://weboodi oulu.fi/oodi/jsp/opeopas/jsp/opastutkrak-kats.jsf?Kieli=1&MD5avain=&vl_tila=5&Opas=1741&Org=22&alusta=1&esikatselu=0 [viitattu 26.3.2018]

Oulun yliopisto 2016b. Weboodi, Usability Testing. <https://weboodi oulu.fi/oodi/opintjakstied.jsp?OpinKohd=24453354> [viitattu 26.3.2018]

Oulun yliopisto 2016c. Weboodi, Emerging Trends in Software Testing. <https://weboodi oulu.fi/oodi/opintjakstied.jsp?OpinKohd=24453348> [viitattu 26.3.2018]

Palomäki, P. 2016. Testaus ohjelmistotuotannon osana. Diplomityö. Tampereen teknillinen yliopisto, Johtamisen ja tietotekniikan koulutusohjelma. <http://URN.fi/URN:NBN:fi:tty-201708241704> [viitattu 2.2.2018]

Pohjolainen, P. A. 2007. The Development of Software Testing in Finland 1950–2000. Teoksessa J. Impagliazzo, T. Järvi & P. Paju (Eds) History of Nordic Computing 2. HiNC

2007. Conference proceedings, 271–282. Berlin: Springer. https://link.springer.com/chapter/10.1007/978-3-642-03757-3_28 doi: <https://doi.org/10.1007/978-3-642-03757-3> [viitattu 14.8.2017]
- Pyhäjärvi, M. 2014. ISO29119-testausstandardi tuli - heräsitkö sinäkin myöhään? <http://testausyhdistys.fi/2014/10/iso-29119-testausstandardi-tuli-herasitko-sinakin-myohaan/> [viitattu 24.3.2018]
- Reid, S. 2014. Response to Stop 29119 Petition. <http://www.softwaretestingstandard.org/29119petitionresponse.php> [viitattu 24.3.2018]
- Riihiaho, S. 2015. Experiences with usability testing: Effects of thinking aloud and moderator presence. Väitöskirja. Aalto-yliopisto, School of Science, Department of Computer Science. Aalto University publication series Doctoral Dissertations 75/2015. <http://urn.fi/URN:ISBN:978-952-60-6227-3> [viitattu 2.2.2018]
- Ruokoja, J. 2012. Tietojärjestelmän suorituskyvyn testaaminen. Pro gradu -tutkielma. Vaasan yliopisto, Teknillinen tiedekunta, Tietotekniikka. <https://www.tritonia.fi/fi/e-opinneytteen/tiivistelma/4684/Tietoj%C3%A4rjestelm%C3%A4n+suorituskyvyn+testaaminen> [viitattu 2.2.2018]
- Sainio, L. 2010. Ohjelmistotestauksen menetelmät ja työvälineet. Opinnäytetyö. Saimaan ammattikorkeakoulu, Tietotekniikan koulutusohjelma, Ohjelmistotekniikka. <http://urn.fi/URN:NBN:fi:amk-201003255929> [viitattu 13.8.2017]
- Salonen, V. 2012. Automatic portability testing. Pro gradu -tutkielma. Jyväskylän yliopisto, Tietotekniikan laitos. <http://urn.fi/URN:NBN:fi:jyu-201210212735> [viitattu 2.2.2018]
- Tekniikan maailma 2015. Kaikki älytelevisiosta – tv:n uudet vaatteet. <https://tekniikanmaailma.fi/kaikki-alytelevisiosta> [viitattu 13.3.2018]
- Tilastokeskus 2016. Toimialaluokitus 2008. 62 Ohjelmistot, konsultointi ja siihen liittyvä toiminta. <http://www.stat.fi/meta/luokitukset/toimiala/001-2008/62.html> [viitattu 13.3.2018]

Tivi 2013. Miksi testauskoulutus ei edisty? <http://www.tivi.fi/Uutiset/2013-03-03/Miksi-testauskoulutus-ei-edisty-3198557.html> [viitattu 13.3.2018]

Tivi 2010. Maailmalla voi valmistua ohjelmistotestaajaksi, Suomessa ei. <http://www.tivi.fi/Arkisto/2010-11-28/Maailmalla-voi-valmistua-ohjelmistotestaajaksi-Suomessa-ei-3181500.html> [viitattu 13.3.2018]

Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällönanalyysi. Tammi: Helsinki.

Tuominen, M. 2014 Graafisten käyttöliittymien regressiotestauksen automatisointi. Pro gradu-tutkielma. Jyväskylän yliopisto, Tietojärjestelmätiede. <http://urn.fi/URN:NBN:fi:jyu-201409262883> [viitattu 2.2.2018]

Turun yliopisto 2016. Opinto-oppaat. Matemaattis-luonnontieteellinen tiedekunta 2016–2018, Tulevaisuuden teknologioiden laitos. <https://nettiopsu.utu.fi/opas/laitos.htm?opsId=260&uiLang=fi&lang=fi&lvv=2016> [viitattu 26.3.2018]

Työ- ja elinkeinoministeriö 2016. Työaika, tietotyö ja tulevaisuus: esimerkkinä ohjelmistoala. Työ- ja elinkeinoministeriön julkaisuja 34. https://julkaisut.valtioneuvosto.fi/bitstream/handle/10024/75601/TEMrap_34_2016_netti.pdf?sequence=1 [viitattu 13.3.2018]

Vaasan ammattikorkeakoulu 2016. Opetussuunnitelma. <http://www.puv.fi/opsweb/?code=TT-2016> [viitattu 30.3.2018]

Vaasan yliopisto 2016a. Tekniikan kandidaatti, Energia- ja informaatiotekniikan ohjelma. https://www.univaasa.fi/fi/for/student/materials/handbooks/technology/tekniikan_kandidaatti-energia-ja_informaatiotekniikan_ohjelma.pdf [viitattu 26.3.2018]

Vaasan yliopisto 2016b. Weboodi, Ohjelmistotestaus. <https://weboodi.uwasa.fi/oodi/opintjakstied.jsp?OpinKohd=17960501> [viitattu 26.3.2018]

Vainio, N. 2017. Testauspolitiikan kehittäminen alihankintaprojekteihin. Diplomityö. Tampereen teknillinen yliopisto, Tietotekniikan diplomi-insinöörin tutkinto-ohjelma. <http://URN.fi/URN:NBN:fi:tyy-201705191441> [viitattu 2.2.2018]

Valtioneuvoston asetus ammattikorkeakouluista 1129/2014. Annettu Helsingissä 18.12.2014. <https://www.finlex.fi/fi/laki/alkup/2014/20141129> [viitattu 13.3.2018]

Valtioneuvoston asetus yliopistojen tutkinnoista 794/2004. Annettu Helsingissä 19.8.2004. <https://www.finlex.fi/fi/laki/alkup/2004/20040794> [viitattu 13.3.2018]

W5W2-hanke 2009. Näin asennat osaamistavoitteet opetussuunnitelmaasi, laaja oppimäärä. https://www2.uef.fi/documents/1526314/1526337/N%C3%A4in+asennat+osaamistavoitteet+opetussuunnitelmaasi-+laaja+oppim%C3%A4%C3%A4r%C3%A4_OY.pdf [viitattu 27.3.2018]

Åbo Akademi 2016. Moodle, Software Safety. <https://moodle.abo.fi/course/view.php?id=1691> [viitattu 26.3.2018]

Liitteet

A Ammattikorkeakoulujen ja yliopistojen testauskurssien luokittelu

Järjestäjä	Kurssin nimi	Laajuus	Pakolli- nen	Kategoria
Haaga-Helia	Vaatimusmäärittäminen ja vaatimuslähtöinen testaus	5 op	Ei	Vaatimuksiin perustuva testaus
HAMK	Testausprosessit	3 op	Kyllä	Testaus ja ohjelmistokehitys
JAMK	Ohjelmistosuunnittelu ja testaus	5 op	Kyllä	Vaatimuksiin perustuva testaus
JAMK	Järjestelmättestaus	4 op	Kyllä	Ohjelmistotestauksen perusteet
JAMK	Käyttöliittymä ja käytettävyys	4 op	Kyllä	Käytettävyystestaus
JAMK	Data Security Testing	3 op	Kyllä	Ohjelmistoturvallisuus
JAMK	Testing	3 op	Ei	Ohjelmistotestauksen perusteet
JAMK	Agile Testing	3 op	Ei	Ketterä testaus
JAMK	Auditing and Testing Technical Security	5 op	Kyllä	Ohjelmistoturvallisuus
JAMK	Web Application Security	5 op	Ei	Ohjelmistoturvallisuus
JAMK	Software Exploitation	5 op	Ei	Ohjelmistoturvallisuus
KAJAK	Ohjelmistokehitysprosessi, versionhallinta ja testaus	3 op	Kyllä	Vaatimuksiin perustuva testaus
LAMK	Software Testing and Maintenance	4 op	Kyllä	Ohjelmistotestauksen perusteet
LAMK	Usability Engineering	3 op	Kyllä	Käytettävyystestaus
Lapin AMK	Testauksen perustaidot	5 op	Kyllä	Ohjelmistotestauksen perusteet
Laurea	Ohjelmistotestaus	5 op	Ei	Ohjelmistotestauksen perusteet
Laurea	Käytettävyuden arviointi ja testaus	5 op	Ei	Käytettävyystestaus

Metropolia	Ohjelmistojen laatu	5 op	Ei	Ohjelmistojen laatu
Metropolia	Tietoturvallinen ohjelmistokehitys	5 op	Ei	Ohjelmistoturvallisuus
Metropolia	Pelaajakeskeinen suunnittelu ja pelitestaus	5 op	Ei	Pelitestaus
Metropolia	Software Quality	5 op	Ei	Ohjelmistojen laatu
Metropolia	Fundamental Practices for Secure Software Development	5 op	Ei	Ohjelmistoturvallisuus
OAMK	Ohjelmistotestaus	3 op	Kyllä	Ohjelmistotestauksen perusteet
OAMK	Laatujärjestelmät, testaus ja katselmointi	3 op	Ei	Ohjelmistojen laatu
OAMK	Tietoturvaohjelmointi	3 op	Kyllä	Ohjelmistoturvallisuus
OAMK	Usability Testing	3 op	Kyllä	Käytettävyystestaus
SAMK	Laadunhallinta	5 op	Kyllä	Ohjelmistotestauksen perusteet
TAMK	Ohjelmistotestaus	3 op	Kyllä	Ohjelmistotestauksen perusteet
TAMK	Playtesting and Usability	5 op	Kyllä	Pelitestaus
TURKU AMK	Pelitestaus	5 op	Ei	Pelitestaus
TURKU AMK	Information Security Testing and Assessment	5 op	Ei	Ohjelmistoturvallisuus
VAMK	Ohjelmistotestaus	5 op	Kyllä	Ohjelmistotestauksen perusteet
XAMK	Pelitestaus	5 op	Kyllä	Pelitestaus
XAMK	Game Testing II	5 op	Kyllä	Pelitestaus
XAMK	Pentesting	5 op	Kyllä	Ohjelmistoturvallisuus
ÅLAND	Testauslähtöinen ohjelmistokehitys	3 op	Ei	Testaus ja ohjelmistokehitys
Järjestäjä	Kurssin nimi	Laajuus	Pakollinen	Kategoria
Aalto	Software Testing and Quality Assurance	5 op	Ei	Ohjelmistotestauksen perusteet
Aalto	Information Security	5 op	Kyllä	Ohjelmistoturvallisuus
Aalto	Usability Evaluation	5 op	Kyllä	Käytettävyystestaus
Aalto/HY	Software Security	4 op	Ei	Ohjelmistoturvallisuus
HY	Ohjelmistoprosessit ja ohjelmistojen laatu	5 op	Ei	Ohjelmistojen laatu

HY	Ohjelmistojen testaus	5 op	Ei	Ohjelmistotestauksen perusteet
HY/Aalto	Software Security	4 op	Ei	Ohjelmistoturvallisuus
JYU	Ohjelmistotestaus	5 op	Ei	Ohjelmistotestauksen perusteet
JYU	Ohjelmistoturvallisuus	5 op	Kyllä	Ohjelmistoturvallisuus
JYU	Järjestelmähaavoittuvuudet	5 op	Ei	Ohjelmistoturvallisuus
LUT	C-ohjelmoinnin ja testauksen periaatteet	6 op	Kyllä	Ohjelmistotestauksen perusteet
OY	Software Quality and Testing	5 op	Kyllä	Ohjelmistotestauksen perusteet
OY	Emerging Trends in Software Testing	5 op	Kyllä	Testaustekniikat
OY	Usability Testing	5 op	Ei	Käytettävyystestaus
TTY	Ohjelmistojen testaus	5 op	Kyllä	Ohjelmistotestauksen perusteet
TTY	Eettinen hakkerointi	5 op	Kyllä	Ohjelmistoturvallisuus
TaY	Usability Evaluation Methods	5 op	Ei	Käytettävyystestaus
TaY	Open Source and Software Quality	5 op	Ei	Ohjelmistojen laatu
TaY	Testing, Security and Trust	5 op	Ei	Ohjelmistoturvallisuus
TY	Practical Agile Software Testing	5 op	Kyllä	Ketterä testaus
TY	Usability Testing	5 op	Ei	Käytettävyystestaus
TY	System and Application Security	5 op	Kyllä	Ohjelmistoturvallisuus
VY/VAMK	Ohjelmistotestaus	5 op	Ei	Ohjelmistotestauksen perusteet
ÅA	Software Testing	5 op	Ei	Ohjelmistotestauksen perusteet
ÅA	Software Safety	5 op	Ei	Ohjelmistoturvallisuus
ÅA	Software Quality	5 op	Ei	Ohjelmistojen laatu

B Ohjelmistotestauksen perusopin kokonaisuus

Kurssi	Laajuus	Esitiedot	AHOT-esimerkki
Ohjelmistotestaus	5 op	Ei mitään	ISTQB Certified Tester
Testausautomaatio	5 op	Ohjelmistotestaus	ISTQB Test Automation Engineer Python-osa: edX: Introduction to Python: Absolute Beginner TAI Learn to Program Using Python TAI Introduction to Computer Science and Programming Using Python
Ketterä ja tutkiva testaus	3 op	Ohjelmistotestaus	ISTQB Agile Tester
Testaustekniikat	3 op	Ohjelmistotestaus	ISTQB (Technical) Test Analyst
Käytettävyystestaus	3 op	Ohjelmistotestaus	ISTQB Usability Testing
Tietoturvatestaus	3 op	Ohjelmistotestaus	ISTQB Security Tester
Testauksen hallinta ja johtaminen	3 op	Ohjelmistotestaus Testausautomaatio	ISTQB Test Manager