

**Tapaustutkimus Aalto-yliopiston, Lahden ammattikorkeakoulun ja
Tampereen yliopiston ohjelmistotekniikan opintosisällöistä**

Jarno Mynttinen

Tampereen yliopisto
Luonnontieteiden tiedekunta
Tietojenkäsittelytieteiden tutkinto-ohjelma
Pro gradu -tutkielma
Ohjaajat: Erkki Mäkinen ja Timo T. Poranen
Toukokuu 2017

Tampereen yliopisto

Luonnontieteiden tiedekunta

Tietojenkäsittelytieteiden tutkinto-ohjelma

Jarno Mynttinen: Tapaustutkimus Aalto-yliopiston, Lahden ammattikorkeakoulun ja

Tampereen yliopiston ohjelmistotekniikan opintosisällöistä

Pro gradu -tutkielma, 50 sivua

Toukokuu 2017

Tutkielmassa haluttiin selvittää, miten ohjelmistotekniikan opetus Suomessa vastaa kansainvälisiä opetussuosituksia. Viimeisimmät ohjelmistotekniikan kansainväliset suositukset on laadittu 2000-luvun aikana. Opetussuositukset ovat annettu erikseen alemmalle ja ylemmälle korkeakoulututkinnolle. Molemmissa suosituksissa on määriteltynä tärkeät tiedot, jotka valmistuvan opiskelijan tulee hallita. Näitä tietoja kutsutaan tietoalueiksi. Suosituksissa jokaiselle tietoalueelle on määriteltynä, kuinka paljon aikaa vähintään niiden opetukseen tulisi käyttää opintojen aikana.

Tutkimus toteutettiin tapaustutkimuksena, jossa tutkimukseen valitun kolmen korkeakoulun opinto-oppaiden sisältö luokiteltiin suositusten tietoalueisiin. Luokittelun tuloksia vertailtiin suositusten ajan osuuksiin ja selvitettiin, onko oppilaitosten opintojen painotus samalla tasolla kuin suosituksissa.

Tutkimuksessa havaittiin, että opetuksessa on puutteita. Kaikissa oppilaitoksissa jää vähintään kaksi tai jopa viisi suositusten tietoalueista ilman opetusta. Kun luokittelun tulokset kootaan yhteen, niin havaitaan, että alemmissa korkeakoulututkinnoissa ohjelmiston laadunvarmistamisen opetus jää vähälle huomiolla. Ylemmissä korkeakoulututkinnoissa selkeästi ilman huomiota jää ohjelmiston ylläpito ja konfiguraatiohallinta.

Koulutuksessa on selkeästi kehitettävää kansallisella tasolla. Hyvä lähtökohta kehitykselle on jokaisen oppilaitoksen omien tutkintojen parantamisella. Tällä tavalla pystytään varmistamaan, että opiskelijat saavat Suomessa parasta mahdollista koulutusta myös tulevaisuudessa.

Avainsanat ja -sanonnat: ohjelmistotekniikka, kansainväliset opetussuositukset, opinto-opas

Sisällys

1. Johdanto.....	1
2. Taustaa	3
2.1. Ohjelmistokehityksen tietopohja	4
2.2. Ohjelmistotekniikan opintosuosituksset	5
2.2.1. Ensimmäiset opintosuosituksset	5
2.2.2. Alemman korkeakoulututkinnon suositukset	6
2.2.3. Ylemmän korkeakoulututkinnon suositukset	11
2.3. Ohjelmistotekniikan opetus Suomessa	15
2.4. Opinto-oppaat	15
2.4.1. Tampereen yliopisto	16
2.4.2. Aalto-yliopisto	18
2.4.3. Lahden ammattikorkeakoulu	21
3. Tutkimuksen toteutus	23
3.1. Tutkimusmenetelmälliset valinnat ja aineisto	23
3.2. Luokittelu	24
4. Analyysi	26
4.1. Tampereen yliopiston tietojenkäsittelytieteen kandidaatin opinto-opas	26
4.2. Tampereen yliopiston filosofian maisterin opinto-opas	28
4.3. Aalto-yliopiston tekniikan kandidaatin opinto-opas	30
4.4. Aalto-yliopiston diplomi-insinöörin opinto-opas	32
4.5. Lahden ammattikorkeakoulun insinöörin (AMK) opinto-opas.....	34
4.6. Luokittelun koonti	36
5. Yhteenveto	40
5.1. Yhteenveto tutkimuksen tuloksista.....	40
5.2. Tutkimuksen luotettavuus.....	43
6. Pohdinta.....	45
Viiteluettelo.....	48

1. Johdanto

Ohjelmistoja on nykyisin lähes kaikissa tuotteissa ja järjestelmissä. Ohjelmistot ohjaavat esimerkiksi ydinvoimaloiden, ajoneuvojen, lentokoneiden ja älypuhelimien toimintaa. Ajoneuvoissa ohjelmistot voivat olla vastuussa lukkiutumattomien jarrujen toiminnasta, polttoaineen syötöstä moottoriin sekä ääniohjautuvasta navigoinnista [5]. Yhdelle koulutukselle on haastavaa valmistaa opiskelija kaikkeen tähän tarjontaan, joten on luotu opetussuosituksia varmistamaan se, että ohjelmistotekniikan koulutus tarjoaa riittävät tiedot ja taidot työelämässä pärjäämiseen.

Tässä tutkielmassa tarkastellaan kolmen erilaisen suomalaisen korkeakoulun ohjelmistotekniikan opetusta ja vertaillaan niitä IEEE:n ja ACM:n työryhmien esittämiin opetussuosituksiin. Korkeakoulut ovat valittu edustamaan erilaisia tapoja, joilla Suomessa voi saada ohjelmistotekniikan koulutusta. Mukaan on valittu esimerkki monialayliopistoista, Tampereen yliopisto, teknillisistä yliopistoista, Aalto-yliopisto, sekä ammattikorkeakouluista, Lahden ammattikorkeakoulu.

Tampereen yliopisto edustaa monialayliopistona suurinta osaa yliopistoja Suomesta. Aalto-yliopiston koulutus on laajuudeltaan yhtä suurin kuin Tampereen yliopiston, mutta historiansa vuoksi edustaa erityyppistä teknillistä yliopistoa. Kolmas oppilaitos, Lahden ammattikorkeakoulu, ei ole laajuudeltaan yhtä suurin kuin kaksi edellistä, mutta sieltä voi erikoistua juuri ohjelmistotekniikkaan.

Puhuttaessa Suomessa ohjelmistotekniikasta (software engineering) ja sen opetuksesta tarkoitetaan sillä usein koulutuslinjaa, josta valmistutaan insinööriksi. Monialayliopistoissa pyritäänkin tästä syystä välttämään ohjelmistotekniikka-termiä ja käytetään mieluummin ohjelmistotuotanto-termiä. Tässä tutkielmassa puhuttaessa ohjelmistotekniikasta ja ohjelmistotekniikan opetussuosituksista tarkoitetaan sillä kuitenkin koulutusta ja suosituksia, jotka ovat tarkoitettu myös monialayliopistoille, joista valmistutaan maisteriksi.

Tutkielmassa pyritään vastaamaan seuraaviin tutkimuskysymyksiin:

1. Miten ohjelmistotekniikan opetus Suomessa vastaa alan kansainvälisiä opetussuosituksia?
2. Missä tutkittavista oppilaitoksista alemman korkeakoulututkinnon opetus vastaa parhaiten opetussuosituksia?

3. Missä tutkittavista oppilaitoksista ylemmän korkeakoulututkinnon opetus vastaa parhaiten opetussuosituksia?

Toista tutkimuskysymystä voidaan tarkastella myös kahdesta näkökulmasta. Ensimmäinen näkökulma on, missä oppilaitoksessa alemman korkeakoulututkinnon opetusmateriaali vastaa parhaiten opetussuosituksia. Toinen näkökulma on sama, mutta ylemmän korkeakoulututkinnon suosituksia.

Tutkimuksessa havaittiin, että opetuksessa on puutteita. Kaikissa oppilaitoksissa jää vähintään kaksi tai jopa viisi suositusten tietoa-alueista ilman opetusta. Kun luokittelun tuloksen kootaan yhteen, niin havaitaan, että alemmissa korkeakoulututkinnoissa ohjelmiston laadunvarmistamisen opetus jää vähälle huomiolla. Ylemmissä korkeakoulututkinnoissa selkeästi ilman huomiota jää ohjelmiston ylläpito ja konfiguraatiohallinta. Koulutuksessa on selkeästi kehitettävää kansallisella tasolla. Hyvä lähtökohta kehitykselle on jokaisen oppilaitoksen omien tutkintojen parantamisella. Tällä tavalla pystytään varmistamaan, että opiskelijat saavat Suomessa parasta mahdollista koulutusta myös tulevaisuudessa.

Aluksi esittelen tutkimuksen taustaa käymällä läpi ohjelmistotekniikan historiaa sekä perehtymällä siihen, mitä ohjelmistotekniikka on. Avaan kansainvälisiä opetussuosituksia, jotka ohjelmistotekniikan opetukselle on annettu. Tämän jälkeen tarkastelen kolmen suomalaisen korkeakoulun opinto-oppaiden sisältöjä sekä kyseisten oppilaitosten opintojen rakennetta. Näiden jälkeen analysoin aineistoa luokittelemalla opinto-oppaiden kursseja suosituksista löytyvien tietoa-alueiden perusteella. Lopuksi perehdyn luokittelun tuloksiin ja teen niiden perusteella johtopäätöksiä. Viimeiseksi pohdin tutkimukseni tuloksia ja ehdotan mahdollisia jatkotutkimus aiheita.

2. Taustaa

Ohjelma on joukko tietokoneen käskyjä, jotka suorittamalla päästään haluttuun päämäärään. Ohjelmia on ollut niistä ajoista asti, kun ensimmäiset tietokoneet on kehitetty vuodesta 1944 eteenpäin [19]. Termiä ohjelmisto (software) käytettiin kuitenkin ensimmäistä kertaa vasta vuonna 1958 [26]. Kymmenen vuotta myöhemmin NATO:n konferenssissa käytettiin termiä ohjelmistotekniikka (software engineering) [20]. Ensimmäiset ohjelmistotekniikkaan liittyvät julkaisut julkaisi IEEE Computer Society vuonna 1972 [15]. Ensimmäinen komitea, jonka tarkoituksena oli ajaa ohjelmistotekniikan kehittämistä, perustettiin IEEE Computer Society'n alaisuuteen vuonna 1976 [15]. Kuitenkin vasta vuonna 1979 muodostettiin ensimmäinen kokonaisvaltainen näkemys siitä, mitä ohjelmistotekniikka on. Tällöin Fletcher Buckley alkoi kehittää standardia ohjelmiston laadunvarmistukselle [17].

Vuosien 1981–1985 aikana IEEE Computer Society piti useita työpajoja, joissa pohdittiin ohjelmistotekniikan standardeja. Työpajojen tuloksena syntyi vuonna 1986 ohjelmistotekniikan taksonomian standardi, joka tarjosi kokonaisvaltaisen näkökulman ohjelmistotekniikkaan [16]. Vuonna 1993 Fletcher Buckley teki aloitteen, jonka tarkoituksena oli luoda raamit yleisesti hyväksytyille tiedolle ohjelmistotekniikasta. IEEE Computer Society'n johtokunta hyväksyi aloitteen toukokuussa 1993 ja Association for Computer Machinery'n (ACM) neuvosto hyväksyi sen saman vuoden elokuussa. Aloitteesta syntyi lopulta julkaisu Software Engineering Body of Knowledge eli SWEBOK [3]. Ensimmäinen vedos julkaistiin vuonna 1999 [10]. Ensimmäinen virallinen versio on julkaistu vuonna 2004 [3], jota on päivitetty viimeksi vuonna 2014 [11].

Kun vuonna 2001 aloitettiin Software Engineering Education Knowledge (SEEK) kehittäminen, niin lähtökohdaksi otettiin SWEBOK:n alustavat aihealueet [12]. SEEK on osa IEEE:n ja ACM:n projektia, jossa oli tarkoituksena muodostaa opintosuosituksia tietokonealalle yleisesti. Hyvin nopeasti työn aloittamisen jälkeen todettiin, ettei suosituksia pysty yleistämään koko tietokonealalle, vaan kukin ala tarvitsee oman työryhmänsä kehittämään suosituksiansa [25]. Yksi näistä työryhmistä lähti kehittämään ohjelmistotekniikalle suosituksia, joista muodostui SEEK.

IEEE ja ACM ovat tehneet yhteistyötä opintosuositusten työstämisessä vuodesta 1991 lähtien [4], jolloin julkaistiin yleiset suositukset tietojenkäsittelytieteelle ja sen läheisille aloille. Vuonna 1998 IEEE:n ja ACM:n yhteinen työryhmä aloitti työn uusien suositusten työstämiseksi ja päivittämiseksi nykyai-

kaan. Tämän työn tuloksena syntyivät suositukset ohjelmistotekniikan opiskelun aloittaneille, jotka julkaistiin vuonna 2004 [25]. Vuonna 2007 akateemisen maailman, työelämän, julkisen hallinnon ja ammatillisten järjestöjen yhteenliittymä aloitti työn, jonka tarkoituksena oli luoda opintosuositukset ohjelmistotekniikan maasteritason koulutukselle [5]. Projektin tulokset julkaistiin vuonna 2009.

2.1. Ohjelmistokehityksen tietopohja

Software Development Body of Knowledge (SWEBOK) on kehitetty määrittelemään, mitä ohjelmistotekniikka on. SWEBOK:n kehitys alkoi vuonna 1997 tarkoituksena luoda yhteisymmärrys yleisesti hyväksytylle tiedolle ohjelmistotekniikasta [10]. SWEBOK on ensimmäinen kokonaisvaltainen ohjelmistotekniikan ydintiedon ja -taidon määritelmä. Kehitystyön jatkuessa projektille asetettiin viisi tavoitetta, jotka ovat säilyneet nykyisiin versioihin asti [10]. Nämä tavoitteet ovat:

1. Luonnehtia ohjelmistotekniikan sisältöjä.
2. Tarjota pääsy ohjelmistotekniikan tietopohjaan.
3. Ylläpitää maailmanlaajuisesti yhtenäistä kantaa ohjelmistotekniikkaan.
4. Selventää ja asettaa rajat ohjelmistotekniikalle ottaen huomioon muut tieteenalat kuten tietotekniikka, projektihallinta, tietokonetekniikka ja matematiikka.
5. Tarjota perusta opetussuunnitelmien kehitykselle ja yksittäisten sertifiointien materiaalille.

Ensimmäinen versio SWEBOK:sta julkaistiin vuonna 2004 [3] ja viimeisin versio on julkaistu vuonna 2014 [11].

Alkuperäinen SWEBOK koostui kymmenestä tietoaalueesta [3]. Uusimmassa julkaisussa tietoaalueita on 15 kappaletta. Tietoaalueet on jaettu vielä tarkemmin aiheisiin, jotka kuvaavat ymmärrettävästi tietoaalueen sisältöjä.

Oppaassa tunnustetaan tietotekniikan läheinen yhteys muihin tieteenaloihin, joiden tietoihin ja taitoihin ohjelmistokehittäjien tulisi kiinnittää huomiota. Oppaassa ei kuitenkaan pyritä kuvaamaan näiden alojen tietopohjaa. Läheisesti yhteydessä tietotekniikan kanssa olevat tieteenalat ovat [10]:

1. Tietokonetekniikka
2. Tietojenkäsittelytiede
3. Yleinen johtaminen

4. Matematiikka
5. Projektihallinta
6. Laadunhallinta
7. Järjestelmäteknikka.

SWEBOK on merkittävä julkaisu, koska sitä on käytetty pohjana luotaessa uusia opetussuosituksia. SWEBOK:n julkaisun jälkeen sekä alemman että ylemmän korkeakoulututkinnon kansainväliset suositukset perustuvat SWEBOK:n tietoi-alueisiin [5, 25]. Niitä ei kuitenkaan ole otettu suoraan SWEBOK:sta, sillä koulutuksen kannalta se on liian laaja, joten kaikkia tietoi-alueita ei ole otettu suoraan siitä.

2.2. Ohjelmistotekniikan opintosuositukset

ACM on tarjonnut opintosuosituksiaan tietojenkäsittelytieteelle vuodesta 1968 lähtien [9]. Ensimmäisen kerran suosituksia ohjelmistotekniikalle julkaisi Carnegie Mellon yliopisto vuonna 1988 [8]. Näitä suosituksia käytettiin pitkään tärkeimpänä lähteenä opinto-oppaiden kehittämiseen [7]. Vuonna 2004 julkaistu ohjelmistotekniikan opetusohjelma Computing Curricula - Software Engineering (CCSE) suositukset ovat tuoreimmat suositukset alemmalle korkeakoulututkinnolle [25]. Pian CCSE:n julkaisun jälkeen aloitettiin työ myös ohjelmistotekniikan ylemmän korkeakoulututkinnon opetussuosituksien päivittämiseksi nykyaikaan. Tämän työn tuloksena syntyi Graduate Software Engineering 2009 (GSWE2009) [5].

2.2.1. Ensimmäiset opintosuositukset

Carnegie Mellon yliopiston Software Engineering -instituutin (SEI) opintosuositukset ovat ensimmäiset ohjelmistotekniikan maisteritutkinnolle kehitetyt suositukset, joita on käytetty laajasti eri oppilaitoksissa ohjelmistotekniikan tutkinto-ohjelmien rakentamiseen [5]. SEI:n suositukset julkaistiin vuonna 1989 ja niitä päivitettiin vuonna 1991, mutta ne säilyivät käytännössä alkuperäisessä muodossaan [5].

SEI:n opintosuositukset rakentuvat kuuden ydinkurssin ympärille. Nämä kurssit käsittivät tärkeimmät tiedot ja taidot, jotka opiskelijoiden täytyy oppia. Tämän lisäksi tutkintoon kuuluu kaksi tai kolme valinnaista syventävää kurssia, jotka oppilaitokset saavat itse määrittää. Näiden kurssien lisäksi suositukseen sisältyi vielä ohjelmistotekniikan projekti.

SEI:n suositukset olettavat tutkinto-ohjelmaan osallistuvalla opiskelijalla olevan kandidaatintutkinto tietojenkäsittelytieteestä tai sitä vastaava määrä

työkokemusta. Lisäksi opiskelijalla tulisi olla taustatietoja diskreetistä matematiikasta, todennäköisyyslaskennasta ja tilastotieteestä. SEI:n suositusten mukaan opiskelijalta ei vaadita työkokemusta, jos hänellä on olemassa alempi korkeakoulututkinto. Työkokemusta pidetään tärkeänä, mutta sitä ei haluta vaatia. On olemassa useita muita menestyksekkäitä insinööritieteiden maisteriohjelmiä, joissa työkokemusta ei vaadita, joten näissäkään suosituksissa sitä ei vaadita [8].

2.2.2. Alemman korkeakoulututkinnon suositukset

Vuoden 1998 syksyllä IEEE:n ja ACM:n yhteinen komissio käynnisti projektin, jonka tarkoituksen oli tehdä katsaus sen hetkisiin tietokonealan alemman korkeakoulutason opetussuosituksiin. Projekti, jonka nimi oli Computing Curricula, jaettiin neljään osaan eri tieteenalojen mukaan: tietojenkäsittelytiede, tietokonetekniikka, ohjelmistotekniikka ja tietojärjestelmä. Ohjelmistotekniikan työryhmän tuloksena syntyivät opintosuositukset, joille annettiin nimi Computing Curricula - Software Engineering (CCSE) [25].

CCSE rakennettiin 11 periaatteen pohjalta, jotka tuli kehitystyön aikana ottaa huomioon:

1. CCSE keskittyy ohjelmistotekniikan tietoon ja pedagogiikkaan, mutta tietokonealan rajat ovat epäselviä, joten muilta aloilta on haettava opastusta tarpeen vaatiessa.
2. Ohjelmistotekniikka perustuu voimakkaasti moneen tieteenalaan, kuten tietojenkäsittelytieteeseen, matematiikkaan, insinööritaitoihin sekä projektijohtamiseen. Opiskelijan on opittava tietoja ja taitoja myös näistä aloista.
3. Ohjelmistotekniikan nopean kehityksen johdosta opetussuositukset tulee rakentaa siten, että sen osakokonaisuuksia pystytään päivittämään helposti.
4. Opetussuositusten kehittämisen pitää pystyä mukautumaan teknologian, käytäntöjen ja ohjelmistojen nopeisiin muutoksiin pedagogisen kehityksen ja elinikäisen oppimisen tärkeyden vuoksi.
5. CCSE täytyy rakentaa siten, että sen osakokonaisuuksista tulee järkeviä, jotta opetuskokonaisuudet ovat helposti toteutettavissa sekä tiedon jakaminen oppilaitosten välillä tapahtuu vaivattomasti.
6. CCSE täytyy tukea perustavanlaatuisen tietojen ja taitojen tunnistamista, jotka valmistuvan opiskelijan täytyy hallita.

7. Opintosuosituksia tulee perustua yleisesti hyväksytyyn ohjelmistotekniikan tiedon määritelmään.
8. CCSE tulee pyrkiä olemaan mahdollisimman kansainvälisesti käytettävä.
9. Kehityksen tulee tapahtua mahdollisimman laajasti. Kehitystyössä mukana tulisi olla mahdollisimman paljon eri taustaisia ihmisiä.
10. CCSE tulee sisältää näkökulmia ohjelmistotekniikan työn suorittamiseen.
11. CCSE tulee sisältää eri strategioita ja taktiikoita suositusten toteuttamiseen.

Yllä mainitun seitsemän periaatteen perusteella on CCSE työn ohella tuotettu SEEK. SEEK:n tärkein tehtävä on määrittää ohjelmistotekniikan koulutukselle ydintietoalueet ja niiden sisältö. SEEK kehitettiin samoihin aikoihin kuin SWEBOOK, ja nämä kaksi työryhmää tekivätkin yhteistyötä kehitystyön aikana [12]. Yhteistyön perusteella todettiin, että SWEBOOK on suunnattu valmistuneille ja usean vuoden työelämässä olleille ohjelmistokehittäjille, kun taas SEEK on suunnattu opiskelun vasta aloittaneille. SEEK pohjautuu kuitenkin voimakkaasti SWEBOOK:n rakenteeseen, mutta SEEK:n kehityksessä on otettu huomioon vanhat SEI:n opintosuosituksia (alakohta 2.2.1). Toisin kuin SWEBOOK, SEEK pyrkii määrittämään myös läheisten tieteenalojen aihealueita, jotka valmistuvan opiskelijan tulee hallita.

Valmistuvan opiskelijan täytyy hallita seuraavat asiat:

1. Työskennellä osana ryhmää, joka kehittää ja toteuttaa toimivia artefakteja.
2. Ymmärtää prosessi, jossa määritellään asiakkaan tarpeet ja niiden kääntäminen ohjelmiston vaatimuksiksi.
3. Sovitella ristiriitaisia päämääriä ja löytää niille hyväksyttäviiä kompromisseja hinnan, ajan, tiedon, olemassa olevan järjestelmän ja organisaation puitteissa.
4. Suunnitella tarkoituksenmukaisia ratkaisuja yhdelle tai usealle sovel- lusalueelle ottaen huomioon eettiset, sosiaaliset, lailliset sekä taloudelliset aiheet.
5. Ymmärtää ja kyetä käyttämään nykyisiä teorioita, malleja ja tekniikoita, jotka mahdollistavat pohjan ohjelman suunnitteluun, kehittämiseen, toteuttamiseen ja verifiointiin.

6. Osoittaa ymmärtävänsä ja arvostavansa neuvottelun, tehokkaiden työskentelytapojen, johtamisen ja hyvien kommunikaatiotaitojen tärkeys ohjelmistokehitysympäristössä.
7. Oppia uusia malleja, tekniikoita ja teknologioita sitä mukaan kun niitä ilmaantuu.

Ohjelmistotekniikan nopean jatkuvan kehittymisen vuoksi CCSE työryhmä on määrittänyt ydinosamisalueet, jotka kaikki tutkinnot tulisi vähintään täyttää. Ydinosamisalueet ovat määritelty mahdollisimman pieneksi, jotta oppilaitoksilla olisi mahdollisuus rakentaa haluamansa tutkinnot. Pelkkien ydinosamisalueiden sisällyttäminen tutkintoon ei riitä, vaan tutkintoon pitää kuulua myös valinnaisia opintoja, jotka oppilaitos voi itse määrittää.

CCSE on jaettu SEEK:n mukaisesti kymmeneen tietoalueeseen, jotka on jaettu pienempiin yksikköihin ja yhä edelleen aiheisiin. Kunkin aiheen tärkeys on merkitty kolmiasteisesti: oleellinen, toivottava ja valinnainen. CCSE tarjoaa jokaiselle yksikölle aika-arvion tunneissa, kuinka kauan kestää vähintään yksikön tiedon opettaminen. Aika-arvio perustuu perinteiseen luentopohjaiseen opetukseen eikä se sisällä itsenäisesti tehdyn työn määrää tai muita kurseihin liittyvää läsnäoloa. SEEK:n mukaisesti valitut tietoalueet ovat:

1. Ohjelmistojen ydinkohdat
2. Matemaattinen ja tekninen perusta
3. Ammatillinen käytäntö
4. Ohjelmiston mallinnus ja analyysi
5. Ohjelmiston suunnittelu
6. Ohjelmiston verifiointi ja validointi
7. Ohjelmiston kehityskaari
8. Ohjelmiston prosessit
9. Ohjelmiston laatu
10. Ohjelmiston hallinta.

CCSE:n suositusten mukaan kaiken materiaalin opetukseen vaaditaan vähintään 494 tuntia opetusta. Eniten tunteja vaativat ohjelmistojen ydinkohdat -tietoalue, johon kuuluu 172 tuntia opetusta, joka on noin kolmasosa koko opetusmäärästä. Ohjelmistojen ydinkohdat -tietoalue on jaettu neljään yksikköön, joista ensimmäiseen yksikköön eli tietojenkäsittelytieteen perusteisiin kuuluu 140 tuntia suunnitellusta ajasta. Tietojenkäsittelytieteen perusteisiin sisältyy ohjelmoinnin peruseriaatteet, tietokannat ja tietorakenteet. Lähes kaikki oh-

jelmistojen ydinkohdat tietoaalueeseen aiheet ovat suositusten mukaan oleellisia ohjelmistotekniikan kannalta.

Toiseksi eniten aikaa on varattu matemaattiseen ja tekniseen perustaan, johon sisältyy 89 tuntia opetusta. Matemaattinen ja tekninen perusta on jaettu kolmeen yksikköön, joista matemaattiset perusteet yksikköön kuuluu suurin osa ajasta. Tekniseen perustaan lähestytään tässä yksikössä ohjelmistojen näkökulmasta. Opetettaviin aiheisiin sisältyy tilastollista analyysia, järjestelmän kehitystä ja sen skaalautuvuutta sekä kustannustehokkuuden arviointia.

Ohjelmiston mallinnus ja analyysi -tietoaalueeseen on tarkoitus käyttää kolmanneksi eniten aikaa; siihen kuuluu 53 tuntia opetusta. Ohjelmiston mallinnus ja analyysi -tietoaalueen analyysiosio pitää sisällään vaatimusten perusteita, määrittelyä ja validointia.

Seuraavaksi eniten aikaa kuluu ohjelmistojen suunnittelu -tietoaalueeseen, johon on varattu 45 tuntia opetusta. Ohjelmistojen suunnittelu koostuu kuudesta yksiköstä, joista kahden yksikön osuus on 12 tuntia kummallekin. Toinen yksiköistä käsittelee käyttöliittymien suunnittelua. Ensimmäinen yksikkö käsittelee taas yksityiskohtaista suunnittelua, johon kuuluu suunnittelumalleja, -metodeja ja -notaatiota.

Viidenneksi eniten aikaa on varattu ohjelmistojen verifiointi ja validointi -tietoaalueeseen, johon kuluu 42 tuntia opetusta. Tämä tietoaalue koostuu viidestä yksiköstä, joista testaus kattaa puolet tunneista. Testaus yksikössä käsitellään erilaisia testausmalleja integraatiotestauksesta regressiotestaukseen sekä erilaisia testaustyökaluja, kuten yksikkötestaus. Testauksesta on eroteltu omaan yksikköönsä käyttöliittymätestaus evaluointi, joka sekin käsittää kuusi tuntia opetusta.

Ammatillinen käytäntö -tietoaalueeseen on tarkoitus käyttää kuudenneksi eniten aikaa; siihen kuuluu 35 tuntia opetusta. Tämä tietoaalue koostuu kolmesta yksiköstä, joihin sisältyvät ryhmäkäyttäytyminen ja psykologia, kommunikointitaidot sekä ammattimaisuus.

Loppuihin neljään tietoaalueeseen kuhunkin kuluu alle 20 tuntia opetusta. Ohjelmiston hallintaan kuluu 19 tuntia, ohjelmiston laatuun kuluu 16 tuntia, ohjelmistoprosessiin kuluu 13 tuntia ja ohjelmiston evoluutioon kuluu 10 tuntia opetusta.

CCSE:n suosituksissa opiskelijan tulisi myös erikoistua vähintään yhdellä osa-alueella. Opiskelija voi erikoistua yhteen tai useampaan ydintietoaalueeseen, jolloin opiskelijan tulisi osata tietoaalueen materiaali laajemmin kuin suosituksissa on määriteltynä. Tämän lisäksi opiskelija voi erikoistua yhdellä tai use-

ammalla menetelmäalueella. Suosituksissa on mainittuna yhteensä 15 kappaletta menetelmäalueita, mutta listan ei ole tarkoitus olla kaiken kattava, vaan oppilaitokset voivat omien tai yhteisön tarpeiden mukaan sisällyttää tutkintoon omia menetelmäalueita. Seuraavaksi avaan suosituksissa mainittuja menetelmäalueita.

Verkkokeskeisillä järjestelmillä tarkoitetaan yleisesti käytössä olevia www-pohjaisia internetsivuja. Tähän kuuluu myös tietämys verkkojen toiminnasta sekä tietoturvasta.

Informaatiojärjestelmiin ja datan prosessointiin kuuluu tietämystä tietokannoista ja liiketoiminnan ylläpidosta. Näiden lisäksi opiskelijan tulisi hallita datan ylläpitoa ja varastointia.

Taloudellisiin ja verkkokaupankäyntijärjestelmiin kuuluu tietämys kirjainpidosta ja talouden ylläpidosta. Tähän menetelmäalueeseen kuuluu myös tietämys tietoturvasta.

Vikasietoisten ja kestävien järjestelmien -menetelmäalueella opiskelijalla tulisi olla tietämystä ja taitoa heterogeenisissä ja hajautetuissa järjestelmissä. Lisäksi opiskelijan tulisi hallita vikojen analysointia ja palautumista, hyökkäyksen havaitsemista sekä tietoturvaa.

Korkean suojauksen järjestelmissä opiskelijan tulee ymmärtää liiketoimintaan liittyvät suojausongelmat sekä suojausten heikkoudet ja riskit. Näiden lisäksi opiskelijan tulee tuntea erilaiset suojaukset, salaukset ja salaustenanalyysia sekä tietämystä verkoista.

Kriittisen turvallisuuden järjestelmillä tarkoitetaan järjestelmiä, joissa virheellinen toiminta voi johtaa ihmisen loukkaantumiseen tai kuolemaan, laitteen vakavaan vahingoittumiseen tai tuhoutumiseen tai vakavaan ympäristövahinkoon. Tässä menetelmäalueessa opiskelijan tulee hallita formaalit menetelmät ja oikeellisuuden todistamisen. Lisäksi opiskelijalta vaaditaan tietämystä kontrollijärjestelmistä, virhetiloista, seurausanalyysistä sekä vikapuuanalyysistä.

Sulatetuissa ja reaaliaikaisissa järjestelmissä opiskelijan tulisi hallita sulautettujen järjestelmien laitteistoa ja niiden verifikaatiota. Lisäksi opiskelijan tulee tietää ohjelmointikielet sekä työkalut niiden kehittämiseen sekä hallita laitteistoihin liittyvät ajoitusongelmat.

Biolääketieteellisissä järjestelmissä opiskelijan tulee hallita biologiaa ja siihen liittyviä tieteitä sekä tuntea alaan kuuluvien kriittisen turvallisuuden järjestelmiä.

Tieteellisissä järjestelmissä opiskelijalla on tietämystä läheisistä tieteenaloista sekä tilastotieteestä. Lisäksi opiskelijan tulisi osata visualisoida käytössä olevaa dataa.

Telekommunikaatiojärjestelmissä opiskelijalla tulee olla tietoa signaalin käsittelystä ja informaatioteoriasta sekä puhelin- ja telekommunikaatioprotokollista.

Avioniikka- ja kulkuneuvojärjestelmissä opiskelijan tulee tietää konetekniikan peruserätykset. Lisäksi opiskelijan tulee tuntea tähän menetelmäalueeseen liittyvät kriittisen turvallisuuden järjestelmät sekä sulautetut ja reaaliaikaiset järjestelmät.

Teollisten prosessien kontrollijärjestelmissä opiskelijan tulee hallita yleisesti kontrollijärjestelmät. Tämän lisäksi opiskelijan tulee tuntea tuotantotekniikkaa sekä muita läheisiä tekniikan aloja. Näiden lisäksi opiskelijan tulee osata näihin liittyvät sulautetut ja reaaliaikaiset järjestelmät.

Multimedia-, peli- ja viihdejärjestelmissä opiskelijan tulee hallita visualisoinnin, haptiikan sekä grafiikan. Näiden lisäksi tarvitaan tuntemusta käyttöliittymäsuunnittelusta ja verkoista.

Järjestelmät pienille ja mobiileille alustoille -menetelmäalueeseen sisältyy tietämys langattomista teknologioista. Lisäksi siihen kuuluu käyttöjärjestelmien suunnittelua pienille ja mobiileille alustoille. Opiskelijan tulee myös hallita tähän liittyvät sulautetut ja reaaliaikaiset järjestelmät sekä telekommunikaatiojärjestelmät.

Agenttipohjaisissa järjestelmissä opiskelijan tulee hallita koneoppimista, sumeaa logiikkaa ja tietämystekniikkaa.

2.2.3. Ylemmän korkeakoulututkinnon suositukset

Vuonna 2007 yliopistomaailma, työelämä, Yhdysvaltojen puolustusministeriö sekä ammatilliset yhteisöt perustivat yhteenliittymän, jonka päätarkoituksena oli luoda ohjaava opetussuunnitelma maisteritason ohjelmistotekniikan koulutukselle. Opetussuunnitelma valmistui vuonna 2009 ja sitä kutsuttiin nimellä Graduate Software Engineering 2009 (GSWE2009). Opetussuunnitelman tuli sisältää oletettuja lopputuloksia, sisäänpääsyvaatimukset, oppaan arkkitehtuurin sekä tietopohja, jonka avulla laitokset voivat päivittää opetussuunnitelmiaan. Suositusten tarkoituksena oli olla mahdollisimman joustavia, jotta laitokset voisivat omaksua suositukset ja mukauttaa sitä paikallisten vaatimusten mukaiseksi [5].

GSwE2009 valmistumiseen saakka oli olemassa vain SEI:n tekemä opetus-suunnitelma ohjelmistotekniikalle (katso alakohta 2.2.1.), joka valmistui vuonna 1989. GSwE2009:ta lähdettiin kehittämään seuraaville tavoitteilla:

1. Parantamaan olemassa olevia maisteritason ohjelmistotekniikan ohjelmia yliopistojen, opiskelijoiden, valmistuneiden, ohjelmistojen kehittäjien ja ohjelmistojen ostajien näkökulmasta.
2. Mahdollistamaan uusien ohjelmistotekniikan maisteritason ohjelmien muodostamisen tarjoamalla opetussuunnitelman sisältöä ja neuvoja, kuinka sen voi toteuttaa.
3. Tukemaan kasvavaa opiskelijoiden määrää ohjelmistotekniikan maisterilinjalla kasvattamalla ohjelman arvoa tuleville opiskelijoille sekä työnantajille.

GSwE2009 määrittää kolme pääsyvaatimusta, jotka jokainen maisteriopintojen aloittava tulisi täyttää. Ensimmäiseksi opiskelijalla tulee olla alempi korkeakoulututkinto tietotekniikan alalta, kuten esimerkiksi tietojenkäsittelytiede. Vaihtoehtoisesti opiskelijalla voi olla alempi korkeakoulututkinto insinööritieteestä tai muulta luonnontieteellisestä alalta sekä sivuaineena tietotekniikka. Toinen vaatimus on perustietotaso ohjelmistotekniikasta, joka vastaa yhtä ohjelmistotekniikan perusteet kurssia. Kolmantena vaatimuksena opinnot aloitavalla opiskelijalla tulee olla vähintään kahden vuoden työkokemus. Mikäli työkokemusta ei ole, niin oppilaitoksen tulisi tarjota työharjoittelua työkokemuksen kartuttamiseksi. GSwE2009:n työryhmän mielestä harjoittelu ei kuitenkaan vastaa aitoa työkokemusta alalta, joten ohjelman pituutta tulisi samalla kasvattaa.

GSwE2009:hen on määriteltynä seuraavat oppimisen tavoitteet:

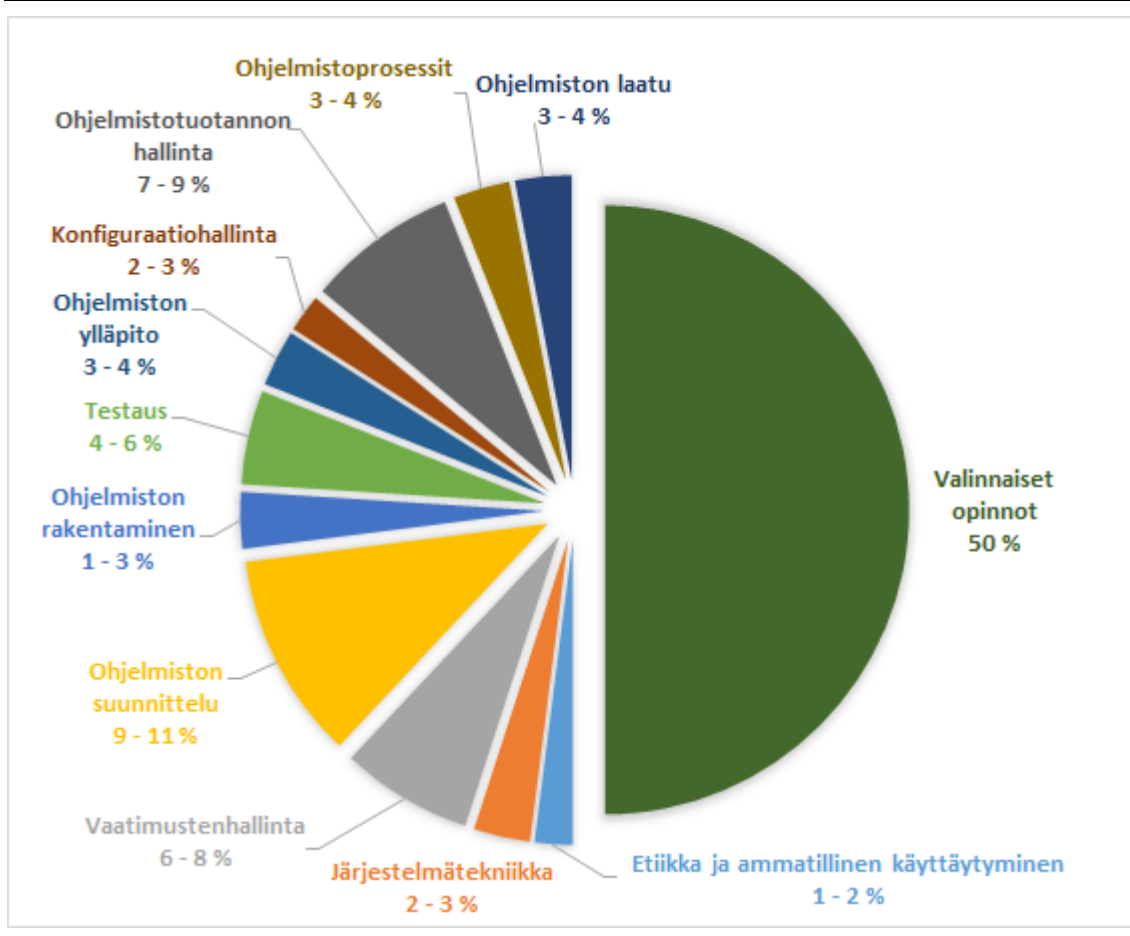
1. CBOK:n hallinta.
2. Hallita ohjelmistonkehitys yhdellä toiminta-alueella, kuten talous, terveys, kuljetusliikenne tai tietoliikenne, ja yhdellä toimintatyypillä, kuten reaaliaikainen, upotettu, turvallisuuskriittinen tai erittäin hajautettu järjestelmä.
3. Hallita vähintään yksi CBOK:n tietoalue tai yksikkö Bloomin taksonomian synteesisitasolla.
4. Tehdä eettisesti päätöksiä ammatillisesti sekä harjoittaa eettisesti hyväksyttävää ammatillista toimintaa.
5. Ymmärtää yhteys järjestelmätekniikkaan ja osata käyttää järjestelmätekniikan periaatteita ohjelmistokehityksessä.

6. Olla tehokas osa ryhmää, joka voi olla monikulttuurinen tai maantieteellisesti hajautunut. Lisäksi täytyy osata kommunikoida suullisesti ja kirjallisesti sekä johtaa ohjelmistokehitystä yhdellä kehityksen osaluueelle, kuten projektin johtaminen, vaatimusten analyysi, arkkitehtuuri, ohjelmiston kehittäminen tai laadunvarmistus.
7. Sovitella ristiriitaisia päämääriä ja löytää niille hyväksyttäviä kompromisseja hinnan, ajan, tiedon, olemassa olevan järjestelmän ja organisaation puitteissa.
8. Ymmärtää ja arvostaa käyttökelpoisuusanalyysia, neuvottelutaitoa ja hyvää kommunikaatiota asianomaisten kanssa.
9. Oppia uusia malleja, tekniikoita ja teknologioita sitä mukaan, kun niitä ilmaantuu sekä ymmärtää elinikäisen oppimisen tärkeys ohjelmistotekniikan alalla.
10. Olla kykenevä analysoimaan nykyisiä ohjelmistoteknologioita ja artikuloimaan niiden vahvuuksia ja heikkouksia, vertailemaan niitä toisiin teknologioihin sekä määrittämään ja tukemaan niiden kehitystä tai laajennusta.

GSwE2009:tä varten kehitettiin ydintietoalueet, Core Body of Knowledge (CBOK), jotka jokaisen opiskelijan täytyy hallita valmistuessaan. CBOK kehitettiin SWEBOK:n pohjalta (kohta 2.1), mutta siihen lisättiin tietoalueita, joita sen hetkessä SWEBOK:n versiossa ei ollut. Tietoalueita GSwE2009:ssä on yhteensä 11 kappaletta, jotka on jaettu yksikköihin sekä yksiköt jaettu pienempiin aiheisiin. CBOK on suunniteltu siten, että siihen määritellyn oppimisen tason hallintaan vaadittava aika on hieman alle puolet koko maisteriin vaadittavasta ajasta, joten lisäkurseja voidaan järjestää oppilaitoksen haluaman painotuksen mukaan tärkeimmiltä aihealueilta.

CBOK:n tietoalueiden opetukseen on määritelty kuluva yhteensä 200 kontaktituntia. Kontaktitunti on osa Pohjois-Amerikassa käytössä olevasta opiskelijan opintojen seurantajärjestelmästä [22]. Kontaktitunnilla tarkoitetaan opetukseen käytettävää aikaa, joten sitä voidaan verrata Suomessa käytössä oleviin opintopisteisiin. Erona näiden kahden järjestelmän välillä on se, että opintopisteellä seurataan opiskelijan opintoihin käyttämää aikaa. Kontaktituntijärjestelmässä oletetaan, että jokaista opetettua tuntia kohden (kontaktitunti) opiskelija käyttää kahden ja kolmen tunnin väliltä aikaa itsenäiseen opiskeluun tai harjoitusten tekemiseen. Tällöin yksi kontaktitunti tuottaa opiskelijalle työtä kolmen ja neljän tunnin väliltä.

CBOK:n suositusten opetussuunnitelma on kuvattuna ympyrädiagrammin avulla (Kuva 1). Käytössä olevat 200 kontaktituntia ovat jaettu mahdollisimman joustavasti tietalueiden kesken, koska siitä ei haluttu tehdä liian ehdotonta. Jaettaessa 200 kontaktituntia tietalueiden kesken, eniten aikaa kuluu ohjelmiston suunnittelu -tietalueeseen, johon kuluu 19 – 22 prosenttia kontaktitunneista. Seuraavaksi eniten aikaa on varattu ohjelmistotuotannon hallinta -tietalueeseen, johon sisältyy 16 – 17 prosenttia kontaktitunneista. Vaatimustenhallinta-tietalueeseen on tarkoitus käyttää 14 – 15 prosenttia kontaktitunneista. Neljänneksi suurin on testaus-tietalue, johon kuluu 10 – 11 prosenttia kontaktitunneista. Noin seitsemän prosenttia kontaktitunneista on varattu ohjelmiston ylläpito, ohjelmistotuotannon prosessi ja ohjelmiston laatu -tietalueeseen. Viisi prosenttia kuluu järjestelmäteknikka ja konfiguraationhallinta -tietalueeseen. Ohjelmiston rakentaminen -tietalueeseen on tarkoitus käyttää 2 – 5 prosenttia kontaktitunneista. Lopuksi vielä 2 – 4 prosenttia kuluu etiikka ja ammatillinen käyttäytyminen -tietalueeseen.



Kuva 1 – CBOK:n tietalueiden ajan jakauma

GSwE2009 on käytetty useammassa yliopistossa onnistuneesti sen kehittämisen jälkeen. Ardis ja kumppanit [6] tekivät kirjallisuuskatsauksen, jossa he kävivät läpi oppilaitokset, joissa oli käytetty apuna GSwE2009:ää joko päivittämään maisteriohjelmaa tai luomaan sellaisen. Onnistunutta GSwE2009:n sisältöä kuvaa hyvin se, että Rose-Hulman teknologiainstituutissa yritettiin vuonna 2006 perustaa maisteritason ohjelmistotekniikan linja, mutta ehdotettua rakennetta ei silloin hyväksytty. GSwE2009:n julkaisun jälkeen Rose-Hulman teknologiainstituutissa aiottiin uudestaan perustaa ohjelmistotekniikan linja ja siinä myös onnistuttiin käyttäen GSwE2009:tä. Lisäksi Uruguay'n suurimmassa yliopistossa, Universidad de la República, käytettiin GSwE2009:ta evaluoimaan olemassa olevaa maisteritason linjaa ja löydettiin useita alueita, joissa koulutusta voidaan parantaa [13, 14].

2.3. Ohjelmistotekniikan opetus Suomessa

Korkeamman korkeakoulututkinnon saanut opiskelija, pääaineenaan tietotekniikkaa tai tietojenkäsittelytiedettä, on Suomessa diplomi-insinööri tai filosofian maisteri. Tutkinnot ovat kuitenkin rakenteeltaan hyvin samankaltaiset, vaikka nimi onkin eri. Kummassakin tutkinnossa opinto-pisteitä vaaditaan opiskelijalta saman verran. Englanniksi molemmat tutkinnot käännetäänkin Master of Science. Suomessa insinööri (AMK) -tutkinto ja kandidaatintutkinto määritellään alemmaksi korkeakoulututkinnoksi, mutta insinööri (AMK) -tutkinto on laajuudeltaan 240 opintopistettä ja kandidaatintutkinto on 180 opintopistettä. Kandidaatin ja insinöörin (AMK) tutkintojen rakenne eroaa muutenkin merkittävästi toisistaan.

Suomessa opintojen edistymistä seurataan opintopisteillä. Opintopisteen määritelmän mukaan yhteen lukuvuoteen mahtuu 60 opintopistettä, mikä vastaa 1600 tuntia työtä opiskelijalta [21]. Yksi opintopiste vastaa silloin noin 27 tuntia työtä. Opiskelijalta vaadittavaan työhön on laskettu kaikki kurssiin vaadittava työ, johon sisältyy myös itsenäisesti tehty työ. Opiskelija voi saada opinto-pisteitä vain läpäistyistä kursseista. Opintopisteillä ei voi mitata opiskelijan ymmärryksen syvyyttä tai tiedon vaatimustaso. Lähtökohtaisesti neljän opintopisteen perusteet kurssin vaatimustaso ei vastaa neljän opintopisteen syventävän kurssin vaatimustaso.

2.4. Opinto-oppaat

Vertailu suoritetaan käyttämällä 2016–2017 lukuvuoden opinto-oppaita kustakin oppilaitoksesta. Vertailussa käytetään Tampereen yliopiston tietojenkäsittelytieteiden kandidaatin tutkinto-ohjelman opinto-opasta [24], Tampereen yli-

opiston ohjelmistokehityksen maisteriohjelman (Master's Degree Programme in Software Development) opinto-opasta [23], Aalto-yliopiston teknistieteellisen kandidaatinohjelman opinto-opasta [2], Aalto-yliopiston Master's Programme in Computer, Communication and Information Science opinto-opasta [1] sekä Lahden ammattikorkeakoulun tieto- ja viestintäteknikan (ohjelmistotekniikka) koulutuksen opinto-opasta [18]. Aalto-yliopiston diplomi-insinööri koulutuksen opinto-oppaassa on useita erikoistumissuuntauksia ohjelmistotekniikalle, jotka kaikki avataan tässä kohdassa.

2.4.1. Tampereen yliopisto

Tampereen yliopiston opinto-oppaat koostuvat kahdesta oppaasta: tietojenkäsittelytieteiden kandidaatinohjelma sekä ohjelmistokehityksen maisteriohjelma. Kandidaatinohjelman laajuus on 180 opintopistettä ja maisteriohjelman laajuus on 120 opintopistettä. Yhteensä opintojen suorittamiseen menee suosituksen mukaan viisi vuotta, josta kolme vuotta kuuluu kandidaatintutkintoon ja kaksi vuotta maisterintutkintoon.

Kandidaatintutkinnon opinnot koostuvat neljästä osuudesta: tietojenkäsittelytieteiden kandidaatinohjelman yhteisistä opinnoista, tietojenkäsittelytieteiden perusopinnoista, tietojenkäsittelytieteiden aineopinnot sekä tietojenkäsittelytieteiden tutkinto-ohjelman valinnaisesta opinnoista 55 opintopistettä.

Tietojenkäsittelytieteiden kandidaatinohjelman yhteiset opinnot ovat laajuudeltaan 40 opintopistettä. Yhteiset opinnot sisältävät opiskeluun orientoivia opintoja sekä viestintää ja kieliopintoja. Lisäksi näissä opinnoissa pääsee tutustumaan informaatiotutkimukseen tai interaktiivisen median opintoihin viiden opintopisteen verran. Yhteisiin opintoihin kuuluu myös matematiikan tai tilastotieteen opintoja kymmenen opintopisteen verran.

Tietojenkäsittelytieteiden perusopinnot sisältävät 25 opintopisteen verran pakollisia opintoja. Nämä opinnot sisältävät ohjelmoinnin ja tietokantojen perusteita, käyttäjäkeskeistä suunnittelua sekä tietojärjestelmien suunnittelua.

Tietojenkäsittelytieteiden aineopinnot ovat laajuudeltaan vähintään 60 opintopistettä. Aineopintoihin kuuluu pakollisia opintoja 35 – 40 opintopisteen verran. Pakolliset kurssit sisältävät käyttöliittymien suunnittelua, olio-ohjelmointia sekä ohjelmoinnin projektityö. Tämän lisäksi pakollisiin aineopintoihin kuuluu myös tieteellisen kirjoittamisen kurssi sekä kandidaatintutkielma. Aineopintoihin sisältyy myös 20 – 25 opintopisteen verran valinnaisia opintoja.

Tietojenkäsittelytieteiden tutkinto-ohjelman valinnaiset opinnot ovat laajuudeltaan 55 opintopistettä. Valinnaiset opinnot sisältävät vähintään yhden 15

- 25 opintopisteen opintokokonaisuus, johon on suositeltu joko matematiikan ja tilastotieteen opintokokonaisuutta tai informaatiotutkimuksen ja interaktiivisen median opintoja. Tämän lisäksi valinnaisiin opintoihin kuuluu täysin valinnaisia opintoja 30 - 40 opintopisteen verran, jotta saadaan kandidaatintutkintoon vaadittava 180 opintopistettä täyteen.

Tampereen yliopistossa ohjelmistokehityksen maisteriopintojen kurssit ovat pääsääntöisesti englanniksi. Tämä ohjelmistotekniikkaan suuntautuva tutkinto on rakennettu siten, että se sopii mahdollisimman hyvin ulkomaalaisille opiskelijoille. Maisteriopinnot koostuvat kolmesta pääkohdasta: yleiset maisteriohjelmien opinnot, syventävät ohjelmistotuotannon opinnot sekä vapaavalintaiset ja muut opinnot. Yhteensä opintoja tulee olla vähintään 120 opintopisteen verran.

Yleiset maisteriohjelmien opinnot on rakennettu siten, että se sopii opiskelijoille, joilla on erilainen tausta. Yleiset opinnot voidaan suorittaa kolmella eri tavalla, jotka riippuvat opiskelijan aiemmista opinnoista. Ensimmäinen tapa on suunnattu ulkomaalaisille opiskelijoille, jotka ovat suorittaneet kandidaatin tutkinnon omassa kotimaassaan tai amk-tutkinnon Suomessa; tässä orientoitumisen ja opiskelutekniikan lisäksi tutustutaan suomalaiseen kulttuuriin sekä tutkimuksen tekemiseen ja tieteelliseen kirjoittamiseen. Toinen tapa on suunnattu opiskelijoille, jotka ovat saaneet opiskeluoikeuden suoraan maisteriohjelmaan. Tämä tapa on rakenteeltaan muuten samanlainen kuin ensimmäinen tapa, mutta Suomen kulttuuriin tutustumisen sijaan opiskelija tulee suorittaa pakollinen ruotsinkielen suullinen ja kirjallinen koe. Kolmas tapa on suunnattu opiskelijoille, jotka ovat aiemmin suorittaneet kandidaatintutkinnon Tampereen yliopistossa. Tällöin opiskelijalle tulee pakollisina kursseina vain tutkimuksen suorittaminen, tieteellinen kirjoittaminen ja opiskelutaitojen kehitys. Toisaalta heiltä edellytetään kandidaattivaiheessa suoritettuja kurssitietorakenteissa ja tietokoneohjelmoinnissa.

Syventävät ohjelmistotuotannon opinnot ovat laajuudeltaan 80 opintopistettä. Puolet opintopisteistä tulee normaaleista luentokursseista ja loput tulevat pro gradu -tutkielmasta. Kursseista viisi ovat pakollisia ja käsittelevät vaatimustenmäärittelyä, formaalia spesifiointia, funktionaalista ohjelmointia sekä projektinhallintaa. Lisäksi pakollisiin kursseihin kuuluu myös graduseminaari. Syventäviin opintoihin sisältyy myös yksi valinnainen kurssi. Valittavana on kursseja, jotka käsittelevät avoimen lähdekoodin ja ohjelmistojen laatua, ohjelmistotyökaluja sekä niiden arviointia, testausta, turvallisuutta ja luottoa, suosi-

tusjärjestelmää, tietokannat ja tiedonhaun integraatiota. Lisäksi valittavana on myös ohjelmistoprojektin suorittaminen.

Valinnaisten ja muiden opintojen kokonaisuus on laajuudeltaan 18 – 39 opintopistettä. Näihin opintoihin sisältyy opiskelijalle pakollisena täydentävä projektityö tai innovaatioprojekti, mikäli opiskelija ei ole opiskellut Tampereen yliopistossa aiemmin. Tämän lisäksi opiskelijan tulee valita itsellensä valinnaisia opintoja, jotta opiskelija saa täyteen maisteritutkinnon vaadittavat 120 opintopistettä.

Tampereen yliopistoon haettaessa opiskelemaan tietojenkäsittelytiedettä opiskelija saa samalla opinto-oikeuden sekä kandidaatin että maisterin tutkintoa varten. Kandidaatintutkinto tulee kuitenkin ensin suorittaa loppuun ennen kuin opiskelijalla on oikeus jatkaa maisteriohjelmaan. Ohjelmistokehityksen maisteriohjelmaan ei ole tämän lisäksi mitään sisäänpääsyvaatimuksia, vaan kaikki halukkaan pääsevät opinnot aloittamaan.

2.4.2. Aalto-yliopisto

Aalto-yliopiston opinto-oppaat koostuvat kahdesta osuudesta: tietotekniikan kandidaatinohjelman sekä tietojenkäsittely-, kommunikaatio- ja informaatiotieteiden diplomi-insinöörin ohjelma (Master's Programme in Computer, Communication and Information Sciences). Molemmat tutkinnon ovat rakenteeltaan samakaltaisia Tampereen yliopiston vastaavien kandidaatin- ja maisterintutkintojen kanssa. Tietotekniikan kandidaatintutkinnon laajuus on yhteensä 180 opintopistettä ja diplomi-insinöörintutkinto on laajuudeltaan 120 opintopistettä. Tutkintojen suorittamiseen suositellaan käytettäväksi yhteensä viisi vuotta.

Tietotekniikan kandidaatinohjelma koostuu neljästä osasta: perusopinnot, pääaineopinnot, sivuaineopinnot sekä vapaasti valittavista opinnoista. Perusopinnot ovat laajuudeltaan 65 opintopistettä. Näihin opintoihin sisältyy tietotekniikan opintoja 15 opintopisteen verran, joihin sisältyy ohjelmointia sekä tietorakenteita ja algoritmeja. Lisäksi perusopinnot sisältävät matematiikan, fysiikan, tuotantotalouden opintoja yhteensä 40 opintopisteen laajuudelta. Näiden lisäksi perusopintoihin kuuluu vielä opiskelutekniikkaa sekä yleiset ja kielioopinnot.

Pääaineopintojen laajuus on 65 opintopistettä, jotka kaikki ovat tietotekniikan kursseja. Pääaineopinnoista 55 opintopistettä tulee pakollisista kursseista. Pakollisilla kursseilla käsitellään tietokantoja, tietoverkkoja, teoreettista tietojenkäsittelytiedettä sekä neljä erilaista ohjelmistoprojektia. Näiden lisäksi pakollisiin opintoihin sisältyy kandidaatintyö seminaareineen sekä kypsyysnäyte.

Pääaineopintoihin kuuluu näiden lisäksi 10 opintopisteen verran vapaavalintaisia kursseja. Opiskelijalla on mahdollista valita seitsemästä kurssista kaksi suoritettavaksi. Valittavat kurssit käsittelevät tietokonegrafiikkaa, käyttöliittymäsuunnittelua, web-ohjelman kehitystä, tietoturvaa, tietotekniikkaa, datan analyysimenetelmiä sekä käyttöjärjestelmiä. Oppaassa muistutetaan, että näitä valinnaisia kursseja voidaan sisällyttää tutkinnon vapaavalintaisiin opintoihin.

Sivuaineopintojen kokonaisuus on 20 – 25 opintopistettä riippuen valittavasta sivuaineesta. Kokonaisuutena kuitenkin sivuaineen ja vapaasti valittavien opintojen kokonaisuus tulee olla yhteensä 50 opintopistettä. Sivuainekokoisuus on täysin vapaavalintainen kokonaisuus. Vapaavalintaiset opinnot ovat myös täysin opiskelijan oman harkinnan varassa. Opinto-oppaassa kuitenkin edelleen korostetaan, että pääaineopintojen vapaavalintaisia kursseja voi sisällyttää vapaasti valittaviin opintoihin. Vaihtoehtoisesti oppaassa suositellaan syventämään matemaattista tai menetelmällistä osaamista.

Aalto-yliopistolla on laaja kirjo opintosuuntia opiskelijoille, jotka haluavat erikoistua ohjelmistotekniikkaan. Opintosuuntia on yhteensä neljä: ohjelmistotekniikka, käyttäjäkeskeinen suunnittelu, palvelujen suunnittelu ja tekniikka sekä yritysjärjestelmät. Näistä neljästä kolme voidaan suorittaa pitkänä pääaineena, jonka laajuus on 55 – 65 opintopistettä. Kaikki voidaan suorittaa kompaktina pääaineena, jolloin opiskelijan tulee suorittaa myös lyhyt sivuainekokoisuus. Kompaktin pääaineen laajuus on 40 – 45 opintopistettä. Diplomi-insinöörin tutkintoon kuuluu tämän lisäksi diplomityö (30 opintopistettä) ja vapaa valintaiset opinnot 25 – 35 opintopistettä, joten diplomi-insinöörin tutkinto on yhteensä 120 opintopisteen laajuinen.

Kaikilla opintosuunnilla on yhteisesti pakollisena kaksi kurssia, ohjelmistotekniikka sekä käyttäjäkeskeiset menetelmät tuotteen ja palvelun suunnittelulle. Lisäksi jokaisella opintosuunnalla on omat kurssinsa pakollisena sekä vapaasti valittavia kursseja. Jotkin pääaineopintojen kursseista ovat osa kandidaatintutkinnon pääaineen valinnaisia opintoja. Tällöin opiskelijan tulee valita niiden tilalle muita syventäviä kursseja.

Pitkässä ohjelmistotekniikan opintosuuntauksessa opiskelijan on suoritettava kolme pakollista kurssia: ohjelman suunnittelu ja mallinnus, ohjelmistoprojekti sekä ohjelmisto- ja palvelutekniikan seminaari. Kompaktissa opinnoissa ohjelmisto- ja palvelutekniikan seminaari siirtyy valinnaisiin kursseihin. Vapaavalintaisia kursseja opiskelijan tulee suorittaa 20 – 40 opintopisteen verran. Valinnaisilla kursseilla käsitellään tutkimus menetelmiä ohjelmisto- ja palvelutekniikoille, ohjelmistoprosesseja ja -projekteja, vaatimustenmäärittelyä, ohjel-

mistoarkkitehtuuria sekä ohjelmiston testausta ja laadun varmistusta. Edellä mainittujen kurssien lisäksi opiskelijan tulee suorittaa myös erikseen luetelluista kursseista niin paljon, että saadaan tutkintoon vaadittava opintopistemäärä suoritettua. Valittavana on kursseja, jotka sisältävät innovointia ja projektin hallintaa, johdantoa IT yrityksiin ja liiketoimintaan, teknologia liiketoimien hallintaa, tutkimuksen ja kehityksen hallinnan perusteita ja johtamisen perusteita.

Käyttäjakeskeisen suunnittelun opintosuuntauksen voi valita pitkänä tai kompaktina pääaineopintoina. Pitkässä suuntauksessa opiskelijan tulee suorittaa pakollisina kursseina suunnitteluprojekti, käytettävyyden arviointi ja käyttöliittymän rakentaminen. Kompaktissa suuntauksessa suunnitteluprojekti siirretään valinnaisiin opintoihin. Valinnaisia kursseja opiskelijan tulee suorittaa 25 opintopisteen verran. Valittavana on samoja kursseja kuin ohjelmistotekniikan puolella, kuten vaatimustenmäärittely ja ohjelmistoprojekti. Näiden lisäksi valinnaiset kurssit käsittelevät nousevia käyttöliittymiä, ihmistä havaitsijana ja laskennallista käyttöliittymäsuunnittelua. Lisäksi valittavana on tutkimusprojekti ihmisen ja tietokoneen vuorovaikutuksesta sekä ihminen ryhmässä.

Palvelujen suunnittelun ja tekniikan opintosuuntauksessa on tarjolla vain kompakti pääaine. Tästä suuntauksesta on lisäksi tarjolla lyhyt sivuainevaihtoehto, joka on laajuudeltaan 20 opintopistettä. Kompaktin pääaineen opinnot ovat laajuudeltaan 40 – 45 opintopistettä. Kaikki pääaineen opintoihin kuuluvat pakolliset ja valinnaiset kurssit ovat samoja ohjelmistotekniikan ja käyttäjakeskeisen suunnittelun opintosuuntauksen kanssa. Eroavaisuudet muodostuvat, kun opintosuuntaus suoritetaan lyhyenä sivuaineena. Sivuaineen pakollisena kurssina on johdanto palvelujen tuottamiseen. Pakollisen kurssin lisäksi opiskelijan tulee valita erikseen luetelluista kursseista vähintään kaksi. Valittavana on kursseja, jotka käsittelevät yritysten palvelutoimintaa, palvelujen suunnittelua, ICT:n mahdollistamia palveluyrityksiä ja innovointia, informaatiotaloutta sekä palveluyrityksen malleja. Edellisten kurssien lisäksi opiskelijan tulee suorittaa valinnaisia kursseja, jotta saadaan täyteen vaadittavat 20 opintopistettä lyhyeen sivuaineeseen. Valinnaiset kurssit sisältävät projektinhallintaa ja konsultoinnin käytäntöjä, kriittisiä ongelmia informaatiojärjestelmien tutkimuksessa, innovointia palveluissa ja toiminnassa, henkilöstöä palvelutoiminnassa sekä palveluiden markkinointia.

Yritysjärjestelmä opintosuunnasta opiskelija voi valita pitkän tai kompaktin. Pitkällä ja kompaktilla opinnoilla ei ole muuta eroa muuta kuin pitkässä tulee suorittaa enemmän kursseja, jotta opintopistevaatimukset täyttyvät. Osa valittavista kursseista on samoja kuin ohjelmistotekniikan, käyttäjakeskeisen

suunnittelun sekä palvelujen suunnittelun ja tekniikan opintosuuntauksilla. Pakollisissa kursseissa on aiheena yritysjärjestelmien arkkitehtuuri, ICT:n mahdollistamat palveluyritykset ja yritysinformaatiojärjestelmien seminaari. Valinnaisina kursseina on kursseja, jotka käsittelevät yrityksen ja projektien hallintaa sekä niihin liittyvää lakia. Pitkässä opinnoissa on vielä valittavana kursseja, jotka käsittelevät yrityksen taloutta.

2.4.3. Lahden ammattikorkeakoulu

Lahden ammattikorkeakoulun insinöörin (AMK) tutkinto on laajuudeltaan 240 opintopistettä ja koulutus kestää neljä vuotta. Opiskelija hakee tieto- ja viestintätekniikan koulutukseen, josta on mahdollista erikoistua ohjelmistotekniikkaan. Tutkinto-ohjelma koostuu Lahden ammattikorkeakoulun yhteisistä opinnoista, ammatillisen ydinosaamisen opinnoista, harjoittelusta, oppinäytetyöstä sekä täydentävän osaamisen opinnoista.

Lahden ammattikorkeakoulun yhteisten opintojen pakollisiin kursseihin kuuluvat ruotsin ja englannin kielen opintoja ja viestintä. Näiden lisäksi yhteisissä opinnoissa käsitellään ammatillisen osaamisen kehittämistä, tutkimusta ja kehittämistä sekä tulevaisuuden ennakkointia. Yhteiset opinnot ovat laajuudeltaan 20 opintopistettä

Ammatillisen ydinosaamisen opinnot ovat laajuudeltaan 110 opintopistettä. Nämä opinnot ovat jaettu kahdeksaan osa-alueeseen.

Digitaalisuuden osa-alueessa käsitellään tulevaisuuden digitaalisuutta, verkkoja, tietoturva ja pilvipalveluja sekä digitaalisia työkaluja. Tämä osa-alue on laajuudeltaan kymmenen opintopistettä.

Mekaniikan osa-alueeseen kuuluu kaksi kurssia, mitkä sisältävät maattisia työvälineitä sekä mekaniikan sovelluksia. Osa-alueen laajuus on kymmenen opinto-pistettä.

Sähkön, lämmön ja energian osa-alueessa tutustutaan alueen nimensä mukaisesti sähköön, lämpöön ja energiaan. Näiden lisäksi tähän alueeseen sisältyy myös englantia insinööreille. Yhteensä tämä osa-alue on laajuudeltaan 15 opintopistettä.

Esineiden internetin osa-alueessa käsitellään WWW-teknologioita, tietoverkkojen perusteita sekä tehdään esineiden internetin projekti. Tämä osa-alue on laajuudeltaan yhteensä 15 opintopistettä.

ICT-järjestelmien osa-alueeseen kuuluu neljä kurssia, joista opiskelijan tulee valita kolme. Kursseihin sisältyy ohjelmoinnin perusteita, järjestelmiä ja lait-

teistoja, sulautettuja järjestelmiä sekä media järjestelmiä. Kaikki kurssit ovat viiden opintopisteen laajuisia, joten osa-alue on yhteensä 15 opintopistettä.

Sovellusten osa-alueella käsitellään olio- ja peliohjelmoinnin perusteita. Lisäksi tutustutaan esineiden internetin tietovarastoihin sekä tehdään sovellus projektissa. Tämä osa-alueen laajuus on 15 opintopistettä.

Alustojen ja ohjelmointikielien osa-alueessa tutustutaan erilaisiin ohjelmointikieliin, hajautettuihin järjestelmiin sekä tarkastellaan ohjelmointi erilaisille alustoille. Osa-alueen laajuus on yhteensä 15 opintopistettä.

Ohjelmistotekniikan osa-alueeseen kuuluu eri alueita ohjelmistotekniikasta, kuten ohjelmistojen arkkitehtuuri, ohjelmistojen testaus ja ylläpito sekä käyttäjäkokemus. Tämä osa-alue on laajuudeltaan 15 opintopistettä.

Täydentävän osaamisen opintojen laajuus on yhteensä 60 opintopistettä. Nämä opinnot ovat opiskelijalle täysin valinnaiset. Kaikkien yllämainittujen lisäksi opiskelijan tulee suorittaa harjoittelu tehden alan töitä yrityksessä. Harjoittelun yhteydessä opiskelija suorittaa yrityksen ympäristössä opinnäytetyön. Harjoittelun laajuus on 30 opintopistettä ja opinnäytetyö on laajuudeltaan 15 opintopistettä.

3. Tutkimuksen toteutus

Tutkimus suoritetaan tapaustutkimuksena, jonka tarkoituksena on tutkia yhtä tai pientä joukkoa tapauksia suuremmasta joukosta [27]. Tapaustutkimus sopii ennen kaikkea silloin, kun tutkija pyrkii löytämään vastauksia kysymyksiin, jotka alkavat ”miten”, ”mitä” tai ”miksi”. Tapaustutkimuksessa ei ole tarkoitus muodostaa yleistettävää tietoa, vaikka on kuitenkin mahdollista muodostaa yksittäisen tapauksen ylittävää tietoa, vaan tarkoitus on ymmärtää tutkittavaa kohdetta paremmin [27].

Tässä tapaustutkimuksessa on tarkoitus kuvailla tapauksia, mikä tapahtuu tarkastelemalla niiden rakennetta. Muodostuvaa rakennetta tullaan vertailemaan olemassa olevaan teoriaan ja arvioimaan teorian perusteella, kuinka hyvin tapaukset vastaavat teoriaa. Tässä tutkimuksessa tapauksia edustavat oppilaitokset ja niiden opinto-oppaat. Teoriaa vastaavat viimeisimmät kansainväliset suositukset niin kandidaatinohjelmalle kuin maisteriohjelmalle.

3.1. Tutkimusmenetelmälliset valinnat ja aineisto

Camilloni ja kumppanit [14] käyttivät GSWE2009:tä Uruguay'n suurimman yliopiston, Universidad de la República, ohjelmistotekniikan maisteriohjelman arvioimiseen. Tässä tutkimuksessa tutkijat selvittivät yliopiston kursseihin käytettävän ajan. Tämän jälkeen kukin kurssi luokiteltiin yhteen tietoaalueeseen kurssin sisällön perusteella. Näin saatiin selville, kuinka monta tuntia kutakin tietoaaluetta oppilaitoksessa opetettiin. Camillonin ja kumppanien luokittelun perusteella heidän tutkimassaan yliopistossa opetetaan ohjelmiston suunnittelu-tietoaaluetta yhteensä 156,8 tuntia, josta he näkivät, että opetus on riittävällä tasolla. Suosituksissa määritellään, että tietoaalueiden opetukseen tulee käyttää 600 kontaktituntia. Kontaktitunnin määritelmän mukaan opiskelijan tulee opiskella itsenäisesti kahden ja kolmen tunnin väliltä jokaista kontaktituntia kohden. Tällöin esimerkiksi opiskelijan tulisi opiskella suositusten mukaan 66 ja 88 tunnin väliltä ohjelmiston suunnittelu-tietoaaluetta. Tutkimuksessa voitiin tällöin todeta, että oppilaitoksen ohjelmiston suunnittelun-tietoaaluetta opetetaan riittävästi.

GSWE2009:ssä arvioidaan, että maisterin tutkintoon kuluu opiskelijalta 1287 - 2016 tuntia aikaa ja Camillonin ja kumppanien tutkiman yliopiston maisteriohjelmassa kului 1650 tuntia aikaa, jolloin tutkielmassa tehtyä vertailua on järkevää suorittaa. Suomessa maisterin tutkinnon laajuus on määritelty tarkasti ja sen tulee olla 120 opintopistettä. Opintopisteen määritelmän mukaan yksi opin-

topiste tarkoittaa 27 tuntia työtä opiskelijalta, jolloin Suomessa opiskelijan tulee opiskella 3240 tuntia maisterin tutkintoa varten. Tällöin Camillonin ja kumppanien käyttämää arviointitekniikkaa ei voida käyttää Suomen linjojen tutkimiseen.

CCSE:ssä ei ilmoiteta, kuinka suuri osuus tutkinnosta kuluu tietoalueiden opetukseen. Mikäli käytetään samaa arviota kuin GSWE2009:n suosituksissa, niin tietoalueiden opetukseen kuluu noin puolet tutkinnon ajasta. Tietoalueiden opetukseen on käytettävä suositusten mukaan vähintään 494 tuntia opetusta. Käytettäessä kontaktitunnin määritelmää, opiskelijan kuormaksi tulee 1976 tuntia. Tällöin kandidaatin tutkinnon laajuus olisi 3954 tuntia. Suomessa kandidaatin tutkintoon kuluu opiskelijalla aikaa 4860 tuntia, joka olisi noin 22 prosenttia laajempi tutkinto kuin suosituksissa. Camillonin ja kumppanien käyttämää arviointitekniikkaa ei myöskään voida soveltaa kandidaatintutkinnon arviointiin.

Tässä tutkielmassa täytyy vertailua suorittaa jotenkin muuten kuin Camillonin ja kumppanien käyttämä tunteihin pohjautuvaa vertailua, koska suositukset eivät oleta tutkintojen olevan yhtä pitkiä kuin Suomessa. On otettava myös huomioon, että Suomessa oppilaitokset tarjoavat usein paljon valinnaisuutta opiskelijalle, jota Camillonin ja kumppanien tutkimuksessa ei mainittu. Tämä käytännössä tarkoittaa sitä, että opiskelija pystyy sopivilla valinnoilla luomaan itsellensä opintopolun, joka vastaa joko hyvin suosituksia tai erittäin huonosti. Tällöin ei ole mielekästä vertailla pelkkään tutkintoon johtavia kursseja suosituksiin vaan olisi parempi peilata tarjontaa suosituksiin.

Tutkimuksessa otetaan tutkintojen opinto-oppaista mainitut kurssit ja ne luokitellaan suosituksista löytyviin tietoalueisiin. Luokiteltujen kurssien opintopisteitä käyttäen kullekin tietoalueelle pystytään määrittämään osuus. Tuota osuutta voidaan verrata suosituksista löytyviin osuuksiin. Osuudet voidaan myös järjestää suuruusjärjestykseen, jota voidaan myös vertailla suositusten painotuksiin.

3.2. Luokittelu

Tutkimuksen aineisto kerätään siten, että opinto-oppaissa olevat kurssit luokitellaan suosituksista löytyviin tietoalueisiin tai menetelmäalueisiin. Lahden ammattikorkeakoulun insinööri (AMK), Tampereen yliopiston luonnontieteiden kandidaatti ja Aalto yliopiston tekniikan kandidaatti luokitellaan CCSE:n tietoalueiden ja menetelmäalueiden mukaan (alakohta 2.2.2) ja Tampereen yli-

opiston filosofian maisterin ja Aalto-yliopiston diplomi-insinöörin tutkinnot luokitellaan GSwE2009:n suositusten perusteella (alakohta 2.2.3).

Mikäli kurssia ei voida oppimistavoitteiden tai kurssin kuvauksen perusteella luokitella mihinkään tietoonalueeseen, niin kurssia ei oteta aineistoon mukaan. Tällaisia kursseja edustavat muun muassa kieliopinnot, opiskelutekniikan kurssit sekä tutkielmakurssit.

On myös huomioitava, että kaikki oppilaitokset tarjoavat paljon enemmän kursseja, esimerkiksi matematiikasta kuin mitä opinto-oppaissa on mainittuna. Matematiikkaa kuitenkin korostetaan CCSE:n tietoonalueissa. Tässä tutkielmassa ei ole tarkoitus kartoittaa yliopistojen koko tarjontaa ja peilata niitä kansainvälisiin suosituksiin, vaan arvioida opinto-oppaiden ohjaavaa luonnetta opiskelijan opintojen suunnittelussa. Materiaaliksi on otettu kaikki kurssit, jotka opinto-oppaissa mainitaan, eikä luokittelussa ole otettu huomioon kurssin pakollisuutta tai valinnaisuutta.

4. Analyysi

Tutkimuksen analyysia suoritetaan kahdessa osassa. Ensimmäiseksi katsotaan, miten eri opinto-oppaiden sisältö vastaa opetussuosituksia. Viisi opinto-opasta käydään kukin yksitellen läpi ja suoritetaan vertailu suosituksiin. Ensimmäisessä osassa kootaan myös kaikkien, sekä kandidaatin että maisterin, kurssien luokittelu yhteen. Koonnista pyritään arvioimaan ohjelmistotekniikan opetusta Suomessa. Toisessa osassa vertaillaan oppilaitosten kesken, josta pyritään saamaan selville, mikä oppilaitos tarjoaa opetusta, joka vastaa parhaiten opetussuosituksia.

4.1. Tampereen yliopiston tietojenkäsittelytieteen kandidaatin opinto-opas

Tampereen yliopiston tietojenkäsittelytieteen kandidaatin opinto-oppaassa [24] on mainittuna yhteensä 85 kurssia, joista 26 luokiteltiin CCSE:n tietoalueisiin ja viisi menetelmäalueisiin. Luokitelluista kursseista on mahdollista saada yhteensä 170 – 190 opintopistettä riippuen siitä, miten laajasti osan kursseista suorittaa. Opintopisteistä 140 – 160 tulee tietoalueista ja loput 30 opintopistettä tulee menetelmäalueista.

Tietoalueisiin luokitelluista kursseista kahdeksan on ohjelmistojen ydinkohdat -tietoaaluetta. Näistä kursseista opiskelija voi saada 50 – 55 opintopistettä riippuen siitä, kuinka laajasti yhden kursseista suorittaa. Kun näiden kurssien opintopisteitä verrataan kaikkien luokiteltujen kurssien opintopisteisiin, niin 29 prosenttia kursseista kuuluu tähän tietoalueeseen. Jätettäessä vertailusta pois menetelmäalueiden opintopisteet, niin tietoalueen osuudeksi tulee 34 – 36 prosenttia.

Toiseksi eniten kursseja on matemaattinen ja tekninen perusta -tietoalueessa, johon luokiteltiin yhteensä seitsemän kurssia. Kursseista voi saada 35 opintopistettä, jolloin verrattaessa opintopisteitä kokonaistarjontaan, niin tietoalue kattaa 18 – 21 prosenttia kokonaistarjonnasta. Verrattaessa pelkkiin tietoalueisiin, niin osuudeksi tulee 22 – 25 prosenttia.

Seuraavaksi eniten kursseja on ohjelmiston suunnittelu -tietoalueessa, jossa on neljä kurssia. Kursseista on mahdollista saada yhteensä 20 – 25 opintopistettä riippuen siitä, miten laajasti yhden kursseista suorittaa. Opintopisteitä verrattaessa luokiteltujen kurssien opintopisteisiin, niin tietoalueen osuudeksi tulee 12 – 13 prosenttia. Pelkkiin tietoalueiden opintopisteisiin verrattaessa osuudeksi tulee 14 – 16 prosenttia.

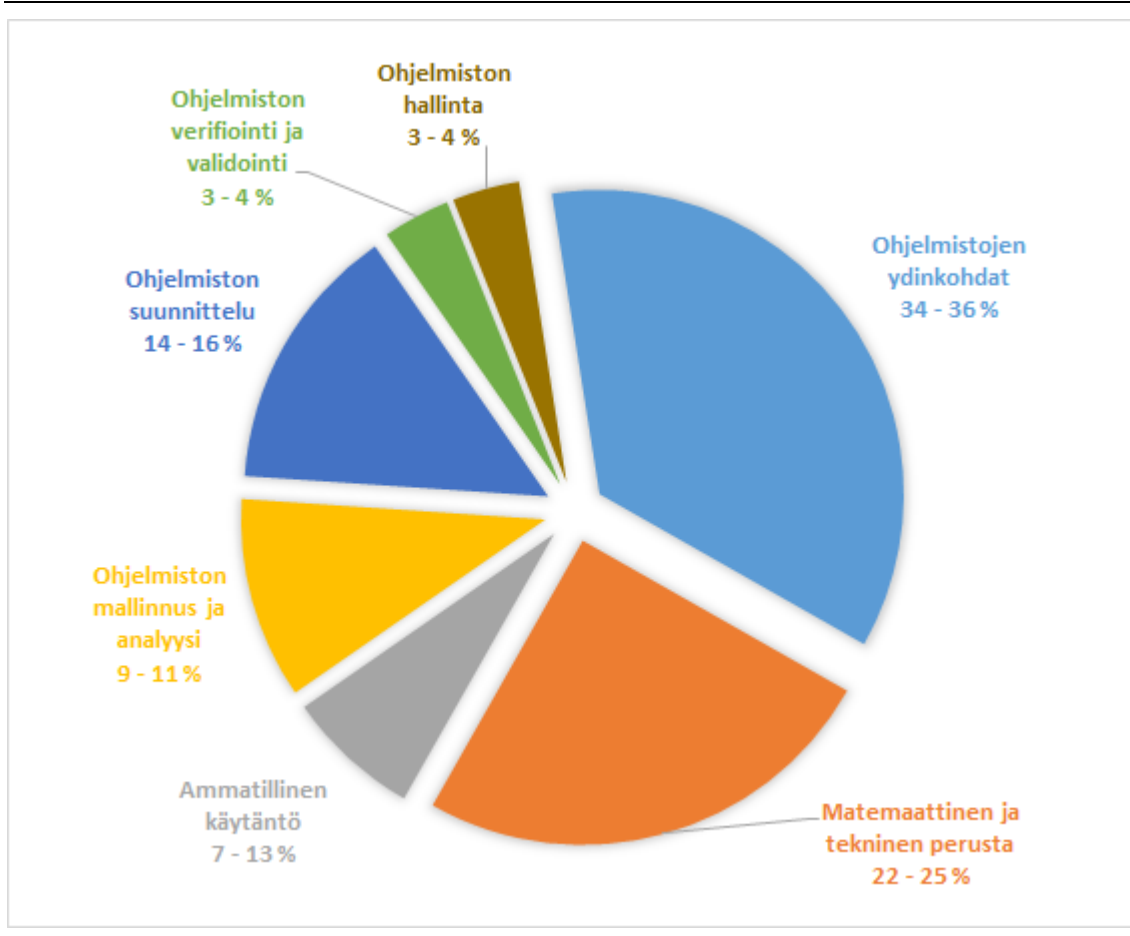
Kolme tietoalueiden kursseista on luokiteltuna ohjelmiston mallinnus ja analyysi -tietoalueeseen. Näistä kursseista jaetaan yhteensä 15 opintopistettä, jolloin verrattaessa kokonaistarjontaan tulee tietoalueen osuudeksi 8 - 9 prosenttia. Kun jätetään vertailusta menetelmäalueet pois, tulee osuudeksi 9 - 11 prosenttia.

Viidenneksi suurin tietoalue on ammatillinen käytäntö, johon on luokiteltuna kaksi kurssia. Näistä kursseista voi saada 10 - 20 opintopistettä riippuen siitä, miten laajasti molemmat kursseista suorittaa. Verrattaessa näiden kursseiden opintopisteitä kokonaistarjontaan tulee tietoalueen osuudeksi 6 - 11 prosenttia. Jätettäessä menetelmäalueiden opintopisteet pois tulee osuudeksi 7 - 13 prosenttia.

Kahteen tietoalueeseen, ohjelmiston verifiointi ja validointi sekä ohjelmiston hallinta, kuuluu kuhunkin vain yksi kurssi. Molempien kurssien laajuus on viisi opintopistettä. Kun viittä opintopistettä verrataan kaikkien luokiteltujen kurssien opintopisteisiin, niin osuudeksi tulee 3 - 4 prosenttia. Pelkkiin tietoalueisiin verrattaessa osuus jää 3 prosenttiin. Luokiteltujen tietoalueiden osuudet on havainnollistettuna ympyrädiagrammin avulla (Kuva 2).

Kolmelle tietoalueelle ei pystytty luokittelemaan yhtään kurssia. Nämä tietoalueet ovat ohjelmiston kehityskaari, ohjelmiston prosessit ja ohjelmiston laatu.

Kolme menetelmäalueelle luokitelluista kursseista kuuluu verkkokeskeiset järjestelmät -menetelmäalueeseen. Näistä kursseista voi saada yhteensä 20 opintopistettä. Kaksi muuta menetelmäalueen luokiteltua kurssia kuuluvat eri alueisiin. Ensimmäinen kuuluu informaatiojärjestelmät ja datan prosessointi -menetelmäalueeseen ja toinen kuuluu multimedia-, peli- ja viihdejärjestelmät -menetelmäalueeseen. Molemmat näistä kursseista ovat laajuudeltaan viisi opintopistettä.



Kuva 2 Tampereen yliopiston tietojenkäsittelytieteen opinto-oppaan luokittelu

4.2. Tampereen yliopiston filosofian maisterin opinto-opas

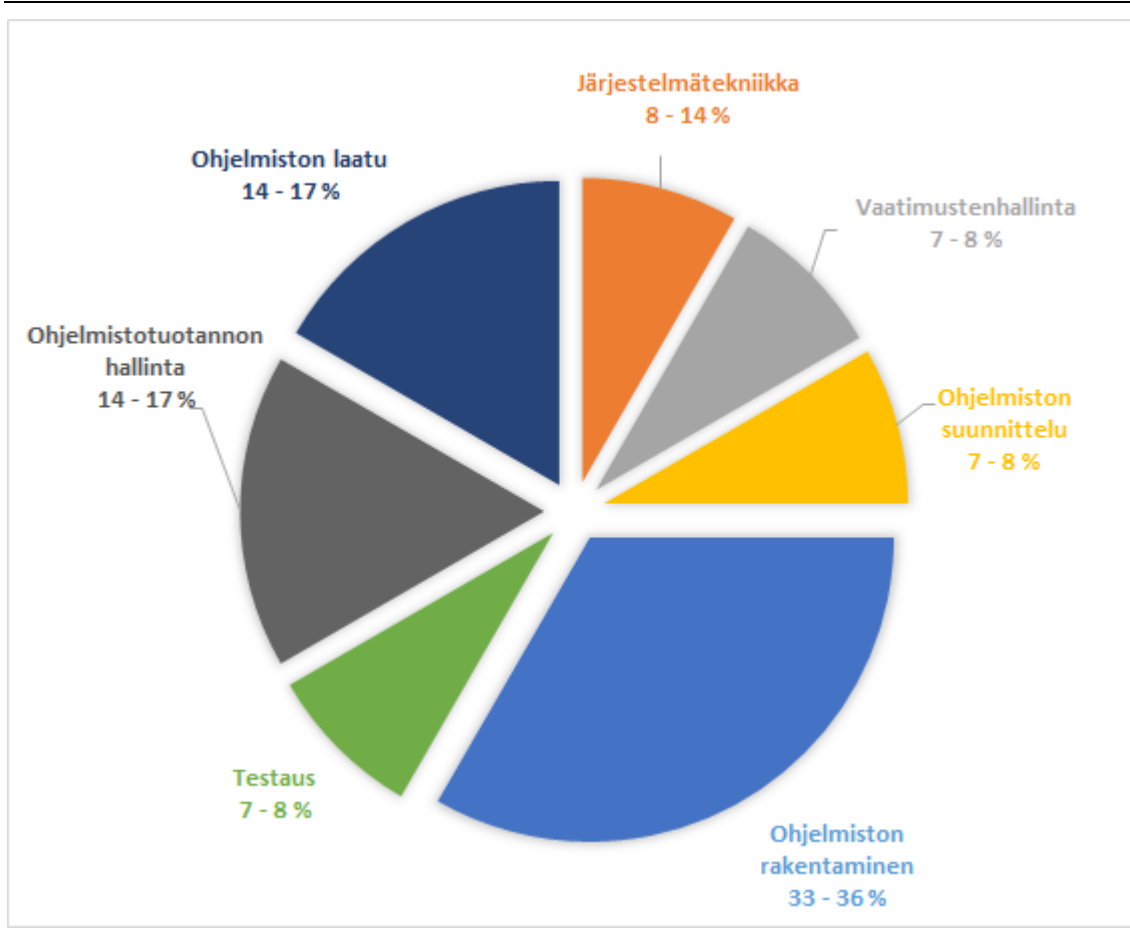
Tampereen yliopiston filosofian maisterin opinto-oppaassa on mainittuna yhteensä 23 kurssia, joista kymmenen kurssia luokiteltiin GSwE2009:n tietoi-alueisiin (alakohta 2.2.3). Luokittelemattomiin kursseihin kuuluu kursseja, jotka eivät liity ohjelmistotekniikkaan millään tavalla. Nämä kurssit käsittelevät muun muassa opiskeluun orientaatiota, opiskelutekniikkaa sekä suomalaista yhteiskuntaa ja kulttuuria. Luokittelemattomia kursseja oli myös pääaineopintojen kokonaisuudessa, josta opiskelijan tulee valita haluamansa kurssi tai kurssit. Nämä kurssit käsittelevät tietokantoja ja tiedonhakua integraatioiden näkökulmasta sekä suositusjärjestelmiä. Näitä kursseja ei kyetty luokittelemaan, koska niiden oppimistavoitteiden tai sisällön perusteella ei löytynyt mitään, mikä sopisi GSwE2009:n tietoi-alueisiin. Lisäksi Tampereen yliopistolla opiskelijalla on mahdollisuus saada opintopisteitä harjoittelusta, mutta näitä opintopisteitä ei voi luokitella mihinkään tietoi-alueeseen.

Kymmenestä luokitellusta kurssista kolme on luokiteltu ohjelmiston rakentaminen -tietoalueeseen. Näistä kursseista voi saada yhteensä 20 – 25 opintopistettä riippuen kuinka laajasti yhden kurssista suorittaa. Verrattaessa opintopisteitä luokiteltujen kurssien opintopiste määrää, niin osuudeksi tulee 33 – 36 prosenttia.

Toiseksi suurin tietoalue on ohjelmiston laatu, johon luokiteltiin kaksi kurssia. Näistä kursseista voi saada yhteensä kymmenen opintopistettä. Opintopisteillä on 14 – 17 prosentin osuus verrattuna kokonaistarjontaan.

Yhteensä viiteen tietoalueeseen kuuluu vain yksi kurssi. Nämä tietoalueet ovat järjestelmäteknikka, vaatimusten hallinta, ohjelmiston suunnittelu, testaus ja ohjelmistotuotannon hallinta. Kurssien laajuudet ovat kuitenkin erikokoisia, joten tietoalueiden osuudet vaihtelevat. Vaatimustenhallinta, ohjelmiston suunnittelu ja testaus -tietoalueiden kurssien laajuus on viisi opintopistettä. Tällöin näiden tietoalueiden osuudeksi tulee 7 – 8 prosenttia. Järjestelmäteknikka-tietoalueen kurssin laajuus on 5 – 10 opintopistettä riippuen, miten laajasti sen suorittaa. Suhteutettuna opintopisteet kokonaistarjontaan osuudeksi tulee 8 – 14 prosenttia. Ohjelmistotuotannon hallinnan -tietoalueen kurssin laajuus on kymmenen opintopistettä, jolloin tietoalueen osuudeksi kokonaistarjonnasta tulee 14 – 17 prosenttia. Luokiteltujen tietoalueiden osuudet on havainnollistettuna ympyrädiagrammin avulla (Kuva 3).

Neljälle tietoalueista ei luokiteltu yhtään kurssia. Nämä tietoalueet ovat etiikka ja ammatillinen käyttäytyminen, ohjelmiston ylläpito, konfiguraatiohallinta sekä ohjelmistoprosessit.



Kuva 3 Tampereen yliopiston filosofian maisterin opinto-oppaan luokittelu

4.3. Aalto-yliopiston tekniikan kandidaatin opinto-opas

Aalto-yliopiston tekniikan kandidaatin opinto-oppaasta löytyy yhteensä 56 kurssia, joista 20 ei pystytty luokittelemaan mihinkään tietoonalueeseen. Yhteensä 36 luokitellusta kurssista 26 kuuluu CCSE:n tietoonalueeseen ja kymmenen menetelmälueisiin. Kaikista luokitelluista kursseista voi saada yhteensä 178 opintopistettä. Pelkkien tietoonalueiden kursseista jaetaan yhteensä 130 opintopistettä.

Suurin osa tietoonalueisiin luokitelluista kursseista kuuluu ohjelmistojen ydinkohdat -tietoonalueeseen, johon kuuluu 14 kurssia. Näistä kursseista voi saada yhteensä 70 opintopistettä. Verrattaessa tietoonalueeseen luokiteltujen kurssien opintopisteitä kaikkien luokiteltujen kurssien opintopisteisiin, niin tietoonalueen osuudeksi tulee 39 prosenttia. Jätettäessä menetelmälueiden opintopisteet pois vertailusta, niin osuudeksi tulee 54 prosenttia kokonaistarjonnasta.

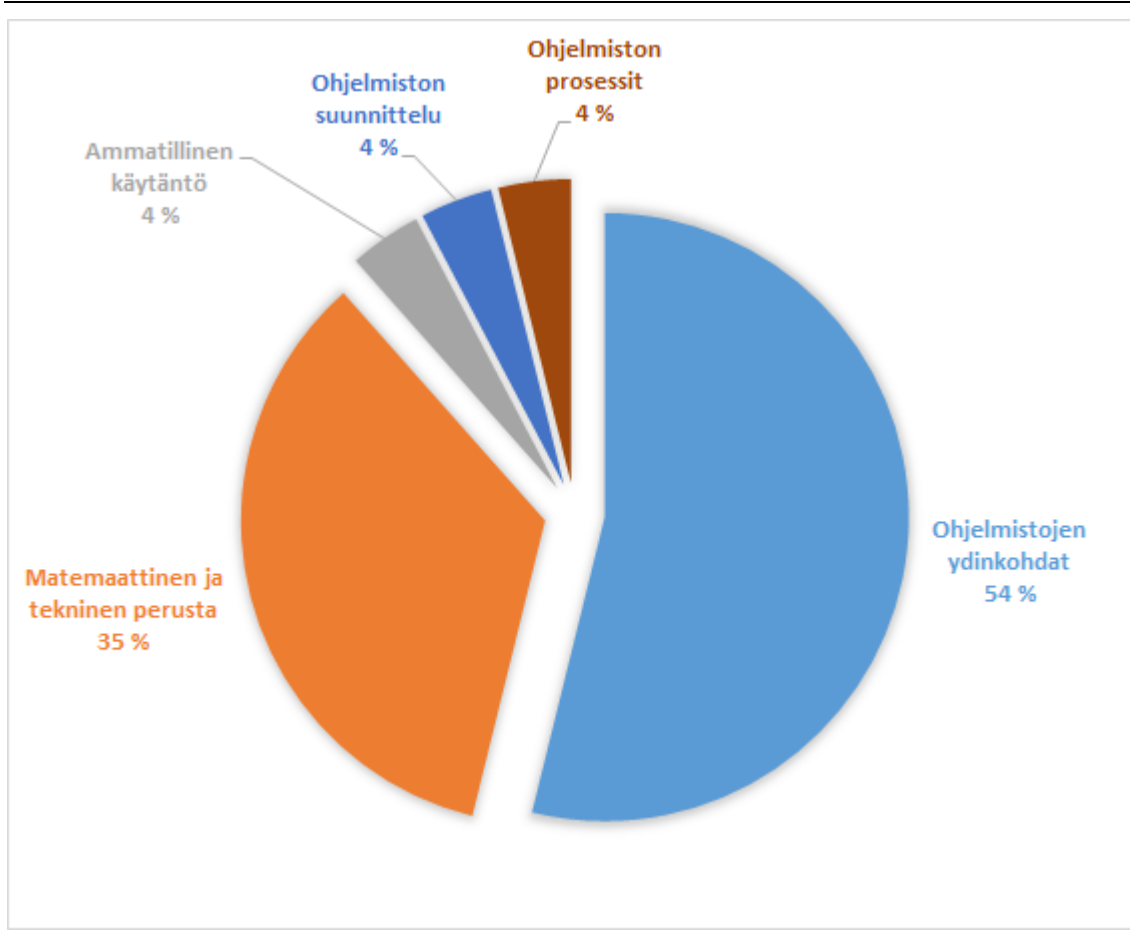
Toiseksi eniten luokiteltuja kursseja on matemaattinen ja tekninen perusta -tietoonalueessa, jossa on yhdeksän kurssia. Kursseista voi saada yhteensä 45 opintopistettä, jolloin opintopisteillä on 25 prosentin osuus verrattaessa opin-

topisteitä kokonaistarjonnan opintopisteisiin. Verrattaessa pelkkien tietoaalueiden opintopistemäärään tulee osuudeksi 35 prosenttia.

Kolmelle tietoaalueelle on luokiteltuna yksi kurssi. Nämä tietoaalueet ovat ammatillinen käytäntö, ohjelmiston suunnittelu ja ohjelmiston prosessit. Kunkin tietoaalueen kurssista saa viisi opintopistettä, jolloin kokonaistarjontaan verrattaessa tietoaalueiden osuus on kolme prosenttia. Tietoaalueiden opintopisteisiin verrattaessa osuudeksi tulee neljä prosenttia. Luokiteltujen tietoaalueiden osuudet on havainnollistettuna ympyrädiagrammin avulla (Kuva 4).

Loppuihin viiteen tietoaalueeseen ei luokiteltu yhtäkään kurssia. Nämä alueet ovat ohjelmiston mallinnus ja analyysi, ohjelmiston verifiointi ja validointi, ohjelmiston kehityskaari, ohjelmiston laatu sekä ohjelmiston hallinta.

Kymmenestä menetelmäalueeseen luokitellusta kurssista kolme kurssia on luokiteltu tieteelliset järjestelmät -menetelmäalueeseen. Näistä kursseista voi saada yhteensä 13 opintopistettä. Seuraavaksi eniten kursseja on luokiteltuna telekommunikaatiojärjestelmät-menetelmäalueeseen, joista voi saada yhteensä kymmenen opintopistettä. Loput viisi menetelmäalueiden kurssia on luokiteltuna eri menetelmäalueisiin. Nämä menetelmäalueet ovat: taloudelliset ja verkko-kaupankäyntijärjestelmät, korkean suojauksen järjestelmät, teollisten prosessien kontrollijärjestelmät, multimedia-, peli- ja viihdejärjestelmät sekä agenttipohjaiset järjestelmät. Näistä viidestä kurssista kustakin on mahdollista saada viisi opintopistettä.



Kuva 4 Aalto-yliopiston tekniikan kandidaatin opinto-oppaan luokittelu

4.4. Aalto-yliopiston diplomi-insinöörin opinto-opas

Aalto-yliopiston diplomi-insinöörin opinto-oppaasta löytyy yhteensä 52 kurssia, joista 40 luokiteltiin GSwE2009:n tietoi-alueisiin. Luokitelluista kursseista jaetaan yhteensä 192 – 219 opintopistettä riippuen siitä, kuinka laajasti osan kursseista suorittaa. Luokittelemattomista kursseista suurin osa on opiskelutekniikka, opintojen seurantaan liittyviä kursseja tai tutkimusmenetelmien kursseja.

Luokitelluista kursseista eniten kursseja luokiteltiin ohjelmistotuotannon hallinta -tietoi-alueeseen, johon on luokiteltu 15 kurssia. Näistä kursseista on mahdollista saada 71 – 73 opintopistettä riippuen, kuinka laajasti yhden kurssista suorittaa. Kun opintopisteitä vertaa kaikkien luokiteltujen kurssien opintopistemäärään, niin tämän tietoi-alueen osuudeksi tulee 33 – 37 prosenttia.

Toiseksi suurin osa kursseista luokiteltiin järjestelmäteknikka -tietoi-alueeseen, johon luokiteltiin yhdeksän kurssia. Näiden kurssien opintopistemäärä on 39 – 54 opintopistettä riippuen siitä, miten laajasti osan kursseista

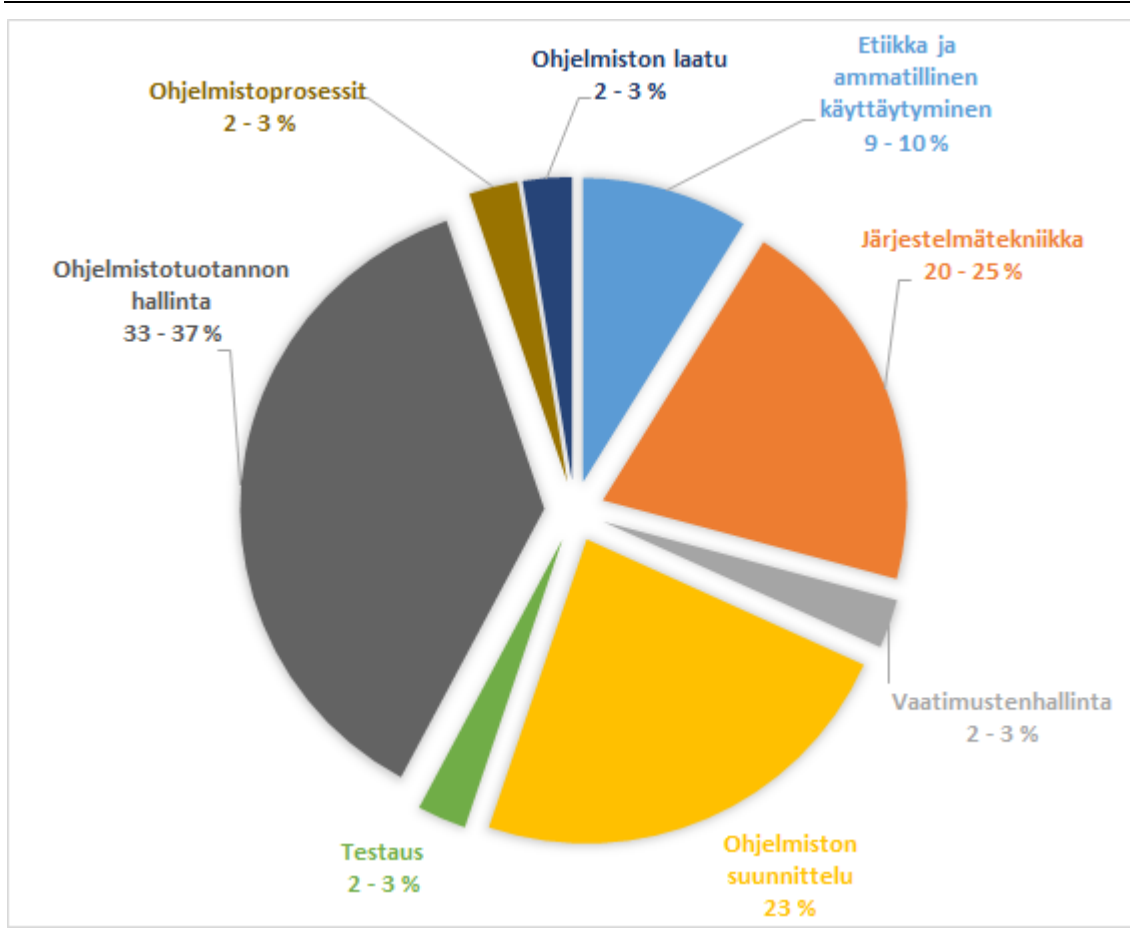
suorittaa. Tällöin verrattaessa opintopisteitä kokonaistarjontaan, niin osuudeksi tulee 20 – 25 prosenttia.

Seuraavaksi eniten kursseja on luokiteltu ohjelmiston suunnittelu-tietoalueeseen, johon luokiteltiin kahdeksan kurssia. Kursseista voi saada yhteensä 45 – 50 opintopistettä, jolloin opintopisteiden osuudeksi tulee 23 prosenttia.

Neljä kurseista on luokiteltu etiikka ja ammatillinen käyttäytyminen-tietoalueeseen, joista on mahdollista saada 17 – 22 opintopistettä. Verrattaessa luokiteltujen kurssien opintopisteisiin, niin tietoalueen osuudeksi tulee 9 – 10 prosenttia.

Yhteensä neljään tietoalueeseen luokiteltiin vain yksi kurssi, joista kustakin voi saada viisi opintopistettä. Luokitellut tietoalueet ovat vaatimustenhallinta, testaus, ohjelmistoprosessit sekä ohjelmiston laatu. Kun opintopisteitä verrataan kokonaistarjonnan opintopisteisiin, niin kunkin tietoalueen osuudeksi tulee 2 – 3 prosenttia. Luokiteltujen tietoalueiden osuudet on havainnollistettuna ympyrädiagrammin avulla (Kuva 5).

Kolmelle tietoalueelle ei pystytty luokittelemaan ollenkaan kursseja. Nämä tietoalueet ovat ohjelmiston rakentaminen, ohjelmiston ylläpito sekä konfiguraatiohallinta.



Kuva 5 Aalto-yliopiston diplomi-insinöörin opinto-oppaan luokittelu

4.5. Lahden ammattikorkeakoulun insinöörin (AMK) opinto-opas

Lahden ammattikorkeakoulun opinto-oppaasta löytyy yhteensä 58 kurssia, joista 24 luokiteltiin johonkin CCSE:n tietoaalueista ja viisi luokiteltiin menetelmäalueisiin. Kaikista luokitelluista kursseista on mahdollista saada yhteensä 123 opintopistettä. Pelkistä tietoaalueiden kursseista jaetaan yhteensä 100 opintopistettä.

Osuudeltaan suurimmat tietoaalueet ovat ohjelmistojen ydinkohdat sekä matemaattinen ja tekninen perusta. Näistä jälkimmäiseen kuuluu kahdeksan kurssia ja jälkimmäiseen seitsemän kurssia. Molempien tietoaalueiden kursseista on mahdollista saada yhteensä 31 opintopistettä. Kun opintopistemäärää vertaillaan kaikkien luokiteltujen kurssien opintopisteisiin, niin tietoaalueiden osuudeksi tulee 25 prosenttia. Jätettäessä vertailusta pois menetelmäalueiden opintopisteet, niin osuudeksi tulee 31 prosenttia.

Toiseksi suurin osuudeltaan on ammatillinen käytäntö -tietoaalue, johon kuuluu kaksi kurssia. Näistä kursseista voi saada yhteensä kymmenen opinto-

pistettä, jolloin verrattaessa kokonaistarjontaan kattaa se kahdeksan prosentin kokonaisuuden. Pelkkiin tietoalueisiin verrattaessa osuudeksi tulee kymmenen prosenttia.

Seuraavaksi suurin osuus on ohjelmiston mallinnus ja analyysi -tietoalueella, johon kuuluu myös kaksi kurssia. Näistä kursseista voi saada yhteensä kahdeksan opintopistettä, joka vastaa seitsemän prosentin osuutta kokonaistarjonnasta. Jätettäessä vertailusta pois menetelmäalueiden opintopisteet, niin osuudeksi tulee kahdeksan prosenttia.

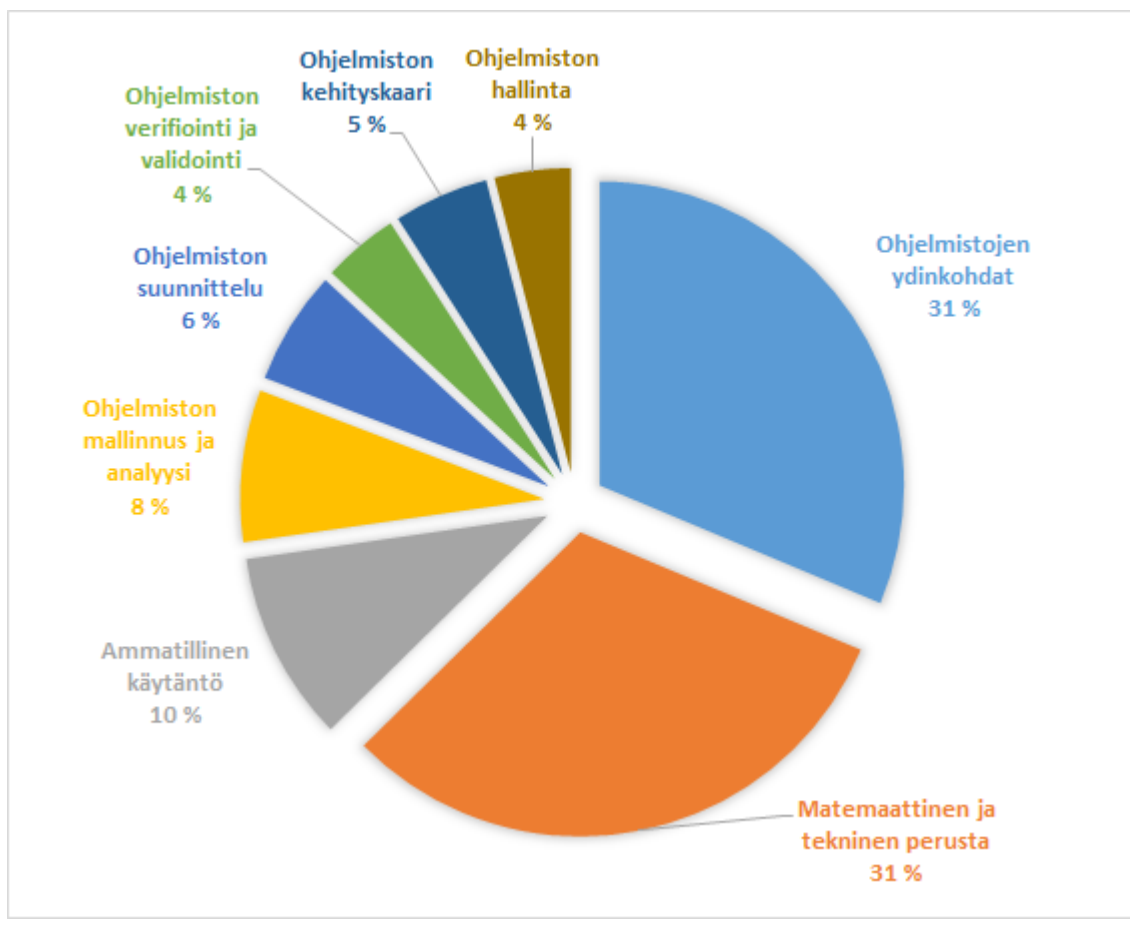
Neljänneksi suurin osuus on ohjelmiston suunnittelu -tietoalueella, johon myöskin kuuluu kaksi kurssia. Kursseista voi yhteensä saada kuusi opintopistettä, jolloin kokonaistarjontaan verrattaessa osuudeksi tulee viisi prosenttia. Pelkkiin tietoalueisiin verrattaessa osuudeksi tulee kuusi prosenttia.

Loput kolme kurssia on luokiteltuna kolmeen tietoalueeseen, joista yhdestä on mahdollista saada viisi opintopistettä. Tämä kurssi on luokiteltuna ohjelmiston kehityskaari -tietoalueeseen. Verrattaessa tietoalueen opintopisteitä kokonaistarjonnan opintopisteisiin, niin tietoalueen osuudeksi tulee neljä prosenttia. Jätettäessä menetelmäalueet vertailusta pois, niin osuudeksi tulee viisi prosenttia.

Pienimmälle osuudelle jäi kaksi tietoaluetta, joiden kursseista voi saada neljä opintopistettä. Nämä tietoalueet ovat ohjelmiston verifiointi ja validointi sekä ohjelmiston hallinta. Tietoalueiden osuudeksi tulee kolme prosenttia, kun opintopisteitä verrataan kaikkien luokiteltujen kurssien opintopisteisiin. Pelkkiin tietoalueisiin verrattaessa osuudeksi tulee neljä prosenttia. Luokiteltujen tietoalueiden osuudet on havainnollistettuna ympyrädiagrammin avulla (Kuva 6).

Kahdelle tietoalueelle ei saatu luokiteltua yhtäkään kurssia. Nämä tietoalueet ovat ohjelmiston prosessit sekä ohjelmiston laatu.

Neljälle menetelmäalueelle pystyttiin luokittelemaan kursseja. Yhdelle menetelmäalueesta luokiteltiin kaksi kurssia, joista voi saada yhteensä yhdeksän opintopistettä. Tämä menetelmäalue on multimedia-, peli- ja viihdejärjestelmät. Lopuille luokitelluille menetelmäalueille on luokiteltuna yksi kurssi kuhunkin. Kustakin kurssista on mahdollista saada viisi opintopistettä. Nämä menetelmäalueet ovat verkkokeskeiset järjestelmät, sulautetut ja reaaliaikaiset järjestelmät sekä telekommunikaatiojärjestelmät.



Kuva 6 Lahden ammattikorkeakoulun insinööri (AMK) opinto-oppaan luokittelu

4.6. Luokittelun koonti

Tampereen yliopiston ja Aalto-yliopiston kandidaatin opinto-oppaissa ja Lahden ammattikorkeakoulun opinto-oppaassa on mainittuna yhteensä 199 kurssia, joista 76 kurssia luokiteltiin CCSE:n tietoon alueisiin ja 20 menetelmäalueisiin. 76 luokitelluista kursseista saatiin luokiteltua tietoon alueisiin.

Suurimmaksi tietoon alueeksi muodostui ohjelmistojen ydinkohdat -tietoon alue, johon luokiteltiin 29 kurssia. Kun tietoon alueen opintopisteitä verrataan tietoon alueiden kokonaistarjontaan, niin tietoon alueen osuudeksi tuli 40 – 41 prosenttia.

Toiseksi suurin tietoon alue on matemaattinen ja tekninen perusta. Tähän saatiin luokiteltua 24 kurssia. Tämä vastaa 29 - 30 prosenttia tietoon alueiksi luokitelluista kursseista.

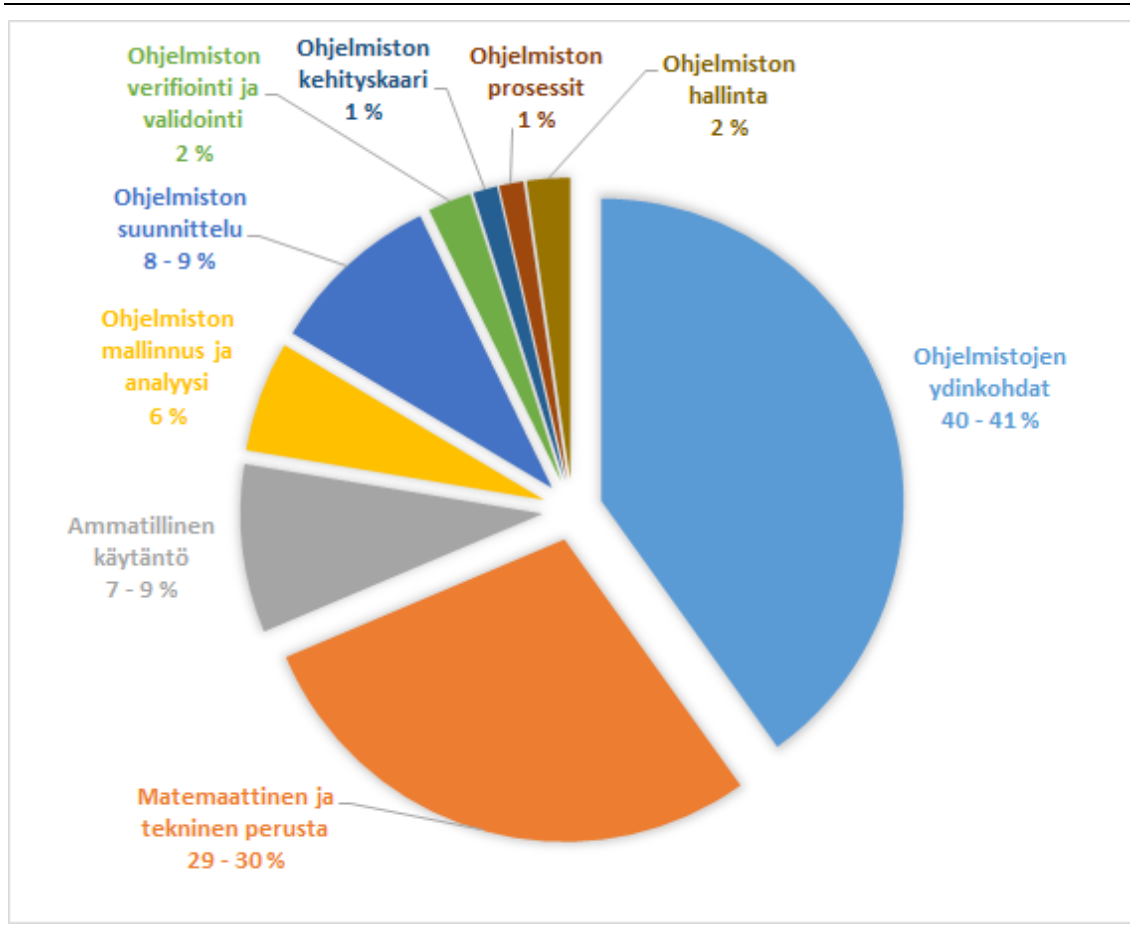
Seuraavaksi suurimmaksi tietoon alueeksi muodostui ohjelmiston suunnittelu, johon luokiteltiin yhteensä seitsemän kurssia. Kurssien opintopisteet vastaavat 8 – 9 prosentin osuutta kaikista tietoon alueiden opintopisteistä.

Neljänneksi suurin tietoalue on ammatillinen käytäntö, johon luokiteltiin viisi kurssia. Näiden kurssien opintopisteiden perusteella tietoalueen osuudeksi muodostui 7 – 9 prosenttia.

Seuraavaksi suurimmat tietoalueet ovat ohjelmiston verifiointi ja validointi sekä ohjelmiston hallinta. Kumpaakin tietoalueeseen luokiteltiin kaksi kurssia ja niiden osuus kokonaistarjonnasta on kaksi prosenttia.

Ohjelmiston kehityskaari ja ohjelmiston prosessit -tietoalueisiin luokiteltiin yksi kurssi kumpaankin. Näiden kurssien osuudeksi muodostui yksi prosentti kokonaistarjonnasta.

Mistään oppilaitoksesta ei pystytty luokittelemaan yhtään kurssia ohjelmiston laatu -tietoalueeseen. Kandidaatin opinto-oppaiden luokittelun osuudet ovat havainnollistettu ympyrädiagrammissa (Kuva 7).



Kuva 7 CCSE:n mukaan luokiteltujen kurssien osuudet

Tampereen yliopiston filosofian maisterin opinto-oppaasta ja Aalto-yliopiston diplomi-insinööri opinto-oppaasta pystyttiin luokittelemaan yhteensä 50 GSwE2009:n tietoaluetta. Suurimmaksi tietoalueeksi muodostui ohjelmistotuotannon hallinta, johon luokiteltiin 16 kurssia. Verrattaessa kurssien opin-

topisteitä kaikkien luokiteltujen kurssien opintopisteisiin tietoaalueen osuudeksi tulee 29 – 32 prosenttia.

Seuraaviksi suurimmiksi tietoaalueiksi muodostui järjestelmäteknikka ja ohjelmiston suunnittelu. Järjestelmäteknikan tietoaalueeseen luokiteltiin kymmenen kurssia, joista oli mahdollista saada 44 – 64 opintopistettä. Verrattaessa kokonaistarjontaan niin osuudeksi tulee 17 – 22 prosenttia. Ohjelmiston suunnittelu -tietoaalueeseen luokiteltiin yhdeksän kurssia, joista jaettiin yhteensä 50 – 55 opintopistettä. Tällöin tietoaalueen osuudeksi tulee 19 – 20 prosenttia.

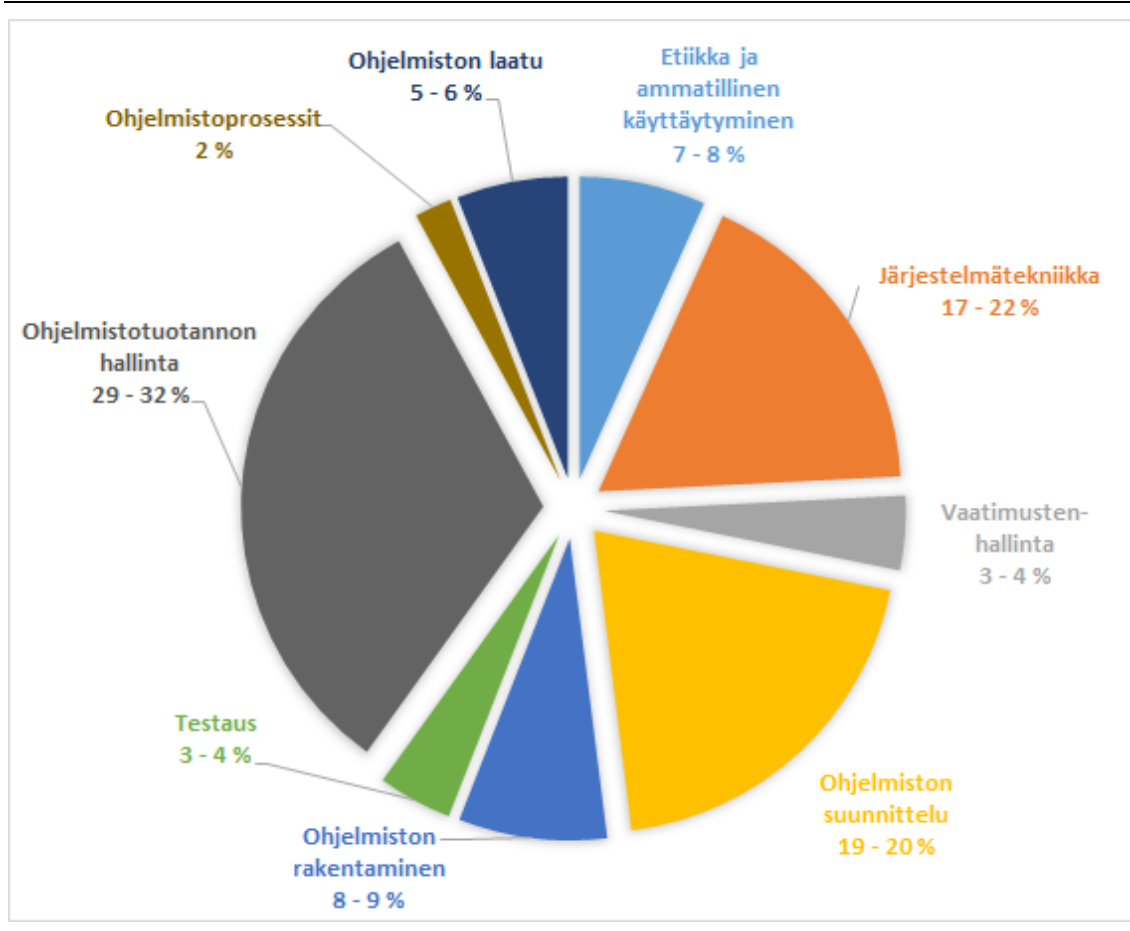
Ohjelmiston rakentamisen -tietoaalue on neljänneksi suurin, johon luokiteltiin kolme kurssia. Näistä kursseista oli mahdollista saada 20 – 25 opintopistettä, jolloin tietoaalue vastaa 8 – 9 prosentin osuutta kokonaistarjonnasta.

Viidenneksi eniten kursseja on etiikka ja ammatillinen käyttäytyminen -tietoaalueessa, johon luokiteltiin neljä kurssia. Näistä kursseista on mahdollista saada 17 – 22 opintopistettä. Osuudeksi tulee silloin 7 – 8 prosenttia.

Seuraavaksi suurimmaksi tietoaalueeksi muodostui ohjelmiston laatu, johon luokiteltiin kolme kurssia. Kursseista jaettiin yhteensä 15 opintopistettä, jolloin verrattaessa kokonaistarjonnan opintopistemäärään, niin tietoaalueen osuus on 5 – 6 prosenttia.

Kaksi seuraavaa tietoaaluetta ovat laajuudeltaan yhtä suuret. Nämä tietoaalueet ovat: vaatimusten hallinta sekä testaus. Niihin luokiteltiin kumpaankin kaksi viiden opintopisteen kurssia. Tällöin näiden tietoaalueiden osuudeksi kokonaistarjonnasta on 3 – 4 prosenttia.

Pienin osuus tuli ohjelmistoprosessit-tietoaalueelle, johon luokiteltiin vain yksi viiden opintopisteen kurssi. Osuudeksi tälle tietoaalueelle tulee kaksi prosenttia. Kahdelle tietoaalueelle ei luokiteltu yhtään kurssia. Nämä tietoaalueet olivat ohjelmiston ylläpito sekä konfiguraatiohallinta. Maisterin opintooppaiden luokittelun osuudet on havainnollistettuna ympyrädiagrammissa (Kuva 8).



Kuva 8 GSwE2009:n mukaan luokiteltujen kurssien osuudet

5. Yhteenveto

Tässä luvussa esittelen tulokset, jotka tutkimuksesta ovat ilmenneet. Käyn tulokset läpi kootusti vertailemalla opinto-oppaiden luokittelun osuuksia suositusten osuuksiin. Ensin vertaan alemman korkeakoulututkinnon tuloksia, minä jälkeen vertailen ylemmän korkeakoulututkinnon tuloksia. Tulosten esittelyn lisäksi arvioin tutkimuksen luotettavuutta.

5.1. Yhteenveto tutkimuksen tuloksista

Kandidaatin opintojen luokittelu on havainnollistettuna taulukossa 1. Taulukossa ensimmäisessä sarakkeessa on CCSE:n tietalueet ja toisessa on CCSE:n osuudet kullekin tietoaueelle. Kolmannessa sarakkeessa on Tampereen yliopiston (UTA) luokittelun osuudet. Neljännessä sarakkeessa on Aalto-yliopiston (Aalto) luokittelun osuudet ja viidennessä Lahden ammattikorkeakoulun (LAMK). Viimeisessä sarakkeessa on koottuna kaikkien korkeakoulujen luokittelun yhteenlasketut osuudet.

Taulukosta voidaan havaita, että Tampereen yliopistossa kaksi suurinta tietoaueetta vastaa hyvin suosituksia. Seuraavat viisikin järjestyksenä vastaavat hyvin suosituksia vain korkeintaan kahden järjestyspaikan erolla. Prosentuaalisten osuuksien perusteella tosin ohjelmiston verifiointi ja validointi-tietoaueen osuus on pieni verrattuna suosituksiin. Suositusten kolme pienintä tietoaueetta jää Tampereen yliopistossakin pienimmäksi, mutta näihin tietoaueille ei kuitenkaan pystytty luokittelemaan yhtään kurssia.

Taulukko 1 CCSE:n perusteella suoritettu luokittelu

Tietoaue	CCSE	UTA	Aalto	LAMK	Yhteensä
Ohjelmistojen ydinkohdat	35 %	34 - 36 %	54 %	31 %	40 - 41 %
Matemaattinen ja tekninen perusta	18 %	22 - 25 %	35 %	31 %	29 - 30 %
Ammatillinen käytäntö	7 %	7 - 13 %	4 %	10 %	7 - 9 %
Ohjelmiston mallinnus ja analyysi	11 %	9 - 11 %	0 %	8 %	6 %
Ohjelmiston suunnittelu	9 %	14 - 16 %	4 %	6 %	8 - 9 %
Ohjelmiston verifiointi ja validointi	9 %	3 - 4 %	0 %	4 %	2 %
Ohjelmiston kehityskaari	2 %	0 %	0 %	5 %	1 %
Ohjelmiston prosessit	3 %	0 %	4 %	0 %	1 %
Ohjelmiston laatu	3 %	0 %	0 %	0 %	0 %
Ohjelmiston hallinta	4 %	3 - 4 %	0 %	4 %	2 %

Aalto-yliopiston luokittelun perusteella kaksi suurinta vastaa suositusten suurimpia. Kuitenkin Aalto-yliopiston kahden suurimman tietoaueen opetus vastaa noin 90 prosenttia kokonaistarjonnasta. Tämä johtaa siihen, että muiden

tietoalueiden osuudet jää hyvin pieneksi. Huomioitavaa on myös, että suosituksen sekä kolmanneksi että neljänneksi suurimmat tietoalueet jäivät täysin ilman luokiteltuja kursseja. Kuitenkin ohjelmisto prosessit -tietoalueelle on pystytty luokittelemaan yksi kurssi. Kaiken kaikkiaan Aalto-yliopiston kurssitarjonta vastaa huonosti suosituksia.

Lahden ammattikorkeakoulun opinto-oppaan luokittelun perusteella kaksi suurinta tietoaluetta vastaa suosituksia. Matemaattinen ja tekninen perusta -tietoalueeseen on luokiteltuna suurempi osuus kuin mitä suosituksissa on osuutena. Tämä vaikuttaa myös muihinkin osuuksiin, mutta muutkin tietoalueet vastaavat hyvin suosituksien kanssa. Lahden ammattikorkeakoulussa jäi kaikista tutkituista opinto-oppaista vähiten tietoalueita ilman luokiteltuja kursseja. Lisäksi Lahden ammattikorkeakoulu oli ainoa oppilaitos, jossa käsiteltiin ohjelmiston kehityskaari -tietoaluetta.

Kaikissa oppilaitoksissa suurin tietoalue on ohjelmistojen ydinkohdat -tietoalue, joka on myös suosituksen perusteella suurin. Toisena tuleva matemaattinen ja tekninen perusta -tietoalue on selkeästi suuremmalla painotuksella kuin mitä suosituksissa on, vaikkakin on myös toiseksi suurin suosituksen mukaan. Ammatillinen käytäntö sekä ohjelmiston suunnittelu -tietoalueet ovat myös suosituksiin verrattuna samantasoisella painotuksella. Kaikki muut tietoalueet jäävät kuitenkin paljon pienemmälle painotuksella. Ohjelmiston laatu -tietoalueeseen ei pystytty mistään oppilaitoksesta luokittelemaan yhtään kurssia. Kyseinen tietoalue on suosituksen mukaan suuremmalla painotuksella kuin ohjelmiston kehityskaari -tietoalue, johon pystyttiin luokittelemaan yksi kurssi.

Maisteriopintojen luokittelu on havainnollistettuna taulukossa 2. Taulukossa ensimmäisessä sarakkeessa on GSwE2009:n tietoalueet ja toisessa on samojen suosituksen osuudet kullekin tietoalueelle. Kolmannessa sarakkeessa on Tampereen yliopiston (UTA) osuudet ja neljännessä Aalto-yliopiston (Aalto) osuudet. Viimeiseen sarakkeeseen on laskettu kummankin oppilaitoksen yhteenlasketut osuudet.

Taulukko 2 GSwE2009:n perusteella suoritettu luokittelu

Tietoalue	GSwE2009	UTA	Aalto	Yhteensä
Etiikka ja ammatillinen käyttäytyminen	2 - 4 %	0 %	9 - 10 %	7 - 8 %
Järjestelmäteknikka	5 %	8 - 14 %	20 - 25 %	17 - 22 %
Vaativuushallinta	14 - 15 %	7 - 8 %	2 - 3 %	3 - 4 %
Ohjelmiston suunnittelu	19 - 22 %	7 - 8 %	23 %	19 - 20 %
Ohjelmiston rakentaminen	2 - 5 %	33 - 36 %	0 %	8 - 9 %
Testaus	10 - 11 %	7 - 8 %	2 - 3 %	3 - 4 %
Ohjelmiston ylläpito	7 %	0 %	0 %	0 %
Konfiguraatiohallinta	5 %	0 %	0 %	0 %
Ohjelmistotuotannon hallinta	16 - 17 %	14 - 17 %	33 - 37 %	29 - 30 %
Ohjelmistoprosessit	7 %	0 %	2 - 3 %	2 %
Ohjelmiston laatu	7 %	14 - 17 %	2 - 3 %	5 - 6 %

Taulukosta nähdään, että Tampereen yliopistossa suurin painotus on ohjelmiston rakentaminen -tietoalueelle, joka on suositusten mukaan toiseksi pienimmällä painotuksella. Toiseksi suurimpana tietoalueena tulee ohjelmistotuotannon hallinta, joka on suositusten perusteella myös toiseksi suurin.

Tampereen yliopiston tuloksia selittää osittain se, että Tampereen yliopistossa ohjelmistokehityksen opinto-oppaasta voitiin luokitella vain kymmenen kurssia yhteensä. Näistä kolme kurssia luokiteltiin ohjelmiston rakentamisen -tietoalueeseen, joista kaikista voi saada jopa kymmenen opintopistettä. Tampereen yliopiston luokittelun perusteella on muutenkin vaikeata tehdä muita johtopäätöksiä, koska luokiteltuja kursseja on vähemmän kuin GSwE2009:ssä on tietoalueita. Luokittelusta voidaan kuitenkin todeta, että peräti neljään tietoalueeseen ei saatu luokiteltua yhtään kurssia, vaikka suositusten toiseksi pienin tietoalue on suurimmalla painotuksella Tampereen yliopistossa. Lisäksi esimerkiksi kurssikuvausten perusteella konfiguraatiohallinta-tietoaluetta ei käsitellä ollenkaan.

Aalto-yliopiston kaksi suurinta tietoaluetta ovat samat kuin suosituksissa, mutta vain eri järjestyksessä. Aalto-yliopistossa painotetaan myös paljon järjestelmä tekniikka -tietoaluetta, joka on suositusten mukaan vasta kahdeksas. Vaativuushallinta-tietoalue jää kuitenkin hyvin pienelle painotukselle, mikä on suosituksissa kolmantena. Lisäksi peräti kolmeen tietoalueeseen ei ole luokiteltuna yhtään kurssia.

Yhteenlasketuista osuuksista kaksi suurinta tietoaluetta ovat samat kuin suosituksissakin tosin vain eri järjestyksessä. Järjestelmäteknikka-tietoalue on suurella painotuksella toisin kuin suosituksissa, jossa kyseinen tietoalue tulee vasta kahdeksantena. Yhteenlasketuista osuuksista nähdään se, että Aalto-

yliopiston luokittelu määrää paljon tuloksia. Se kuitenkin nähdään, että kahden tietoaalueeseen, ohjelmiston ylläpito ja konfiguraatiohallinta, ei ole luokiteltu yhtään kurssia. Molemmat näistä tietoaalueista on suositusten perusteella vähintään suuremmalla painotuksella kuin kaksi muuta tietoaaluetta, etiikka ja ammatillinen käyttäytyminen sekä ohjelmiston rakentaminen. Ohjelmiston rakentamisen -tietoaalue muodostui luokittelun perusteella jopa neljänneksi suurimmaksi.

5.2. Tutkimuksen luotettavuus

Tehtäessä kvantitatiivista tutkimusta tulee tutkimuksessa tarkistaa kerätyn materiaalin ja käytettyjen mittarien luotettavuus. Tässä apuna käytetään tutkimuksen reliabiliteettia ja validiteettia. Reliabiliteetilla tarkoitetaan sitä, ovatko tutkimuksen tulokset toistettavissa. Tutkimuksen reliabiliteetti on korkea, kun kuka tahansa ja koska tahansa saa toistettua samat tulokset. Validiteetilla tarkoitetaan taas sitä, onko tutkimuksessa tutkittu tai mitattu sitä, mitä on ollut tarkoituskin.

Vaikka tässä tutkimuksessa ei käytettykään perinteisesti kvantitatiivisessa tutkimuksessa käytettäviä tilastollisia mittareita, niin on hyvä silti nostaa esille tutkimuksen reliabiliteettiongelmat. Tutkimuksen aineisto on kerätty luokittelemalla opinto-oppaista löytyviä kursseja kansainvälisten suositusten tietoaalueiden mukaan. Tutkimuksessa kurssi luokiteltiin vain yhteen tietoaalueeseen kerrallaan. Tämä heikentää tutkimuksen reliabiliteettia, koska jotkin kurssit voitaisiin luokitella useampaan eri tietoaalueeseen sisältönsä perusteella. Kaikissa oppilaitoksissa löytyy kurssi, jossa tuotetaan oikea ohjelmisto oikealle asiakkaalle. Näissä kursseissa opiskelija joutuu käyttämään tietojansa, jotka liittyvät useampaan tietoaalueeseen. Tällaiset kurssit on vaikeata luokitella yhteen tietoaalueeseen, jolloin tutkimus voi saada eri tuloksia eri tutkimus kerroilla.

Tutkimuksen validiteettiin vaikuttavat käytetyt mittarit ja mitä niillä on mitattu. Tässä tutkimuksessa mittareina käytettiin opetustarjonnan osuuksia, joita vertailtiin kansainvälisten opetussuositusten osuuksiin. Osuuksia voidaan siinäkin hyvinkin vertailla keskenään eivätkä ne laske tutkimuksen validiteettia, mutta se, miten opetustarjonnan osuudet on kerätty, laskee sitä.

Opetussuosituksia ovat luotu ohjaamaan tutkinnon rakennetta ja siihen liittyvää opetusta. Tämä tarkoittaa alemman korkeakoulututkinnon suosituksissa sitä, että suositusten ydintiedon opetukseen käytettävä aika jää alle sen, mitä tutkintoa varten tulee yhteensä käyttää. Ylemmän korkeakoulututkinnon suosituksissa ydintiedon opetukseen on määritelty puolet tutkintoon käytettävästä

ajasta. Tässä tutkimuksessa osuudet on kerätty oppilaitosten kokonaistarjonnan pohjalta. Suositusten osuuksien ja kokonaistarjonnan suositusten vertaileminen voi olla harhaanjohtavaa. Kokonaistarjonnasta opiskelija voi sopivilla valinnoilla valita itsellensä tutkinnon sisällön, joka vastaa heikosti suosituksia. Tässä tutkimuksessa käytetty luokittelumenetelmä osoittautui kuitenkin hyväksi, sillä aineistosta pystyttiin havaitsemaan, että osa tietoalueista jää pienelle painotukselle.

6. Pohdinta

Tässä luvussa pohdin tutkimuksen tuloksia, mitä ne tarkoittavat ja voiko tulok-
sista tehdä yleistettäviä johtopäätöksiä. Tarkastan myös analyysin tuloksia tut-
kimuskysymysten pohjalta. Kolmen suomalaisen korkeakoulun ohjelmistotek-
niikan opetusta on arvioitu käyttäen kansainvälisiä suosituksia. Arvioinnissa
on käytetty oppilaitosten kokonaistarjontaa, joka on luokiteltu suosituksista
löytyvien ydintietoalueiden perusteella.

Tässä tutkimuksessa lähdettiin selvittämään, miten ohjelmistotekniikan
opetus Suomessa vastaa alan kansainvälisiä opetussuosituksia. Tämä toimi en-
simmäisenä tutkimuskysymyksenä. Tutkimus suoritettiin tapaustutkimuksena,
johon valittiin joukko suomalaisia korkeakouluja. Tapauksia tutkittaessa selvi-
tettiin samalla, missä korkeakoulussa koulutus vastaa kansainvälisiä opetus-
suosituksia parhaiten. Tämä toimi samalla toisena ja kolmantena tutkimusky-
symyksenä, kun oppilaitosten alemman ja ylemmän korkeakoulututkintoja tut-
kittiin erikseen.

Avataan aluksi toista ja kolmatta tutkimuskysymystä, sillä niiden avulla
voidaan vastata ensimmäiseen kysymykseen. Tampereen yliopiston kandidaa-
tintutkinnon ja Lahden ammattikorkeakoulun insinööri (AMK) vastaavat
alemman korkeakoulun kansainvälisiä opetussuosituksia parhaiten. Kuitenkin
molemmissa oppilaitoksissa osaan suositusten ydintietoalueista ei pystytty
luokittelemaan yhtään kurssia. Näissäkin oppilaitoksissa löytyy siis kehittämis-
tä, mikäli haluttaisiin koulutuksen vastaavan paremmin kansainvälisiä opetus-
suosituksia. Aalto-yliopiston tietotekniikan kandidaatintutkinto vastaa heikosti
alemman korkeakoulututkinnon suosituksia. Aalto-yliopistossa jopa 90 pro-
senttia opinnoista luokiteltiin tässä tutkimuksessa kahteen tietoalueista, joita oli
yhtensä kymmenen.

Arvioitaessa Tampereen yliopiston ja Aalto-yliopiston alempia korkeakou-
luntutkintoja täytyy ottaa huomioon, että nämä tutkinnot valmistavat opiskeli-
jaa useampaan eri ylemmän korkeakoulututkinnon erikoistumislinjaan. Jokai-
sella tutkinnolla on olemassa erilainen tietopohja, jotka voivat erota huomatta-
vasti toisistaan. Tämän tutkimuksen pohjalta ehdotetaankin, että alemman kor-
keakoulututkintojen rakennetta uudistettaisiin siten, että opiskelijaa ohjattai-
siin enemmän valinnoissaan. Tutkinnon rakenne voisi olla sellainen, että jotkin
kurssit ovat pakollisia, mikäli haluaa jatkaa haluamaansa maisteriohjelman.
Tämä tulisi kuitenkin toteuttaa siten, että opiskelijan on mahdollista muuttaa
mieltään eikä opintojen jatkaminen vaikeudu. Tällöin pystyttäisiin varmista-

maan laadukas ohjelmistotekniikan opetus viemättä liikaa opiskelijan valinnan vapautta.

Kansainvälisten ylemmän korkeakoulututkinnon opetussuositusten perusteella Aalto-yliopiston koulutus vastaa parhaiten tutkituista korkeakouluista opetussuosituksia. Tampereen yliopiston tarjonta painottuu selkeästi ohjelmiston rakentamiseen, joka ei suositusten mukaan ole tärkeässä osassa maisteritutkinnossa. Tämä osaltaan johtuu siitä, että Tampereen yliopiston tarjonta on suppeaa ja osa ohjelmistojen rakentamisen kursseista ovat täydentäviä opintoja.

Palataan ensimmäiseen tutkimuskysymykseen: miten ohjelmistotekniikan opetus Suomessa vastaa alan kansainvälisiä opetussuosituksia. Koska tutkimus on toteutettu tapaustutkimuksena, niin tähän kysymykseen ei voida vastata täydellä varmuudella. Tutkittujen korkeakoulujen perusteella kuitenkin ohjelmistotekniikan koulutuksessa on parannettavaa. Kandidaatin opetus selkeästi painottuu ohjelmoinnin sekä matematiikan ja tilastotieteen koulutukseen. Nämä tietalueet ovat kyllä tärkeässä osassa suosituksissa, mutta painottaminen vaikuttaa heikentävästi muihin tietoalueisiin. Lisäksi opetussisältöjen perusteella ei kummassakaan oppilaitoksessa opetettu lainkaan ohjelmiston ylläpito ja konfiguraatiohallinta -tietoalueita. Tässä oli selkeimmät puutteet koulutuksen tasossa. On hyvä muistaa, että ylemmän korkeakoulututkinnon tarjoavat oppilaitokset eivät vaadi opiskelijalta työkokemusta. Tämä on suositusten mukaan pakollinen sisäänpääsyvaatimus, jonka voi korvata tarjoamalla opiskelijalle harjoittelua. Tampereen yliopistossa harjoittelusta on mahdollista saada opintopisteitä.

Tutkimustulosten arviointia vaikeuttaa kaksi asiaa. Ensimmäiseksi käytetty kurssien luokittelumenetelmä ei ottanut huomioon kurssien laajaa sisältöä. Yksi kurssi luokiteltiin vain yhden tietoalueen perusteella. On harhaanjohtavaa luokitella kurssi, jossa tuotetaan oikea ohjelma asiakkaalle, yhteen tietoalueeseen. Tällaisella kurssilla opiskelija pääsee käyttämään tietojaan ja taitojaan, jotka oikeasti kuuluvat useaan tietoalueeseen. Tämä ei näy luokittelun tuloksissa lainkaan.

Toinen arviointia vaikeuttava seikka on, että opiskelija pystyy valinnoillaan saamaan itsellensä rakennettua tutkintoonsa sisällön, joka vastaa huonosti kansainvälisiä suosituksia. Valinnaisista opinnoista opiskelijan on mahdollista valita opintoja, jotka eivät kuulu suositusten ydin tietoalueisiin. Tutkimuksessa pyrittiin ratkaisemaan tämä ongelma luokittelemaan opinto-oppaissa ilmoitettu kokonaistarjonta ja arvioimaan koulutusta tämän pohjalta. Tästä seuraa kuitenkin

kin muita ongelmia, kuten se, että osuuksien arvioinnissa ei pystytä havaitsemaan, kuinka hyvin ne vastaavat oikeata tutkinnon rakennetta ja sisältöä.

Valinnaiset opinnot ovat hyvä asia, sillä niiden avulla opiskelija saa valita opintoja oman mielenkiintonsa mukaan. Lisäksi opiskelija pystyy rakentamaan opintonsa sen mukaan, mihin haluaa myöhemmin erikoistua. Opiskelijan on kuitenkin mahdollista usein valita opintoja, jotka eivät edesauta myöhempiä opintojaan. Olisikin hyvä, jos joko tutkinnon tai opinto-oppaan rakenne olisi sellainen, että opiskelija pystyisi tekemään opintojensa kannalta parhaat valinnat. Tutkinnon rakenne voisi olla sellainen, että erikoistumisvaiheessa opiskelijalta voitaisiin vaatia ennalta määrättyjä opintoja. Tämä kuitenkin vähentää opiskelijan valinnaisuutta, joka ei välttämättä ole hyvä asia. Toinen vaihtoehto on, jos opinto-opas pystyisi opastamaan opiskelijaa valinnoissa. Esimerkiksi opinto-oppaassa voisi olla suosituksia, mitä kurseja kannattaisi valita, jos haluaa jatkaa ohjelmistotekniikan opintoja maisteritutkinnossa. Henkilökohtainen opintosuunnitelma (HOPS) -opinnot onkin tarkoitettu myös tähän tarkoitukseen, mutta olisi hyvä jos opinto-oppaat hoitaisivat tätä työtä myös.

Tämän tutkimuksen tuloksia voidaan hyödyntää ohjelmistotekniikan koulutuksen kehityksessä. Käytetty luokittelumenetelmä vaatii vielä parannusta, sillä kurssit täytyy pystyä luokittelemaan useampaan tietoonalueeseen. Parannetun luokittelun avulla pystyttäisiin koulutuksen vastaavuutta kansainvälisiin suosituksiin luotettavammin. Samalla on mahdollista tarkastaa koulutuksen sisältöä ja harkita, tarvitaanko lisää koulutusta erialueille. Mielenkiintoinen jatkotutkimusaihe olisi myös selvittää työelämän mielipide koulutuksen rakenteeseen ja sisältöön. Ollaanko työelämässä sitä mieltä, että antaako koulutus sopivat eväät työelämässä pärjäämiseen ja tarvittaisiinko koulutusta lisää joillakin osa-alueilla.

Viiteluettelo

- [1] Aalto-yliopisto, (2016). *Master's programme in computer, communication and information sciences study guide 2016 - 2017*. <http://studyguides.aalto.fi/2016-ccis/majors/software-and-service-engineering.html> Viitattu 3.3.2017.
- [2] Aalto-yliopisto, (2016). *Teknillistieteellisen kandidaattiohjelman opinto-opas 2016 - 2017*. <http://studyguides.aalto.fi/sci/2016-kand/tietotekniikka/tutkinto.html> Viitattu 3.3.2017.
- [3] Alain Abran, James W. Moore, Pierre Bourque, Robert Dupuis & Leonard L. Tripp. (2004). *Guide to the Software Engineering Body of Knowledge: 2004 version*. IEEE Computer Society.
- [4] ACM/IEEE-CS Joint Curriculum Task Force. (1991). *Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force*. Technical Report, ACM.
- [5] Rick Adcock, et al. (2009). *Curriculum Guidelines for Graduate Degree Programs in Software Engineering*. Technical Report, ACM.
- [6] Mark Ardis, Shawn Bohner, Lucia Camilloni, Diego Vallespir & Sylvia Ilieva. (2013). Using GSWE2009 in the creation and modification of graduate software engineering programs and related curricula. In: *Proceedings of the 2013 26th International Conference on Software Engineering Education and Training (CSEE&T)*, 109-118.
- [7] Mark Ardis, Scott Lucero, Pierre Bourque, James McDonald, Thomas Hilburn, Art Pyster, Kahina Lasfer & Mary Shaw. (2011). Advancing software engineering professional education. *IEEE Software* 28:4 58.
- [8] Mark Ardis & Gary Ford. (1989). *SEI Report on Graduate Software Engineering Education*. Springer. 208-249.
- [9] William F. Atchison, et al. (1968). Curriculum 68: Recommendations for academic programs in computer science: a report of the ACM curriculum committee on computer science. *Commun ACM* 11:3 151-197.
- [10] Pierre Bourque, Robert Dupuis, Alain Abran, James W. Moore & Leonard Tripp. (1999). The guide to the software engineering body of knowledge. *IEEE Software* 16:6 35.
- [11] Pierre Bourque & Richard E. Fairley. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.
- [12] Pierre Bourque, François Robert, Jean-Marc Lavoie, Ansik Lee, Sylvie Trudel & Timothy C. Lethbridge. (2002). *Guide to the Software Engineering*

Body of Knowledge (SWEBOK) and the Software Engineering Education Knowledge (SEEK) - a preliminary mapping. *In: Proceedings of the Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on*, 8-23.

- [13] Lucía Camilloni & Diego Vallespir. (2014). Using GSwE2009 for the creation and evaluation of Master's degree in Software Engineering: Case study Universidad de la República. *In: Proceedings of the 2014 XL Latin American Computing Conference (CLEI)*, 1-12.
- [14] Lucía Camilloni, Diego Vallespir & Mark Ardis. (2015). Using GSwE2009 for the Evaluation of a Master Degree in Software Engineering in the Universidad de la República. *In: Proceedings of the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, 323-332.
- [15] IEEE Computer Society, *IEEE computer society history - the 1970s*. <https://www.computer.org/web/about/history-1970s> Viitattu 20.3.2016.
- [16] IEEE STD 1002-1987. (1987). *IEEE Standard Taxonomy for Software Engineering Standards*. IEEE Computer Society.
- [17] IEEE STD 730-1975. (1975). *IEEE Standard for Software Quality Assurance Processes*. IEEE Computer Society.
- [18] Lahden ammattikorkeakoulu, (2016). *Tieto- ja viestintätekniiikan koulutus 16S, ohjelmistotekniikka opinto-opas*. <http://opinto-opas.lamk.fi/index.php/fi/68177/fi/68139/TETVT17/135/year/2017> Viitattu 3.3.2017.
- [19] Scott McCartney. (1999). *ENIAC: The Triumphs and Tragedies of the World's First Computer*. Walker & Company.
- [20] Peter Naur & Brian Randell. (1969). *Software Engineering: Report on a Conference Sponsored by the NATO Science Committee*. Scientific Affairs Division, NATO.
- [21] Opetusministeriö. (2004). *Valtioneuvoston asetus yliopistojen tutkinnoista, 794/2004*.
- [22] Elena Silva & Taylor White. (2015). The Carnegie Unit: Past, Present, and Future. *Change: The Magazine of Higher Learning* 47:2 68-72.
- [23] Tampereen yliopisto, (2016). *Master's degree programme in software development opinto-opas 2016 - 2017*. <https://www10.uta.fi/opas/tutkintoOhjelma.htm?rid=9245&uiLang=fi&lang=fi&lvv=2016> Viitattu 3.3.2017.
- [24] Tampereen yliopisto, (2016). *Tietojenkäsittelytieteiden tutkinto-ohjelman opinto-opas 2016 - 2017*.

<https://www10.uta.fi/opas/tutkintoOhjelma.htm?rid=9254&uiLang=fi&lang=fi&lvv=2016> Viitattu 3.3.2017.

- [25] The Joint Task Force on Computing Curricula. (2004). *Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering*. ACM.
- [26] John W. Tukey. (1958). The Teaching of Concrete Mathematics. *The American Mathematical Monthly* 65:1 1-9.
- [27] Robert K. Yin. (2013). *Case Study Research: Design and Methods*. Sage Publications.