



Ohjelmistokehityksen osaamistarpeet työelämässä 2020–2024: katsaus päiväkirjamuotoisiin AMK-opinnäytetöihin

Lotta Seppälä

Katarina Zec

Haaga-Helia ammattikorkeakoulu

Tradenomi, tietojenkäsittely

Opinnäytetyö

2025

Tiivistelmä

Tekijät Lotta Seppälä, Katarina Zec
Tutkinto Tradenomi, tietojenkäsittely
Opinnäytetyön nimi Ohjelmistokehityksen osaamistarpeet työelämässä 2020–2024: katsaus päiväkirjamuotoisiin AMK-opinnäytetöihin
Sivu- ja liitesivumäärä 58 + 2
<p>Opinnäytetyössä tarkasteltiin ohjelmistokehityksen osaamistarpeita työelämässä vuosina 2020–2024. Tavoitteena oli selvittää, millaisia taitoja ja teknologioita ohjelmistokehittäjät käyttivät työtehtävissään. Tutkimusaineistona käytettiin Haaga-Helia ammattikorkeakoulun vuosina 2020–2024 julkaistuja ohjelmistokehittäjän roolissa työskennelleiden opiskelijoiden päiväkirjamuotoisia opinnäytetöitä.</p> <p>Tietoperustassa tarkasteltiin osaamisen käsitteitä, kvalifikaatioluokituksia, ohjelmistoalan suuntaa Suomessa sekä kansainvälisiä viitekehyksiä. Tutkimus noudatti mixed methods research -lähestymistapaa (MMR), ja siinä yhdistettiin induktiivinen sisällönanalyysi osaamistarpeisiin liittyvien aiheiden esiintymistiheyksien kvantifiointiin.</p> <p>Tulokset osoittivat, että aineiston perusteella keskeisimpiä osaamisia olivat muun muassa ohjelmointikielien, versionhallinta, testaaminen, tietokannat sekä tietoliikenne ja rajapinnat. Eniten mainittuja ohjelmointikieliä olivat JavaScript, TypeScript ja Java. Myös pehmeät taidot näkyivät monipuolisesti tuloksissa. Näistä eniten mainittuja olivat viestintä- ja vuorovaikutustaidot ja ongelmanratkaisutaidot.</p> <p>Tulosten vertailu viitekehyksiin ja aiempaan tutkimustietoon osoitti, että tulokset olivat pääasiassa linjassa kansainvälisten suositusten kanssa ja että keskeiset käytännön osaamiset olivat samoja. Suurimmat erot vertailussa ilmenivät siinä, että viitekehyksissä ja aiemmissa tutkimuksissa painotettiin myös abstrakteja ja teoreettisia osaamisia, kun tässä tutkimuksessa korostuivat käytännön osaamiset. Vertailu aiempiin opinnäytetöihin ja kyselyihin oli myös pääosin linjassa tulosten kanssa. Ajankohtaisia trendejä oli huomattavissa DevOpsin, pilvipalveluihin, tekoälyyn ja suuriin kielimalleihin liittyen.</p>
Asiasanat Ohjelmistokehitys, työelämä, osaamistarpeet, osaaminen, ammattitaito

Sisällys

1	Johdanto	1
1.1	Tutkimuksen tausta ja tavoitteet	1
1.2	Keskeiset käsitteet	3
2	Tietoperusta	5
2.1	Osaaminen.....	5
2.2	Kvalifikaatioluokitukset osaamistarpeiden ennakkoinnin perustana.....	6
2.3	Osaamistarpeiden ennakkointi	8
2.4	Alan suunta Suomessa	8
2.5	Ohjelmistokehityksen kansainväliset viitekehykset	9
2.6	Aiempi aiheeseen liittyvä tutkimus.....	14
2.7	Aiemmat opinnäytetyöt ja kyselyt aiheesta	16
3	Empiirinen osa.....	19
3.1	Aineisto	19
3.1.1	Päiväkirjamuotoiset opinnäytetyöt tutkimusaineistona.....	19
3.1.2	Aineiston rajaus	20
3.2	Tutkimusmenetelmien esittely	21
3.3	Tutkimuksen kulku.....	23
4	Tutkimuksen tulokset.....	27
4.1	Ohjelmistokehityksen keskeiset aiheet aineistossa.....	27
4.2	Ohjelmointikielet.....	30
4.3	Tietokannat	31
4.4	Viitekehykset	32
4.5	Testaus ja testausteknologiat	33
4.6	Pilvipalvelut	35
4.7	Toistuvat ohjelmistokehityksen tehtävät	36
4.8	Pehmeät taidot.....	37
5	Pohdinta.....	39
5.1	Tulokset ja tietoperusta	39
5.1.1	Tulosten vertailu: Kvalifikaatioluokitus ja osaamisen typologia	39
5.1.2	Tulosten vertailu: Raportit ja viitekehykset	40
5.1.3	Tulosten vertailu: Aiemmat tutkimukset.....	43
5.1.4	Tulosten vertailu: Aiemmat opinnäytetyöt ja kyselyt	47
5.2	Tulokset ja ajankohtaiset trendit	49
5.3	Tutkimuksen luotettavuuden varmistaminen.....	50
5.4	Jatkotutkimus	52

Lähteet.....	53
Liitteet.....	59
Liite 1. Aineistosta sisällönanalyysissä muodostetut aiheet	59

1 Johdanto

1.1 Tutkimuksen tausta ja tavoitteet

Ohjelmistoalan liikevaihto Suomessa oli vuonna 2023 noin 3,23 miljardia Yhdysvaltain dollaria (noin 2,85 miljardia euroa), ja sen ennustetaan kasvavan viidessä vuodessa noin 3,85 miljardiin dollariin (noin 3,4 miljardiin euroon) (Statista 2024; valuuttamuunnokset kirjoittajien laskemia). Samana vuonna ala työllisti yli 71 000 henkilöä (Tilastokeskus 2024a). Vaikka alan näkymät ovat hie- man heilahdelleet viime vuosina suhdanteiden myötä, eivät taustalla vaikuttavat digitalisaation kal- taiset megatrendit ole häviämässä mihinkään. Osaamistarpeet kehittyvät nopeasti, ja yksi merkittä- vimmistä kasvua rajoittavista tekijöistä viime vuosina on ollut pula osaavista työntekijöistä (Ek 2020, 25). Tästä syystä ohjelmistoalan osaamistarpeiden tarkastelu on erityisen tärkeää.

Aihe on meille myös henkilökohtaisesti merkityksellinen, sillä olemme opintojemme loppuvai- heessa olevia alanvaihtajia ilman aiempaa kokemusta tai verkostoja IT-alalta. Oman ammatillisen suuntautumistemme pohdintoja varten on hyödyllistä saada tietoa siitä, millaista osaamista ohjel- mistoalalla tällä hetkellä arvostetaan, ja millaisia taitoja tulevaisuudessa tarvitaan. Vastaavassa ti- lanteessa on todennäköisesti myös suuri joukko muita opiskelijoita, mikä osaltaan tekee aihees- tamme entistä merkityksellisemmän tarkastelun kohteen.

Lisäksi alan jatkuva teknologinen kehitys edellyttää, että myös koulutussisällöt pysyvät ajan tasalla, mikä tekee opetussuunnitelmien suunnittelusta entistä vaativampaa. Opetussuunnitelmia pitää pystyä tarpeen vaatiessa myös muokkaamaan, jotta koulutus pystyy vastaamaan ajankohtaisiin osaamistarpeisiin. Uusia teknologioita ja käytäntöjä on tarpeen sisällyttää opetukseen harkitusti, ja koulutuksen painotuksia voi olla tarpeen mukauttaa vastaamaan työelämän jatkuvasti muuttuvia osaamistarpeita. (Warr ym. 2023, 129–130.)

Näistä lähtökohdista tutkimuksemme keskittyy ohjelmistokehityksen osaamistarpeisiin. Tarkaste- lemme vuosina 2020–2024 julkaistuja päiväkirjamuotoisia opinnäytetöitä ja tutkimme, mitkä osaa- miset ja teknologiat ovat aineistomme perusteella olleet laajimmin käytössä. Päiväkirjamuotoinen opinnäytetyö on opinnäytetyötyyppi, jonka opiskelija toteuttaa raportoiden ja analysoiden työtehtä- viään työnsä ohessa. Tavoitteena on opiskelijan ammatillinen kasvu ja kehittyminen. (Opinnäyte- työkoordinaattorit 2022, 2.) Meidän tutkimuksemme tavoitteena on syventää ymmärrystä ohjelmis- tokehityksen ajankohtaisista osaamistarpeista tarkastelemalla työelämässä toimineiden opiskelijoi- den käytännön kokemuksia. Tämän perusteella asetimme tutkimuskysymykseksemme:

Millaisia osaamisia ja teknologioita ohjelmistokehityksen tehtävissä on esiintynyt päiväkirjamuotoi- sissa opinnäytetöissä vuosina 2020–2024?

Tutkimuksemme sai alkunsa Haaga-Helia ammattikorkeakoulun lehtorin Outi Virkin ideasta, joka syntyi hänen arvioidessaan opinnäytetöitä. Kuulimme Outin ideasta keskustelussamme opinnäyte-työkoordinaattorimme, Haaga-Helian lehtorin Mirja Jaakkolan kanssa. Keskustelussamme Mirja mainitsi, että päiväkirjamuotoiset opinnäytetyöt voisivat auttaa tuottamaan uutta tietoa ohjelmisto-
alan osaamistarpeista. Päiväkirjamuotoiseen aineistoon tutustumalla voisi saada tietoa siitä, millai-
sia taitoja uransa alussa olevat ohjelmistoalan työntekijät ovat todella käyttäneet ja tarvinneet työ-
sään.

Tiedossamme ei ollut, että kukaan olisi tähän mennessä tarkastellut aihetta käyttämällä aineistona
juuri päiväkirjaopinnäytetöitä. Aiemmat aiheesta tehdyt tutkimukset ja opinnäytetyöt ovat perustu-
neet kyselyihin ja haastatteluihin tai esimerkiksi työpaikkailmoituksista kerättyihin tutkimusaineistoi-
hin. Päiväkirjamuotoisista opinnäytetöistä koostuva aineisto tarjoaa syvemmän näkökulman arjen
työkaluihin ja siihen, millaista osaamista työelämässä todella tarvitaan. Tämän ainutlaatuisen lä-
hestymistavan vuoksi koimme tutkimusaiheen paitsi kiinnostavaksi, myös poikkeuksellisen merki-
tykselliseksi – sekä alan osaamistarpeiden syvällisemmän ymmärtämisen että oman ammatillisen
suuntautumisemme kannalta.

Tutkimuksemme tulokset voivat hyödyttää oppilaitoksia ja opiskelijoita, sillä tutkimus tuottaa uutta
tietoa, joka voi auttaa tunnistamaan työmarkkinoilla kysytyjä taitoja. Tuloksista voi olla hyötyä
myös opiskelijoiden ja alanvaihtajien urasuunnittelun tukena. Tutkimuksemme liittyy myös elinikäi-
sen oppimisen teemaan, koska se voi auttaa alalla työskenteleviä arvioimaan osaamisensa ajanta-
saisuutta ja tunnistamaan uusia kehitysalueita. Näin ollen tutkimuksemme voi parhaimmillaan edis-
tää ammatillista kehittymistä ja vahvistaa koulutuksen ja työelämän välistä vuoropuhelua.

Tutkimuksemme on luonteeltaan laadullinen, ja tutkimusmenetelmänä käytämme induktiivista si-
sällönanalyysia. Tarkastelemme lisäksi taitojen ja teknologioiden esiintyvyyttä määrällisesti sekä
määrien kehitystä tutkittavalla ajanjaksolla. Näiltä osin tutkimusote lähestyy kvantitatiivista analy-
ysia. Kvalitatiivisen ja kvantitatiivisen otteen yhdistäminen kuuluu niin sanottuun mixed methods re-
search -lähestymistapaan (MMR), jossa ajatellaan, että näiden kahden yhdistelmällä voidaan
saada aikaan hedelmällisempiä tuloksia kuin jommallakummalla tavalla yksinään (Tuomi & Sara-
järvi 2018, luku 2.5).

Tutkimustamme ei ole tarkoitettu kattavaksi kokonaisanalyysiksi ohjelmistoalan osaamistarpeista,
sillä aineistomme perustuu rajattuun otokseen opinnäytetöitä, jotka heijastavat tietyllä tavalla vali-
koituneita työtehtäviä tietyllä ajanjaksolla. Näin ollen tutkimuksemme ei kuvaa kattavasti kaikkien
yritysten yleisimmin käyttämiä teknologioita. Tulosten perusteella voidaan kuitenkin tehdä alustavia
johtopäätöksiä osaamistarpeista. Jatkotutkimukset erilaisilla tutkimusaineistoilla ja tutkimusotteilla
voisivat laajentaa ymmärrystä aiheesta ja tarjota syvällisempää tietoa aiheesta.

1.2 Keskeiset käsitteet

Termi	Määritelmä
Ohjelmistokehitys	Ohjelmistojen mitattavissa oleva ja järjestelmällinen kehittäminen, ylläpito ja hallinta. Perustuu tieteellisen ja teknologisen tietämyksen sekä insinööriperiaatteiden soveltamiseen ohjelmistojen suunnittelussa, toteutuksessa, testauksessa ja dokumentoinnissa. Tavoitteena on tuottaa laadukkaita, luotettavia ja kustannustehokkaita ohjelmistoratkaisuja, jotka täyttävät käyttäjien ja sidosryhmien tarpeet. (ISO/IEC/IEEE 24765:2017, 422.)
Päiväkirjamuotoinen opinnäytetyö	Opinnäytetyötyyppi, jossa kirjoittaja oman työnsä ohessa tarkastelee omaa ammatillista kehittymistään päivittäisten havaintojen ja viikoittaisten analyysien kautta, nojaten ennakkoon valittuihin teemoihin ja hyödyntäen alan tietoperustaa (Opinnäytetyökoordinaattorit 2022, 1–3).
Osaaminen	Yksilön kokemuksen tai koulutuksen kautta hankkimat tiedot, taidot ja valmiudet, jotka mahdollistavat teoreettisen tai käytännöllisen ymmärryksen jostakin aihealueesta (Soanes & Stevenson 2005; knowledge).
Kompetenssi	Kyky suorittaa jokin tehtävä menestyksekkäästi ja tehokkaasti (Soanes & Stevenson 2005; competence).
Osaamistarve	Työmarkkinoilla ilmenevä kysyntä tietyille tiedoille, taidoille ja kyvykkyyksille esimerkiksi maantieteellisen alueen tai tietyn toimialan sisällä (Cedefop 2010; skill needs).
Affektis-konatiiviset valmiudet	Yksilön tunneperäiset (affektiiviset) ja tahdonalaiset (konatiiviset) ominaisuudet, jotka ohjaavat hänen suhtautumistaan, motivaatiotaan ja pyrkimystään toimintaan. Affektiivinen kattaa mielialat, tunteet sekä asenteet ja konatiivinen liittyy tahtoon, tavoitteellisuuteen ja toiminnan suuntaamiseen. (Soanes & Stevenson 2005; affective; conation; conative.)
Kvalifikaatio	Pätevyyden saavuttaminen tai sen virallinen tunnustaminen, joka perustuu vaatimusten täyttämiseen, saavutuksiin tai ominaisuuksiin, jotka tekevät yksilöstä soveltuvan tiettyyn tehtävään, ammattiin tai toimintaan; voi tarkoittaa myös ehtoa tai vaatimusta, joka on täytettävä ennen oikeuden tai aseman saamista. (Soanes & Stevenson 2005; qualification.)
Kvalifikaatioluokitus	Kvalifikaatioluokitus on kokoelma yksittäisistä kvalifikaatioista, jotka on järjestetty luokitusjärjestelmän mukaan (Hanhinen 2010, 150).
Kovat taidot vs. pehmeät taidot	Kovat taidot (hard skills) ovat mitattavia teknisiä valmiuksia, jotka opitaan usein koulutuksen ja harjoittelun kautta. Pehmeät taidot (soft skills) ovat usein vuorovaikutukseen, itseohjautuvuuteen ja tunneälyyn liittyviä valmiuksia, jotka kehittyvät kokemuksen kautta. Vaikka kovat ja pehmeät taidot usein erotellaan käsitteellisesti toisistaan, niiden välinen vuorovaikutus on keskeistä yksilön ammatillisen osaamisen kokonaisvaltaisessa ymmärtämisessä. (Lamri & Lubart 2023, 3–4.)
Yleiset vs. spesifit taidot	Yleiset taidot ovat taitoja, joiden ajatellaan lisäävän yksilön arvoa työmarkkinoilla – eri yrityksissä, toimialoilla ja ammateissa. Spesifit taidot puolestaan lisäävät yksilön arvoa vain siinä organisaatiossa, jossa ne on hankittu, eikä niitä voida välttämättä hyödyntää muualla ilman, että niiden arvo heikkenee. (Balcar ym. 2011, 21.)

Termi	Määritelmä
Induktiivinen sisälönanalyysi	Laadulliseen tutkimusperinteeseen kuuluva aineistolähtöinen analyysimenetelmä, jossa aineistossa esiintyvistä yksittäisistä asioista muodostetaan laajempia, yleisiä luokkia tai teemoja (Tuomi & Sarajärvi 2018, luku 4.2).
Hermeneuttinen esiyymmärrys	Ennakolta olemassa oleva ymmärrys, käsitys tai ajattelutapa tutkittavasta ilmiöstä, joka ohjaa sen tulkintaa ja ymmärtämistä. Tutkimuksessa sillä tarkoitetaan niitä tapoja, joiden varassa tutkija luonnostaan hahmottaa kohdettaan jo ennen varsinaista analyysia. (Laine 2018, luku Hermeneuttinen ulottuvuus.)
Mixed methods research (MMR)	Tutkimusote, jossa yhdistetään laadullisia ja määrällisiä menetelmiä samassa tutkimuksessa. Tavoitteena on saada aikaan parempaa tutkimusta kuin kummallakaan yhdellä menetelmällä ilman toista. (Tuomi & Sarajärvi 2018, luku 2.5.)
Triangulaatio	Lähestymistapa, jossa tutkittavaa ilmiötä tarkastellaan monesta näkökulmasta tutkimuksen luotettavuuden lisäämiseksi. Tutkimuksessa voidaan esimerkiksi yhdistää erilaisia aineistoja, teorioita tai menetelmiä, tai useampi kuin yksi tutkija voi osallistua tutkimuksen tekoon. (Tuomi & Sarajärvi 2018, luvut 6.4 ja 6.5.)
Tekoäly	Tietokonejärjestelmien teoria tai kehitys, joiden avulla voidaan suorittaa tehtäviä, jotka normaalisti vaativat ihmisen älykkyyttä, kuten visuaalinen havaitseminen, puheentunnistus, päätöksenteko ja kielten välinen kääntäminen (Soanes & Stevenson 2005; artificial intelligence).
Suuret kielimallit (LLM, Large Language Models)	Generatiiviset matemaattiset mallit, jotka on koulutettu suurella määrällä ihmisten tuottamaa tekstiä. Niiden toiminta perustuu sanojen ja merkkien (ns. tokenien) tilastollisen jakauman mallintamiseen, eikä niillä ole ymmärrystä, tietoisuutta tai uskomuksia ihmisen tapaan. (Shanahan 2024, 69–71.)
Pilvipalvelut	Malli, jolla mahdollistetaan kaikkialla saatavilla oleva, helppokäyttöinen ja tarpeen mukaan toimiva verkkoyhteys jaettuun, mukautettavaan laskentaresurssien kokonaisuuteen. Resursseja (esimerkiksi verkot, palvelimet, tallennus, sovellukset ja palvelut) voidaan ottaa käyttöön ja vapauttaa nopeasti, vähäisellä hallinnallisella vaivalla tai ilman suoraa vuorovaikutusta palveluntarjoajan kanssa. (Mell & Grance 2011, 2.)
DevOps	Ohjelmistokehitysmenetelmä, jonka tavoitteena on kuroa umpeen kehityksen (Dev) ja operatiivisten toimintojen (Ops) välinen kuilu. Korostaa viestintää ja yhteistyötä, jatkuvaa integraatiota, laadunvarmistusta sekä ohjelmistojen toimitusta hyödyntäen automaattista käyttöönottoa ja joukkoa kehityskäytäntöjä. (Jabbari, Ali, Petersen & Tanveer 2016.)
Ketterät menetelmät	Ohjelmistokehitysmenetelmät, jotka perustuvat iteratiiviseen ja inkrementaaliseen kehitykseen, tiheään tarkasteluun ja jatkuvaan mukautumiseen. Vaatimukset ja ratkaisut muotoutuvat yhteistyössä itseohjautuvien, monialaisista asiantuntijoista koostuvien tiimien ja sidosryhmien palautteen kautta. (ISO/IEC/IEEE 24765:2017, 15.)
Tyypitys (ohjelmointikielissä)	Ohjelmointikielen mekanismi, jolla määritetään ja valvotaan tietojen tyyppejä. Tyypitys perustuu tietotyyppeihin (data type), jotka koostuvat arvojoukosta ja siihen liittyvistä operaatioista. Vahvasti tyypitetyissä (strongly typed) kielissä tietotyypit on ilmoitettava eksplisiittisesti, minkä avulla kieli estää virheelliset operaatiot yhteensopimattomien tietotyyppien välillä. (ISO/IEC/IEEE 24765:2017, 120, 443.)

2 Tietoperusta

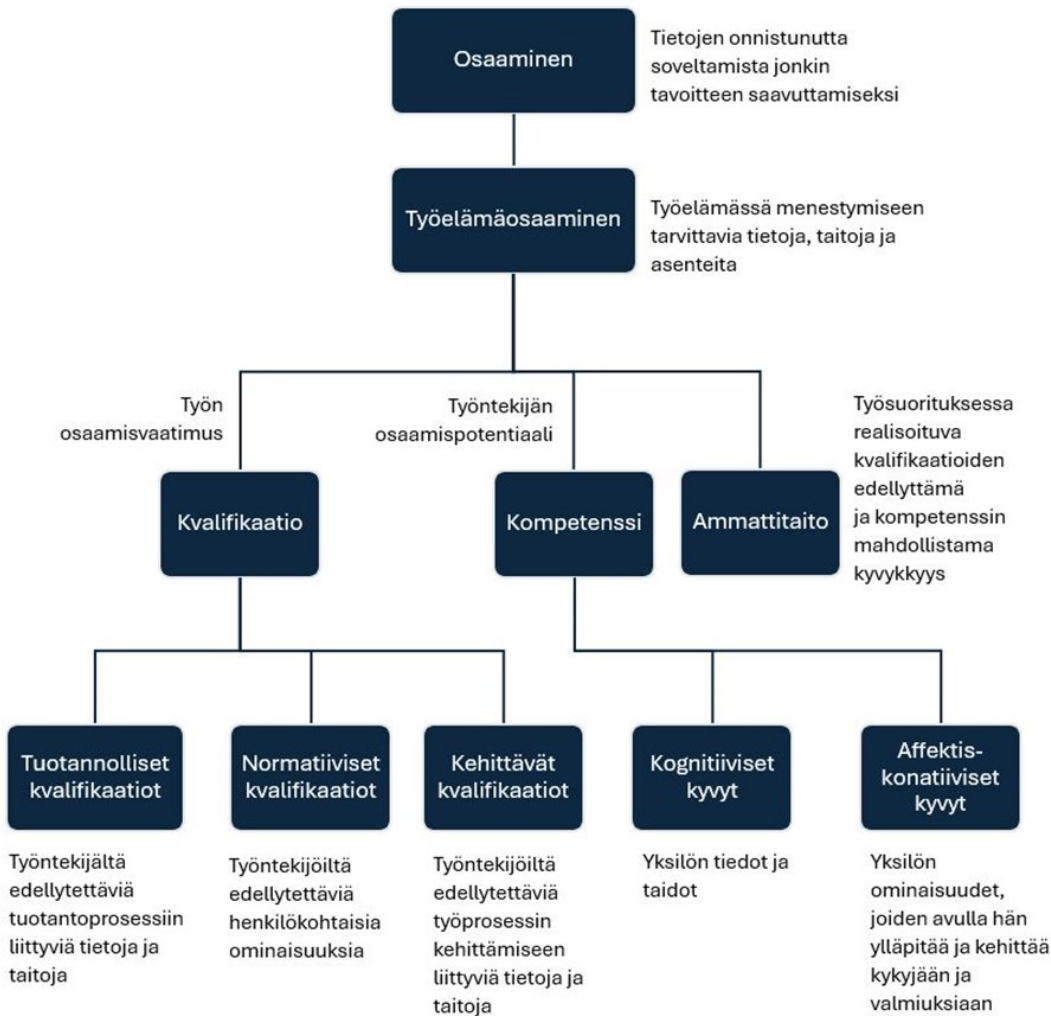
2.1 Osaaminen

Osaaminen on monimuotoinen ja kontekstista riippuva käsite. Niiniluoto (1996, 49) määrittelee osaamisen kyvyiksi ja valmiuksiksi, jotka liittyvät ihmisten ja eläinten käyttäytymiseen. Toiminnallisen osaamisen ja tiedon välillä on Niiniluodon (1996, 53) mukaan ero: osaaminen ei ole pelkästään tiedon hallintaa, vaan myös kykyä soveltaa tätä tietoa käytännön tilanteissa. Englannin kielessä osaaminen määritellään yksilön kokemuksen tai koulutuksen kautta hankkimiksi tiedoiksi, taidoiksi ja valmiuksiksi, jotka mahdollistavat teoreettisen tai käytännöllisen ymmärryksen jostakin aihealueesta (Soanes & Stevenson 2005; knowledge). Kompetenssi taas määritellään englannin kielessä kyvyksi suorittaa tehtävä onnistuneesti tai tehokkaasti (Soanes & Stevenson 2005; competence).

Taina Hanhinen (2010, 52–53) toteaa väitöskirjassaan, että kompetenssi on tieteellisessä kirjallisuudessa yleisimmin käytetty termi osaamisen kuvaamiseen. Hänen mukaansa osaaminen tarkoittaa tiedon onnistunutta soveltamista jonkin tietyn tavoitteen saavuttamiseksi, ja siihen sisältyy kolme keskeistä tekijää: toiminta, tieto ja onnistuminen. Työelämäosaaminen puolestaan viittaa niihin tietoihin, taitoihin ja asenteisiin, joita tarvitaan työssä menestymiseen. (Hanhinen 2010, 48–49.)

Hanhinen määrittelee työelämäosaamisen keskeisiksi käsitteiksi kompetenssin, kvalifikaation ja ammattitaidon. Kompetenssi tarkoittaa yksilön kognitiivisia kykyjä sekä affektis-konatiivisia valmiuksia suoriutua työn vaatimuksista. Kvalifikaatiot ovat työelämän asettamia vaatimuksia osaamiselle, jota työntekijältä edellytetään. Ammattitaito on puolestaan kvalifikaatioiden ja kompetenssin mahdollistamaa kyvykkyyttä, joka realisoituu työntekijän työsuorituksessa. (Hanhinen 2010, 96–98.)

Osaamisen käsitteen määrittely on tärkeää, jotta voimme tarkastella sitä johdonmukaisesti. Tässä tutkimuksessa osaaminen ymmärretään laajana kattokäsitteenä, jonka alle kuuluvat työelämäosaaminen, kvalifikaatio, kompetenssi ja ammattitaito. Näiden käsitteiden välisiä suhteita ja ulottuvuuksia havainnollistetaan kuvassa 1 ja taulukossa 1. Kuvassa 1 esitetään tiivistetysti, miten osaamisen käsitteet asettuvat hierarkkisesti toisiinsa nähden.



Kuva 1. Osaamisen käsitteiden tiivistetty hierarkkinen asema toisiinsa nähden (mukailen Hanhinen 2010, 97)

2.2 Kvalifikaatioluokitukset osaamistarpeiden ennakoinnin perustana

Opetushallituksen raportti (Leveälähti & Nyssölä 2019) käsittelee tulevaisuuden osaamistarpeita Suomessa ja niiden muutosta työmarkkinoiden kehityksen myötä. Raportti pohjautuu Osaamisen ennakointifoorumin (OEF) ennakointiprosessiin, jossa hyödynnetään kvalifikaatioluokitusta ja skenaariopohjaista analyysiä. Ennakointiaineistona käytetään osaamistarvekyselyä sekä sen pohjalta toteutettua työpajatyöskentelyä. Näiden menetelmien avulla raportissa ennakoidaan, mitkä osaamiset korostuvat eri toimialoilla ja työtehtävissä vuoteen 2035 mennessä. (Leveälähti & Nyssölä 2019, 5.)

Raportin mukaan osaamistarpeiden ennakkoinnin pohjana voidaan pitää kolmitasoista kvalifikaatio- luokitusta. Kvalifikaatioluokitus jäsentää osaamiset geneerisiin osaamisiin, yleisiin työelämäosaami- siin sekä ammattialakohtaisiin osaamisiin. Geneeriset osaamiset ovat laaja-alaisia taitoja, jotka parantavat yksilön kilpailukykyä työmarkkinoilla. Yleiset työelämäosaamiset, kuten viestintä- ja ICT-taidot, ovat puolestaan toimialariippumattomia ja tukevat ammatillista kehittymistä. Ammat- tialakohtaiset osaamiset ovat tiettyihin ammatteihin tai organisaatioihin liittyviä taitoja, jotka voivat sisältää myös geneerisiä ja yleisiä työelämäosaamisia. (Leveälähti & Nyysölä 2019, 18–20.)

Kvalifikaatioluokituksia analysoidaan myös Euroopan unionin (Balcar ym. 2011) raportissa, joka tarkastelee osaamisen siirrettävyyttä ja sen merkitystä työmarkkinoilla sekä yksilöiden työllistettä- vyydessä. Raportin mukaan kvalifikaatioluokitus perustuu kahteen ulottuvuuteen, joihin jaottelu voi auttaa hahmottamaan osaamisia, vaikka todellisuudessa työelämässä rajat eivät ole yhtä selkeät.

Jaottelu jäsennetään osaamisen typologian kautta, joka esitellään taulukossa 1. Olemme lisänneet taulukkoon esimerkkejä taidoista havainnollistamaan osaamisen ulottuvuuksia.

Taulukko 1. Osaamisen typologia (mukaillen Balcar ym. 2011, 22)

	Kovat taidot	Pehmeät taidot
Yleiset osaamiset	Yleiset kovat taidot, kuten talousosaa- minen, ICT-taidot ja kielitaidot	(Yleiset ja spesifit) pehmeät tai- dot, kuten kommunikointi, esiinty- minen, stressinsietokyky ja ongel- manratkaisukyky
Spesifit osaamiset	Spesifit kovat taidot, kuten hitsaustai- dot, ohjelmointitaidot tai kemiallisten prosessien hallinta	

Ensimmäinen ulottuvuus jaotellaan kahteen osaan: yleisiin, laajasti hyödynnettäviin osaamisiin ja spesifeihin, esimerkiksi tiettyyn työtehtävään liittyviin osaamisiin. Toisessa ulottuvuudessa osaami- nen jakautuu myös kahteen osaan: niin kutsuttuihin koviin (hard skills) ja pehmeisiin (soft skills) tai- toihin. Kovat taidot voivat olla työspesifejä, helposti havaittavissa ja mitattavissa sekä koulutuksen avulla omaksuttavia taitoja, kun taas pehmeät taidot ovat vaikeammin mitattavia, asenteisiin ja vuorovaikutukseen liittyviä taitoja. Vaikka pehmeät taidot voidaan teoriassa jakaa yleisiin ja spesi- feihin taitoihin, ne kuvataan raportin mukaan kirjallisuudessa pelkästään yleisinä. Tämä tarkoittaa, että pehmeät taidot eivät ole sidottuja tiettyyn ammattiin tai toimialaan, vaan ne ovat siirrettävissä moniin eri työtehtäviin ja aloille. (Balcar ym. 2011, 22.)

Hyödynnämme tutkimuksessamme taulukossa edellä mainittuja kvalifikaatioluokituksia ja osaami- sen jäsentelyä. Näiden rakenteet mahdollistavat aineistossa esiintyvien osaamisten johdonmukai- sen tarkastelun. Ne voivat auttaa meitä myös tunnistamaan ja luokittelemaan aineistostamme kes- keisiä osaamisia.

2.3 Osaamistarpeiden ennakointi

Osaamistarpeiden ennakointi on keskeistä koulutusorganisaatioille, joiden tehtävänä on vastata tulevaisuuden työelämän tarpeisiin. Työelämän ja teknologian jatkuva kehitys tekee ennakkoinnista kuitenkin haastavaa, sillä teknologinen murros vähentää tarvetta tarkasti määritellyille osaamistarpeille ja korostaa yleisten taitojen merkitystä. Kaikki osaaminen ei kuitenkaan muutu yleisluonteiseksi, vaan erityisesti tietointensiivisillä aloilla tarvitaan edelleen ammattialakohtaista, spesifiä osaamista. (Nyysölä 2022, 113–114.)

Yritykset keskittyvät tyypillisesti nykyisiin osaamistarpeisiinsa, kun taas julkinen sektori ja koulutusorganisaatiot harjoittavat perinteisesti laajempaa ennakointityötä, kohdistuen ensisijaisesti ammattiryhmiin ja toimialoihin yksittäisten taitojen sijaan (Balcar ym. 2011, 52). Vaikka osaamistarpeiden voidaan ajatella kumpuavan suoraan työelämän vaatimuksista (Hanhinen 2010, 121), eivät ne yksinään määritä koulutuksen kehityssuuntaa. Korkeakoulut voivat myös vaikuttaa työelämän kehitykseen tuottamalla uutta tietoa ja teknologiaa sekä luomalla uusia mahdollisuuksia yrityksille ja opiskelijoille. Esimerkiksi Helsingin yliopiston käynnistämä T&K Kiihdyttämö -palvelu tukee yrityksiä hyödyntämään tieteellistä tutkimusta liiketoiminnan kehittämisessä (Humalamäki 2025).

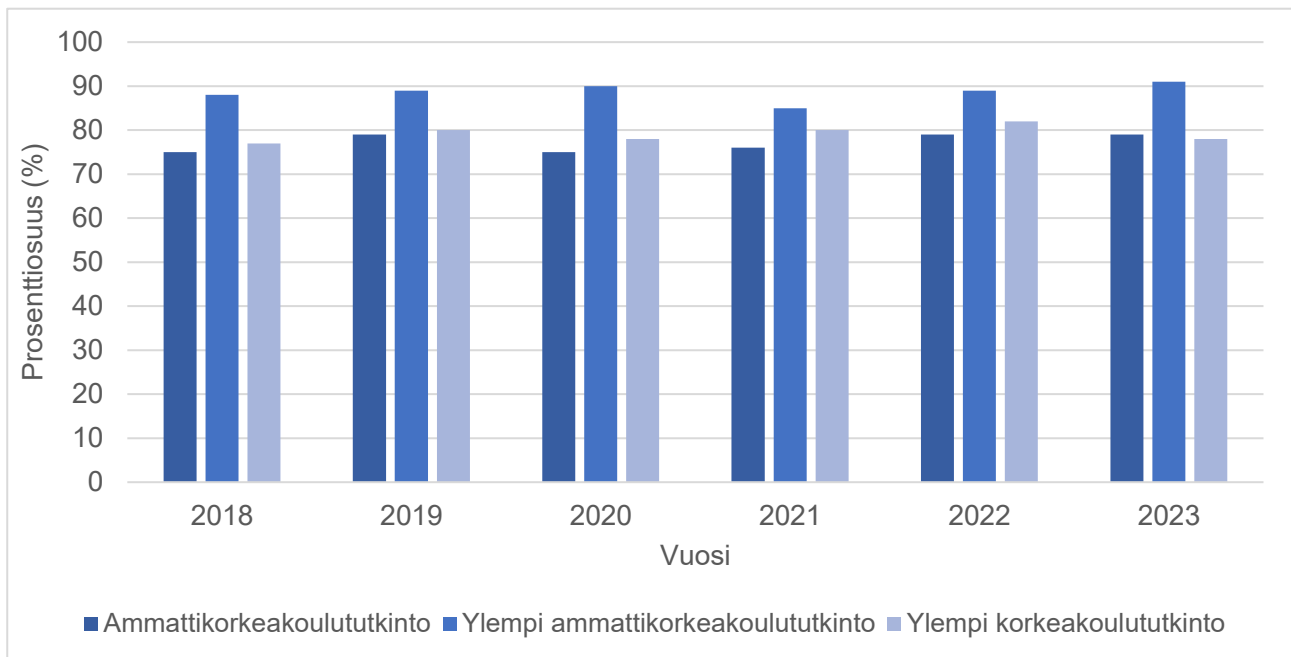
Koulutusorganisaatioiden ja työelämän tiivis yhteistyö onkin ratkaisevassa asemassa osaamistarpeiden ennakkoinnissa. Koulutus ei ainoastaan valmista opiskelijoita työmarkkinoille, vaan myös hyödyntää työelämän asiantuntemusta opetuksen ja tutkimuksen kehittämisessä. Tiivis yhteistyö mahdollistaa sekä koulutuksen että työelämän kehittymisen siten, että ne vastaavat entistä paremmin muuttuviin osaamistarpeisiin.

2.4 Alan suunta Suomessa

Työ- ja elinkeinoministeriön julkaisema raportti (Ek 2020) avaa myönteisiä näkymiä Suomen ohjelmistoalan kehitykseen. Raportin mukaan alan jatkuvan kasvun ja kehityksen vuoksi alan ammattilaisilta vaaditaan jatkuvaa teknisen osaamisen päivittämistä. Toimialan kasvua voi kuitenkin rajoittaa osaavan työvoiman saatavuus. (Ek 2020, 24–25.)

Tätä näkemystä näyttäisi tukevan myös Mäntysaaren (2022) pro gradu tutkielma, jossa tutkittiin IT-alan koulutustarjonnan ja työmarkkinoiden osaamistarpeiden kohtaantoa analysoimalla 517 IT-alan työpaikkailmoitusten osaamistarvetta ja 1139 korkeakoulujen opetussuunnitelmista tunnistettua osaamistavoitetta. Ohjelmistokehitys ja tietojärjestelmäprojektit muodostivat tutkimushetkellä IT-alan suurimman työtehtäväkategorian työpaikkailmoituksissa, mikä tutkimuksen mukaan antaa selkeitä viitteitä osaajapulasta (Mäntysaari 2022, 122).

Vuonna 2023 ICT-alan AMK- ja YAMK-tutkintoon sekä ylempään korkeakoulututkintoon valmistuneista päätoimisesti työllisiä vuoden päästä valmistumisestaan oli keskimäärin 81,7 %. Kaikkien alojen vastaava keskimääräinen työllisyysaste oli hieman korkeampi, noin 82,3 %. ICT-alan tutkinnon suorittaneiden työllisyysaste on pysynyt vuosina 2018–2023 tasaisen hyvänä, vaikka vuoden 2023 ICT-alan työllisyysasteen keskiarvo jääkin hieman alle kaikkien alojen keskiarvon. (Tilastokeskus 2024b.)



Kuva 2. ICT-alalta valmistuneiden työllistymistilanne vuoden kuluttua valmistumisesta (Tilastokeskus 2024b)

Myös Teknologiateollisuus ry:n tilauskantatiedustelun mukaan Suomen tietotekniikka-ala on kasvanut merkittävästi vuosina 2010–2025 (pl. pelialan ohjelmistoyritykset ja datakeskukset). Ala on suhdanneherkkä, ja lyhyen aikavälin vaihtelut voivat olla merkittäviä, minkä vuoksi tutkimuksemme kannalta olennaista on keskittyä pitkän aikavälin tarkasteluun. Vuosina 2010–2025 alan kasvu näyttäytyy vakaana, ja esimerkiksi uusien tilauksien keskimääräinen vuosikasvu on ollut noin 5,33 %. (Teknologiateollisuus ry 2025.)

2.5 Ohjelmistokehityksen kansainväliset viitekehykset

Oppilaitokset voivat hyödyntää kansainvälisiä viitekehyksiä ja oppaita opetussuunnitelmiansa kehittämässä varmistaakseen koulutuksen ajantasaisuuden ja sen vastaavuuden kehittyvän yhteiskunnan tarpeisiin. Ne voivat tarjota myös organisaatioille ja yksittäisille ammattilaisille työkaluja

osaamisen arviointiin, kehittämiseen ja urasuunnitteluun. Tällaiset viitekehykset voivat olla hyödyksi myös meidän tutkimuksessamme, tarjoten vertailupohjan, jonka avulla voimme suhteuttaa aineistostamme nousseet osaamiset kansainvälisesti tunnistettuihin osaamisvaatimuksiin.

Olemme valinneet tutkimuksemme tueksi kolme kansainvälistä raporttia, opasta ja suositusta, jotka tarjoavat viitekehykset vertailupohjaksi ohjelmistokehityksen osaamisvaatimusten tarkasteluun:

- Computing Curricula 2020 (CC2020) (Association for Computing Machinery & IEEE Computer Society 2020)
- Software Engineering Body of Knowledge Guide V4.0 (SWEBOK) (IEEE Computer Society 2024)
- European ICT Professional Role Profiles (Breyer ym. 2018)

CC2020 (Association for Computing Machinery & IEEE Computer Society 2020) esittelee kansainväliset suositukset tietojenkäsittelyn (computing) kandidaattitason koulutussisällöistä ja osaamisvaatimuksista. Raportti korostaa siirtymää tietopohjaisesta opetuksesta osaamisperustaiseen lähestymistapaan (competency-based approach). Raportissa tietojenkäsittely jaetaan useisiin eri alakategorioidiin, jotka esitellään taulukossa 2. On syytä huomata, että taulukon suomenkieliset käännökset ovat vapaasti muotoiltuja eivätkä välttämättä täysin vastaa ACM:n alkuperäisiä määritelmiä. Esimerkiksi suomenkielinen "tietojenkäsittelytiede" voi kattaa laajemman kokonaisuuden kuin ACM:n Computer Science, joka rajautuu teoreettisempaan tietojenkäsittelyyn.

Taulukko 2. Tietojenkäsittelyn kandidaattitason kategorioiden jaottelu ja niihin liittyvät viitekehykset (mukaillen Association for Computing Machinery & IEEE Computer Society 2020, 26–29)

Kategoria ja siihen liittyvä viitekehys	Kuvaus
Computer Engineering, CE2016	Yhdistää tietojenkäsittelyn ja sähkötekniikan keskittyen ohjelmisto- ja laitteistokomponenttien suunnitteluun, toteutukseen ja ylläpitoon.
Computer Science, CS2013	Keskittyy teoreettiseen tietojenkäsittelyyn, jossa painottuvat laskennan matemaattiset perusteet, abstrakti ajattelu, ohjelmointi ja algoritminen ongelmanratkaisu.
Cybersecurity, CSEC2017	Keskittyy tietojen, ohjelmistojen, laitteiden ja verkkojen suojaamiseen sekä kyberturvallisuuden riskienhallintaan.
Information Systems, IS2010	Yhdistää teknologian ja organisaation prosessit, painotuen tiedon hallintaan ja päätöksenteon tukemiseen.
Information Technology, IT2017	Korostaa käytännön teknologiaosaamista, käytettävyyttä, käyttäjälähtöisyyttä ja viestintätaitoja.
Software Engineering, SE2014	Painottaa luotettavien ohjelmistojen suunnittelua ja toteutusta insinöörimäisesti, keskittyen prosesseihin, skaalautuvuuteen ja laatuun.

Kategoria ja siihen liittyvä viitekehys	Kuvaus
Data Science, viitekehys kehitteillä	Yhdistää laskennalliset ja tilastolliset menetelmät suurten datamassojen analysointiin ja uuden tiedon tuottamiseen.

Tutkimuksemme näkökulmasta raportin keskeisimmät viitekehukset ovat Information Technology (IT2017) ja Software Engineering (SE2014). Niiden avulla voimme tarkastella, missä määrin aineistossa esiin nousseet osaamiset vastaavat kansainvälisiä suosituksia. IT2017-viitekehysten mukaan (Association for Computing Machinery & IEEE Computer Society 2017, 19–20) tietotekniikan kandidaatilta edellytetään seuraavien osaamisalueiden hallintaa:

1. Ongelmien analysointi: osaa analysoida monimutkaisia, tosielämän ongelmia, tunnistaa niihin liittyvät laskennalliset vaatimukset ja soveltaa sopivia menetelmiä niiden ratkaisemiseksi.
2. Ratkaisujen suunnittelu ja toteutus: osaa suunnitella, toteuttaa ja arvioida tietoteknisiä ratkaisuja määriteltyjen vaatimusten mukaisesti.
3. Viestintätaidot: osaa viestiä teknisestä sisällöstä tehokkaasti eri kohderyhmille.
4. Eettinen ja oikeudellinen harkinta: osaa tehdä perusteltuja päätöksiä ottaen huomioon eri osapuolten näkökulmat sekä eettiset ja oikeudelliset periaatteet.
5. Tiimityötaidot: osaa toimia tehokkaasti tiimeissä, edistää vuorovaikutusta ja tuottaa sovitut lopputulokset halliten tehtävien aikataulutusta ja riskienhallintaa.
6. Käyttäjatarpeiden tunnistaminen: osaa tunnistaa käyttäjien tarpeet ja huomioida ne järjestelmien valinnassa, integroinnissa ja hallinnoinnissa.

SE2014-viitekehys (IEEE Computer Society & Association for Computing Machinery 2015, 20–21) puolestaan määrittelee seitsemän keskeistä osaamisaluetta, joita ohjelmistotekniikan (software engineering) kandidaatilta odotetaan:

1. Ammattitaito: hallitsee ohjelmistotekniikan keskeiset tiedot ja ammatilliset standardit; tuntee etiikan, taloudelliset ja yhteiskunnalliset näkökulmat.
2. Tekninen osaaminen: osaa soveltaa ohjelmistokehityksen teorioita, malleja ja menetelmiä ongelmien tunnistamiseen, analysointiin, ohjelmiston suunnitteluun, kehittämiseen, toteuttamiseen, testaukseen ja dokumentointiin.
3. Tiimityöskentelytaidot: toimii sekä itsenäisesti että ryhmässä. Ymmärtää roolit, viestinnän, aikataulut ja ryhmädynamiikan merkityksen.
4. Loppukäyttäjän tarpeiden ymmärrys ja sidosryhmäviestintä: tunnistaa asiakkaiden ja käyttäjien tarpeet. Osallistuu projektityöskentelyyn ja harjoittaa tehokasta vuorovaikutusta sidosryhmien kanssa.

5. Ratkaisujen suunnittelu: suunnittelee ohjelmistoratkaisuja ottaen huomioon eettiset, sosiaaliset, oikeudelliset ja taloudelliset reunaehdot sovellusalueen mukaan.
6. Kyky kompromissien tekemiseen: tasapainottaa ristiriitaisia vaatimuksia resurssien, ajan, teknologisten rajoitteiden ja organisatoristen olosuhteiden puitteissa.
7. Elinikäinen oppiminen ja ammatillinen kehittyminen: osaa etsiä ja omaksua uutta tietoa sekä ymmärtää jatkuvan ammatillisen kasvun tärkeyden nopeasti muuttuvalla alalla.

SWEBOK V4.0 on IEEE Computer Society'n (2024) julkaisema kansainvälisen oppaan tuorein versio, jonka tarkoituksena on tarjota yhtenäinen ymmärrys ohjelmistokehittäjän osaamiselle. Oppaan sisältö perustuu alan asiantuntijoiden konsensukseen, ja sen kattavuus ja ajankohtaisuus tekevät siitä erityisen merkittävän viitekehyksen myös tämän tutkimuksen kannalta. (IEEE Computer Society 2024, xxv.) SWEBOK V4.0 kuvaa ohjelmistokehityksen ammatillista tietoperustaa, ja sitä voidaan hyödyntää esimerkiksi osaamisen arvioinnissa, urasuunnittelussa ja ammattistandardien määrittelyssä (IEEE Computer Society 2024, xxvii).

Opas jäsentää 18 osaamisaluetta, jotka kattavat ohjelmistokehityksen ydinalueet vaatimustenhallinnasta testaukseen, laadunhallintaan ja ammattikäytäntöihin. Sen mukaan ohjelmistokehittäjän ei tarvitse hallita kaikkia näitä osa-alueita perustavanlaatuisesti, mutta tulisi tuntea ne ainakin yleisellä tasolla ja soveltaa niistä omaan työkuvaansa olennaisia sisältöjä. (IEEE Computer Society 2024, xxxix.) Kyseiset osaamisalueet sekä niiden sisältämät aiheet esitellään taulukossa 3.

Taulukko 3. SWEBOK V4.0 mukaiset osaamisalueet (mukaiillen IEEE Computer Society 2024)

Osaamisalue	Keskeiset aiheet
Ohjelmistovaatimukset	Vaatimusten kerääminen, analysointi, määrittely, validointi ja hallinta (IEEE Computer Society 2024, Chapter 01: Software Requirements).
Ohjelmistoarkkitehtuuri	Arkkitehtuurityylit, näkymät, arviointimenetelmät ja kuvauskielet (IEEE Computer Society 2024, Chapter 02: Software Architecture).
Ohjelmistosuunnittelu	Suunnittelun peruskäsitteet, menetelmät, suunnittelumallit ja laadun arviointi (IEEE Computer Society 2024, Chapter 03: Software Design).
Ohjelmiston rakennus	Ohjelmointikäytännöt, rakennustyökalut, standardit ja virheenjäljitys (IEEE Computer Society 2024, Chapter 04: Software Construction).
Ohjelmistotestaus	Testaustasot, testausmenetelmät, mittarit ja testiautomaatio (IEEE Computer Society 2024, Chapter 05: Software Testing).
Ohjelmistokehityksen operatiivinen toiminta	DevOps-käytännöt, julkaisunhallinta, käyttöönotto ja seuranta (IEEE Computer Society 2024, Chapter 06: Software Engineering Operations).
Ohjelmistojen ylläpito	Ylläpitotyypit, vaikutusanalyysi, ohjelmakoodin ymmärtäminen ja ylläpitotyökalut (IEEE Computer Society 2024, Chapter 07: Software Maintenance).

Osaamisalue	Keskeiset aiheet
Konfiguraationhallinta	Versionhallinta, muutosten hallinta ja julkaisun hallinta (IEEE Computer Society 2024, Chapter 08: Software Configuration Management).
Ohjelmistoprojektien hallinta	Projektisuunnittelu, riskienhallinta, mittaaminen ja henkilöstöhallinta (IEEE Computer Society 2024, Chapter 09: Software Engineering Management).
Ohjelmistoprosessien hallinta	Prosessimallit, toteutus, arviointi ja prosessien kehittäminen (IEEE Computer Society 2024, Chapter 10: Software Engineering Process).
Mallit ja menetelmät	Formaalit menetelmät, mallinnus, ketterät menetelmät ja menetelmien yhdistely (IEEE Computer Society 2024, Chapter 11: Software Engineering Models and Methods).
Laatu	Laadun peruskäsitteet, laadunvarmistus, laatustandardit ja mittarit (IEEE Computer Society 2024, Chapter 12: Software Quality).
Tietoturva	Tietoturvaperiaatteet, uhkamallinnus, tietoturvatilastus ja turvallinen ohjelmointi (IEEE Computer Society 2024, Chapter 13: Software Security).
Ammattikäytännöt	Ammattietiikka, viestintätaidot, tiimityöskentely ja elinikäinen oppiminen (IEEE Computer Society 2024, Chapter 14: Software Engineering Professional Practice).
Taloudelliset näkökulmat	Kustannusarviointi, tuottoanalyysi, taloudelliset vertailut ja kompromissien analyysi (IEEE Computer Society 2024, Chapter 15: Software Engineering Economics).
Tietojenkäsittelyn perusteet	Ohjelmointikielet, tietorakenteet, algoritmit ja käyttöjärjestelmät (IEEE Computer Society 2024, Chapter 16: Computing Foundations).
Matemaattiset perusteet	Logiikka, diskreetti matematiikka, todennäköisyyslaskenta ja tilastolliset menetelmät (IEEE Computer Society 2024, Chapter 17: Mathematical Foundations).
Insinööritieteiden perusteet	Järjestelmätekniikka, laitteistojen perusteet, mittaaminen, mallintaminen ja simulointi (IEEE Computer Society 2024, Chapter 18: Engineering Foundations).

Edellä mainittujen ACM:n ja IEEE:n viitekehyksien lisäksi eräs tutkimuksemme kannalta merkittävä viitekehys on European ICT Professional Role Profiles (Breyer ym. 2018). Kyseessä on Euroopan standardointikomitean (CEN, European Committee for Standardization) asiantuntijatyöryhmän yhteistyössä laatima suositus, jonka tarkoituksena on edistää yhteistä ymmärrystä ICT-ammattilaisten tehtävistä ja osaamisvaatimuksista sekä tukea alan koulutuksen ja osaamisen kehittämistä eri Euroopan maissa. Suositus esittelee 30 ICT-ammattilaisen roolia, joiden kuvaukset sisältävät rooliyhteenvedon, tehtäväkuvauksen, keskeiset työtehtävät, European e-Competence Frameworkin

(e-CF) perustuvat osaamisvaatimukset, tuotokset sekä suorituskykymittarit. Rooleja ei ole tarkoitettu tulkittavaksi tiukkoina standardeina, vaan toimivat käytännönläheisenä työkaluina, joiden avulla voidaan yhdistää osaamistarpeet konkreettisiin työtehtäviin. (Breyer ym. 2018, 5.)

Ohjelmistokehittäjän rooli (developer) määritellään viitekehyksessä rooliksi, jossa henkilön pääasiainen tehtävä on suunnitella ja toteuttaa ohjelmistokomponentteja vaatimusmäärittelyjen mukaisesti. Roolissa edellytettävät osaamiset määritellään e-CF:n mukaisesti viidellä osaamisalueella: sovelluskehitys, komponenttien integrointi, testaus, dokumentointi ja ongelmanhallinta. (Breyer ym. 2018, 14.) Roolin kuvaus esitellään tarkemmin taulukossa 4.

Taulukko 4. Ohjelmistokehittäjän roolin kuvaus (mukaillen Breyer ym. 2018, 14)

Osa-alue	Sisältö
Roolin nimi	Ohjelmistokehittäjä
Yhteenveto	Suunnittelee ja/tai ohjelmoi komponentteja vaatimusmäärittelyjen mukaisesti.
Tehtäväkuvaus	<ul style="list-style-type: none"> - Vastaa sovellusten toteuttamisesta ja käyttöönotosta. - Osallistuu matalan tason suunnitteluun. - Ohjelmoi sovelluksia optimoiden tehokkuutta, toiminnallisuutta ja käyttäjäkokemusta.
Pääasialliset työtehtävät	<ul style="list-style-type: none"> - Ohjelmistokomponenttien suunnittelu, toteutus ja integrointi. - Käyttäjäkokemuksen huomioiminen. - Tietoturvan huomioiminen. - Dokumentoinnin tuottaminen. - Tekninen tuki. - Ongelmien ratkaisu ennen testausta ja sen jälkeen.
Osaamiset (e-CF:n mukaisesti)	<ul style="list-style-type: none"> - B.1. Sovelluskehitys. - B.2. Komponenttien integrointi. - B.3. Testaus. - B.5. Dokumentaation tuottaminen. - C.4. Ongelmanhallinta.
Tuotokset	<ul style="list-style-type: none"> - Dokumentoitu koodi. - Ohjelmistokomponentit. - Ohjelmiston suunnittelukuvaus. - Testausproseduurit. - Käyttäjäkokemussuunnitelmat.
Suorituskykyindikaattorit	Toimivat komponentit.

2.6 Aiempi aiheeseen liittyvä tutkimus

Valitsimme tutkimuksemme teoriaosaan kaksi tieteellistä artikkelia, jotka käsittelevät koulutuksen ja työelämän osaamistarpeiden välistä suhdetta, ja tuovat edellä mainittujen raporttien ja viitekehys-

ten lisäksi tutkimuksellista perustaa ohjelmistokehityksen osaamistarpeiden tarkastelullemme. Ensimmäinen artikkeli, Timothy C. Lethbridgen *The Relevance of Education to Software Practitioners: Data from the 1998 Survey* (1999), pohjautuu 181 ohjelmistoalan ammattilaiselta vuonna 1998 kerättyyn verkkokyselyaineistoon (Lethbridge 1999, 4). Tutkimuksessa vastaajilta kysyttiin 75 aihealueeseen liittyen, kuinka paljon he olivat oppineet aiheesta koulutuksessa, mikä oli heidän osaamistasonsa, kuinka hyödylliseksi he kokivat aiheen työssä, ja miten aihe oli vaikuttanut heidän ajatteluunsa (Lethbridge 1999, 5). Nämä 75 aihealuetta ryhmiteltiin 13 kategoriaan (mukailen Lethbridge 1999, 6):

1. Ohjelmistokehitysprosessi (Software Engineering Process)
2. Ohjelmistosuunnittelun ydin (Software Design Core)
3. Ohjelmistojen alijärjestelmien suunnittelu (Software Subsystem Design)
4. Muu ohjelmistoihin liittyvä (Other Software)
5. Tietotekniikan ohjelmistoihin liittyvät aiheet (Computer Engineering Software Topics)
6. Tietojenkäsittelytieteen teoreettiset aiheet (Computer Science Theoretical Topics)
7. Tietojenkäsittelyssä laajasti käytetyt matemaattiset aiheet (Mathematical Topics Widely Used in Computer Science)
8. Muu matematiikka (Other Mathematics)
9. Tietotekniikan laitteistoon liittyvät aiheet (Computer Engineering Hardware Topics)
10. Muu laitteistoihin liittyvä (Other Hardware)
11. Perustieteet (Basic Science)
12. Liiketoiminta (Business)
13. Humanistiset aiheet ja taidot (Humanities & Skills)

Toinen artikkeli, Kirsti Ala-Mutkan ja Antti Puhakan *Survey on the Knowledge and Education Needs of Software Professionals* (2009), syventää Lethbridgen tutkimusta suomalaisesta näkökulmasta. Myös tämä tutkimus toteutettiin verkkokyselynä, ja siihen vastasi 212 ohjelmistoalan ammattilaista. Kysymykset käsittelivät seitsemää laajaa osaamisaluetta, jotka jakoutuivat 72 tarkempaan teemaan. (Ala-Mutka & Puhakka 2009, 2.) Kysymyksenasettelu pohjautui Lethbridgen tutkimukseen (Ala-Mutka & Puhakka 2009, 1), ja vastaajat arvioivat omaa oppimistaan ja osaamisensa hyödyllisyyttä työelämässä (Ala-Mutka & Puhakka 2009, 2).

Ala-Mutkan ja Puhakan tutkimuksessa esiintyvät seitsemän laajaa osaamisaluetta olivat (mukailen Ala-Mutka & Puhakka 2009, 2):

1. Tietojenkäsittelytieteen ydin (Computer Science Core)
2. Teoreettinen tietojenkäsittelytiede (Theoretical Computer Science)
3. Ohjelmistokehitys (Software Engineering)
4. Ohjelmistojen sovellutukset (Applications of Software)

5. Muu informaatioteknologia (Other Information Technology)
6. Matematiikka (Mathematics)
7. Muut aiheet (Other Topics).

Vastausvaihtoehdot pohjautuivat aihealueisiin, jotka oli valittu yliopistojen opetussuunnitelmien ja aikaisempaan vastaavaan tutkimukseen osallistuneiden henkilöiden ehdotusten perusteella (Lethbridge 1999, 4). Lethbridgen tutkimuksessa aiheita oli 75 ja Ala-Mutkan ja Puhakan tutkimuksessa 72, mikä tekee vastausvaihtoehtojen kattavuudesta varsin laajan. Tutkimustamme ajatellen meitä kiinnosti eniten se, mitkä taidot näyttäytyivät näissä tutkimuksissa työelämän kannalta tärkeimpinä.

Tulosten mukaan työelämässä hyödyllisimmiksi tai tärkeimmiksi koettuja taitoja olivat muun muassa ohjelmointitaidot, tietorakenteet, tietokannat, vaatimusmäärittely, dokumentointi ja testaus (Lethbridge 1999, 33; Ala-Mutka & Puhakka 2009, 4). Vastaajat kokivat oppineensa useita taitoja koulutuksessa liian vähän (Ala-Mutka & Puhakka 2009, 12). Ala-Mutka ja Puhakka toteavat, että heidän havaintonsa ovat yhteneväisiä aikaisemman Lethbridgen tutkimuksen kanssa, mutta suomalainen koulutus painotti enemmän käytännön osaamista kuin koulutus Lethbridgen tutkimuksen aikaan Pohjois-Amerikassa (Ala-Mutka & Puhakka 2009, 24).

Nämä kaksi tutkimusartikkelia antavat vertailupohjaa aineistostamme löytyvien aiheiden tarkastelulle. Voimme verrata aineistosta esiin nousevia taitoja ja osaamistarpeita aiemmissä tutkimuksissa tehtyihin luokitteluihin. On mielenkiintoista myös verrata aikaisempia tutkimuksia omaamme, joka on myöhemmältä ajalta, ja tarkastella, näkyykö tuloksissa viitteitä siitä, että osaamistarpeet olisivat ajan myötä muuttuneet olennaisesti.

2.7 Aiemmat opinnäytetyöt ja kyselyt aiheesta

Perehdyimme myös aiempiin opinnäytetöihin ja kyselyihin, joissa on tutkittu ohjelmistokehityksen osaamistarpeita. Näistä keskeisiä ovat Joonas Teerimäen (2012) ja Jenny Kallioston (2018) opinnäytetyöt sekä Oulun ammattikorkeakoulun vuonna 2024 toteuttama sidosryhmäkysely.

Teerimäki keskittyi tutkimuksessaan selvittämään ICT-alan osaamistarpeita ja sitä, miten hyvin Haaga-Helia ammattikorkeakoulun IT-tradenomikoulutus antaa näihin tarpeisiin vastaavia valmiuksia (Teerimäki 2012, 2). Kallioston tavoitteena oli kartoittaa vastavalmistuneiden ohjelmistokehittäjien osaamistarpeita (Kalliosto 2018, 12). Molemmat tutkimukset toteutettiin kyselytutkimuksina, ja analyseissa hyödynnettiin sekä kvalitatiivisia että kvantitatiivisia menetelmiä. Sekä Teerimäki että Kalliosto tarkastelivat erityisesti sitä, mitkä taidot olivat olleet vastaajien mielestä tärkeimpiä.

Tuoreempaa näkökulmaa tuo Oulun ammattikorkeakoulussa toteutettu sidosryhmäkysely, jossa selvitettiin, millaista osaamista ICT-alan tutkinnon suorittaneilta vastavalmistuneilta odotetaan (Niva 2024). Kyselyn tavoitteena oli tuottaa tietoa opetussuunnitelmien kehittämiseksi siten, että ne vastaisivat paremmin työelämän tarpeita. Taulukkoon 5 on koottu yhteenveto tutkimuksista ja niiden keskeisistä tuloksista. Lisäksi viimeinen sarake havainnollistaa, miten nämä tutkimukset vertautuvat tutkimusasetelman ja näkökulman tasolla omaan tutkimukseemme.

Oma tutkimuksemme poikkeaa edellä mainituista siten, että tarkastelemme opinnäytetöiden muodostamaan aineistoon perehtyen, mitä taitoja ja teknologioita työtehtävissä on konkreettisesti käytetty. Siinä missä Teerimäki arvioi koulutuksen vastaavuutta ICT-alan osaamistarpeisiin, Kalliosto selvitti vastavalmistuneiden käsityksiä tärkeimmistä taidoista ja Niva kartoitti työnantajien odotuksia valmistuneiden osaamisesta, meidän tutkimuksemme kohdistuu havaittuun, ei arvioituun tai odotettuun osaamiseen. Vaikka tutkimusasetelmat eroavat toisistaan, on kiinnostavaa verrata Teerimäen, Kallioston ja Nivan tärkeimpinä pidettyjä aiheita niihin, jotka aineistomme perusteella näyttävät yleisimpinä. Erityisen mielenkiintoista on selvittää, millaisia mahdollisia yhteneväisyyksiä tai eroja tuloksissa ilmenee.

Taulukko 5. Aiempien opinnäytetöiden ja kyselyiden esittely (Teerimäki 2012, 21–28; Kalliosto 2018, 12–24; Niva 2024)

Julkaisu	Tutkimusmenetelmä ja aineisto	Keskeiset tulokset
Joonas Teerimäki (2012): IT-tradenomikoulutuksen vastaavuus työelämän vaatimuksiin. Opinnäytetyö.	Verkkokysely (107 Haaga-Heliasta valmistunutta vastaajaa); yhdistetty kvalitatiivisia ja kvantitatiivisia menetelmiä	Tärkeimmät teknologiat: ohjelmointi, tietokannat, SQL, Java, PHP, HTML. Muut tärkeimmät taidot: testaus, projektinhallinta, dokumentointi, raportointi, presentaatio-aidot, ajanhallinta, organisointitaidot
Jenny Kalliosto (2018): Ohjelmistokehittäjän työelämän osaamistarpeet. Opinnäytetyö.	Verkkokysely (58 ohjelmistokehityksen parissa työskentelevää vastaajaa); mixed methods	Tärkeimmät teknologiat: Java, JavaScript, C#, React, Docker, tietokannat, pilvipalvelut. Muut tärkeimmät taidot: käyttäjäystävällinen suunnittelu, ketterät menetelmät projektinhallinnassa, testaus, dokumentointi

Julkaisu	Tutkimusmenetelmä ja aineisto	Keskeiset tulokset
<p>Anu Niva (2024): ICT-alan tutkinto-ohjelmasta valmistuneelta odotetaan erityisesti ohjelmointi- ja kommunikaatiotaitoja. Artikkel, Oulun ammattikorkeakoulu.</p>	<p>Webropol-kysely si-dosryhmille (35 vastaajaa); yhdistetty kvalitatiivisia ja kvantitatiivisia menetelmiä.</p>	<p>Tärkeimmät teknologiat: ohjelmointi (Python, JavaScript, Java) ja ohjelmointityökalut, versiohallinta, CI, pilvipalvelut, testaus, tietokannat, tietokone ja työkaluohjelmistot.</p> <p>Muut tärkeimmät taidot: viestintä, tiimityöskentely- ja yhteistyötaidot, ketteruus, asiakasläh-töisyys, projektinhallinta, muutoskyky, sosiaali-set taidot, itseohjautuvuus, DevOps.</p>

3 Empiirinen osa

3.1 Aineisto

3.1.1 Päiväkirjamuotoiset opinnäytetyöt tutkimusaineistona

Tutkimuksemme aineisto muodostuu Haaga-Helia ammattikorkeakoulun päiväkirjamuotoisista opinnäytetöistä. Haaga-Helia on pääkaupunkiseudulla toimiva ammattikorkeakoulu, josta on Suomen ammattikorkeakoulujen joukosta valmistunut useana peräkkäisenä vuonna eniten tietojenkäsittelyn tutkinnon opiskelijoita (Opetushallinnon tilastopalvelu Vipunen 2025).

Opiskelija tekee päiväkirjamuotoista opinnäytetyötä varten työtehtävistään päivittäisiä merkintöjä, joita hän palaa reflektoidaan viikoittaisissa viikkoanalyysissä hyödyntäen ammatillisia ja tieteellisiä lähteitä. Tavoitteena on analysoida työtehtäviä ja tarkastella ammatillista kehittymistä suhteessa ennalta valittuihin teemoihin ja opiskelijan itselleen asettamiin tavoitteisiin. (Opinnäytetyökoordinaattorit 2022, 6.) Prosessi kestää vähintään kahdeksan viikkoa, ja se sisältää 40 työpäivää (Opinnäytetyökoordinaattorit 2022, 1).

Opinnäytetyössä noudatettava rakenne (Opinnäytetyökoordinaattorit 2022, 4–7) koostuu seuraavista osista:

1. Johdanto
2. Lähtötilanteen kuvaus
3. Oman nykyisen työn analyysi
4. Sidosryhmät
5. Vuorovaikutustilanteet
6. Päiväkirjaraportointi.

Päiväkirjamuotoiset opinnäytetyöt tarjoavat ainutlaatuisen ja syvällisen aineiston työelämän osaamistarpeiden tutkimukseen. Toisin kuin haastatteluissa tai kyselyissä, joissa vastaajat tuovat esiin tutkimushetkellä mieleen tulevia näkemyksiään, päiväkirjamuotoisessa aineistossa työtehtäviä ja osaamista raportoidaan ja reflektoidaan säännöllisesti ja pitkäkestoisesti. Päivittäisiin työtehtäviin palataan vielä viikkoanalyysissä kirjoittaessa, mikä syventää reflektiota entisestään. Opinnäytetyöt soveltuvat erinomaisesti työelämän osaamistarpeiden tutkimiseen, koska niistä saadaan konkreettista tietoa tutkittavalla ajanjaksolla käytössä olleista teknologioista ja ohjelmistokehityksen keskeisistä työtehtävistä.

Kuten edellä on kuvattu, opinnäytetyöt noudattavat yhtenäistä rakennetta. Teksteissä on kuitenkin vahvoja yksilöllisiä painoituksia: kirjoittajat valitsevat itse tarkasteltavat teemat sekä sen, mitä työtehtäviinsä liittyviä asioita he nostavat esille raportointijaksolta. Vaikka päivittäiset työtehtävät ja

käytettävät teknologiat raportoidaan yksityiskohtaisesti, aineistossa korostuvat ne asiat, mitä kirjoittajat ovat itse pitäneet tärkeinä ja mainitsemisen arvoisina. Näiden asioiden voi olettaa liittyvän juuri alan osaamistarpeisiin ja tästä voidaan perustellusti tehdä johtopäätös, että aineisto heijastaa aidosti työelämässä tarvittavaa osaamista päivittäisten tehtävien konkreettisella tasolla.

3.1.2 Aineiston rajaus

Tuomen ja Sarajärven (2018, 87–88) mukaan laadullisessa tutkimuksessa aineisto valitaan sen mukaan, mistä uskotaan saatavan parhaiten vastauksia tutkimustehtävään. Tämä näkökulma ohjasi myös omaa valintaprosessiamme. Halusimme valita aineistoksi kaikkein tuoreimmat opinnäytetyöt saadaksemme mahdollisimman ajantasaista tietoa nopeasti kehittyvän ohjelmistoalan osaamistarpeista. Aineiston valinnan kriteereitä on kuvannut myös Salminen (2011, 16–17), ja hänen esittämänsä kirjallisuuskatsauksen aineistonvalintaa kuvaava prosessi sopii myös meidän tutkimukseemme. Salminen kuvaa, kuinka aineistomassaa käydään asetettujen kriteereiden perusteella läpi, ja lopulta mukaan otetaan lopullisen seulan läpäisseet julkaisut (Salminen 2011, 16–17).

Aineistonmuodostuksen ensimmäisessä vaiheessa haimme kaikki vuosina 2020–2023 julkaistut tietojenkäsittelyn päiväkirjamuotoiset opinnäytetyöt Haaga-Heliasta (n = 420). Tämän jälkeen rajasimme pois työt, jotka eivät liittyneet ohjelmistokehitykseen vaan joissa työtehtävinä olivat muut IT-alan tehtävät, kuten lähituki tai esimerkiksi tietoverkkoihin liittyvät tehtävät. Otimme mukaan lopulta vain ne opinnäytetyöt, joissa kirjoittaja työskenteli ohjelmistokehittäjän roolissa tai vastaavissa tehtävissä. Näin muodostui alustava 124 opinnäytetyön aineisto.

Vuonna 2025 päätimme laajentaa aineistoa sisällyttämällä siihen myös vuoden 2024 aikana julkaistut opinnäytetyöt, jotka kävimme läpi samoin sisäänotto- ja poissulkukriteerein kuin aiemmat. Näin saimme lopullisen aineiston, joka käsittää yhteensä 154 opinnäytetyötä. Aineiston valintakriteerit on koottu taulukkoon 6, ja valitut opinnäytetyöt on eritelty julkaisuvuosittain taulukossa 7.

Taulukko 6. Lopulliset sisäänotto- ja poissulkukriteerit

	Sisäänottokriteeri	Poissulkukriteeri
Julkaisuajankohta	Opinnäytetyö on julkaistu ajanjaksolla 1.1.2020-31.12.2024	Muulla ajanjaksolla julkaistut opinnäytetyöt
Ammattikorkeakoulu	Haaga-Helia ammattikorkeakoulu	Muu ammattikorkeakoulu
Koulutusala	Tietojenkäsittelyn tradenomi	Muu koulutusala
Saatavuus	Luettavissa avoimessa tai Haaga-Helian verkossa	Ei luettavissa avoimessa eikä Haaga-Helian verkossa
Opinnäytetyön tyyppi	Päiväkirjamuotoinen opinnäytetyö	Muun tyyppinen kuin päiväkirjamuotoinen työ

	Sisäänottokriteeri	Poissulkukriteeri
Työtehtävän sijoittuminen IT-alalle	Ohjelmistokehityksen alue	Muut IT-alan alueet
Kirjoittajan spesifi työtehtävä/rooli	Ohjelmistokehittäjä tai vastaavat tehtävät muulla nimikkeellä	Muu työtehtävä

Taulukko 7. Lopullinen tutkimukseen valittu aineisto (*"Käyttörajattu" sisältää opinnäytetyöt, joiden julkaisua on rajattu, ja jotka eivät ole vapaasti luettavissa suoraan Theseuksesta)

	2020	2021	2022	2023	2024	Yhteensä
Opinnäytetöitä yhteensä (kpl)	26	24	40	34	30	154
Julkaisukieli						
Suomi	22	19	29	26	18	114
Englanti	4	5	11	8	12	40
Julkaisutapa						
Avoin	8	9	22	13	14	66
Käyttörajattu (*)	18	15	18	21	16	88

Opinnäytetyön kirjoittajan yleisin aineistossa esiintyvä työnimike oli kehittäjä tai ohjelmistokehittäjä (79 kpl). Seuraavaksi yleisimmät nimikkeet olivat front end -kehittäjä (13 kpl), full stack -kehittäjä (13 kpl) ja web- tai verkkokehittäjä (11 kpl). Aineistossa esiintyi myös muita työnimikkeitä, kuten IT-konsultti.

3.2 Tutkimusmenetelmien esittely

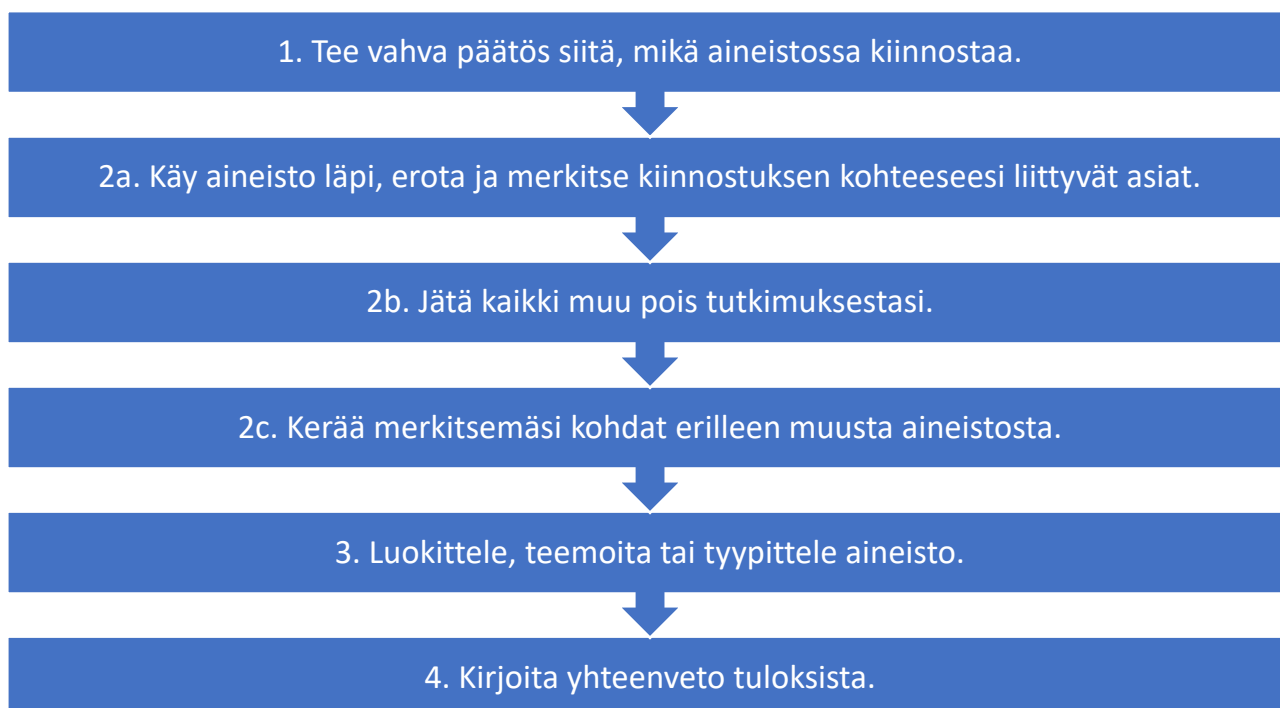
Sisällönanalyysi on laadullisen tutkimuksen menetelmä, jolla voidaan tutkia erilaisia kirjallisessa muodossa olevia tai kirjalliseen muotoon vietyjä aineistoja järjestelmällisesti (Tuomi & Sarajärvi 2018, luku 4.4). Sisällönanalyysi voidaan jaotella induktiiviseen, jossa analyysi etenee yksittäisistä havainnoista kohti yleisiä päätelmiä, ja deduktiiviseen, jossa edetään yleisestä kohti yksittäistä (Tuomi & Sarajärvi 2018, luku 4.2).

Valitsimme tutkimusmenetelmäksi induktiivisen sisällönanalyysin, koska halusimme tarkastella, millaisia osaamistarpeita ohjelmistokehityksen alalla on, ja meitä kiinnosti erityisesti, millaisia ilmiöitä aineistosta löytyy, kun lähestymme aihetta ilman valmiita oletuksia. Induktiivista analyysia tehdessä ei ole etukäteen tiedossa, millaisia luokkia aineistosta löytyy, koska ne muodostetaan analyysiä tehdessä (Tuomi & Sarajärvi 2018, luku 4.4.3). Induktiivisessä analyysissä tavoitteena on

löytää aineistosta se, mikä on tärkeää, määrittelemättä sitä hypoteesien pohjalta etukäteen (Hirsjärvi, Remes & Sajavaara 2016, 164).

Sisällönanalyysia voidaan luokitella myös sen mukaan, onko se teoria- vai aineistolähtöistä. Teorialähtöisessä analyysissä voidaan käyttää valmiiksi hahmotettuja kategorioita, joihin havaintoja peilataan. Aineistolähtöisessä analyysissä pyritään lähestymään aineistoa ilman ennakkokäsityksiä. On kuitenkin tärkeää tiedostaa, että esimerkiksi tutkijan taustaoletukset ja valittu tutkimusasetelma vaikuttavat väistämättä lopputulokseen silloinkin, kun tutkimusote on induktiivinen ja aineistolähtöinen. (Tuomi & Sarajärvi 2018, luku 4.2.)

Kuvassa 3 esitämme Tuomen ja Sarajärven kuvaaman Timo Laineen mallin sisällönanalyysin vaiheista. Etenimme omassa tutkimuksessa tämän mallin mukaisesti, ja aloitimme määrittelemällä analyysin kohteen eli päätimme, mitä halusimme tutkia. Sen jälkeen mallin mukaan toteutetaan aineiston systemaattinen läpikäynti, joka meidän tutkimukseemme oli osaamistarpeisiin liittyvien ilmausten tunnistaminen teksteistä. Aineistoa läpikäydessä merkitään eli koodataan tutkittavaan asiaan liittyvät ilmaisut, ja kun nämä on tunnistettu, aineistoa analysoidaan edelleen esimerkiksi luokittelemalla tai teemoittelemalla. Kaikki tutkimukseen kuulumaton rajataan prosessissa tutkimuksen ulkopuolelle. Kun tutkimukseen kuuluvat asiat on koottu yhteen ja analyysi on suoritettu, kirjoitetaan yhteenveto tai raportti. (Tuomi & Sarajärvi 2018, luku 4.1.)



Kuva 3. Timo Laineen kuvaus sisällönanalyysin kulusta Tuomen & Sarajärven mukaan (mukaillen Tuomi & Sarajärvi 2018, luku 4.1)

Tutkimuksemme linkittyy menetelmien osalta myös mixed methods research -lähestymistapaan (MMR), jossa yhdistetään laadullinen ja määrällinen tutkimus (Tuomi & Sarajärvi 2018, luku 2.5). Analysoimme aineistoa laadulliseen tutkimusperinteeseen kuuluvalla induktiivisen sisällönanalyysin menetelmällä, mutta määrällistä lähestymistapaa tutkimukseemme tuo sisällön erittely lukumääriä laskemalla ja vertailemalla. Näin pystymme arvioimaan esimerkiksi, mitkä aiheet esiintyvät aineistossa muita yleisemmin.

Tutkimusmenetelmien yhdistäminen liittyy läheisesti myös triangulaation käsitteeseen. Triangulaatiolla tarkoitetaan erilaisten menetelmien, aineistojen tai teorioiden yhdistämistä tai useamman tutkijan yhteistyötä samassa tutkimuksessa (Tuomi & Sarajärvi 2018, luku 6.4). Tutkimusaineistomme koostuu vain yhdentyypisestä dokumenttiaineistosta eli päiväkirjamuotoisista opinnäytetöistä, mutta tutkimuksessamme on muilta osin triangulaatioon sopivia elementtejä. Menetelmällisen triangulaation näkökulmasta tutkimuksessamme yhdistyy laadullinen sisällönanalyysi ja määrällinen tarkastelu, sillä poimittuja havaintoja sekä jaoteltiin laadullisesti että laskettiin lukumäärällisesti. Lisäksi kahden kirjoittajan tekemät rinnakkaiset havainnot ja niiden vertailu analyysin edetessä tuovat tutkijatriangulaation ulottuvuuden tutkimusprosessiimme.

3.3 Tutkimuksen kulku

Kun olimme rajanneet tutkimuskysymyksen ja aineiston, opinnäytetöiden läpikäynti vaikutti melko suoraviivaiselta. Halusimme tutkia ohjelmistokehityksen osaamistarpeita poimimalla teksteistä kirjoittajien työssään käyttämiä teknologioita, työkaluja ja taitoja. Ennen työhön ryhtymistä halusimme kuitenkin pysähtyä miettimään, millä perusteilla ja periaatteilla poimimme merkityksiä aineistossa esiintyvistä ilmauksista. Tutkijan on tärkeää kertoa riittävän yksityiskohtaisesti, miten tutkimus on toteutettu, jotta lukija voi muodostaa käsityksen tutkimuksen luotettavuudesta (Tuomi & Sarajärvi 2018, luku 6.3). Pyrimme seuraavaksi selittämään tutkimuksemme kulun läpinäkyvästi ja johdonmukaisesti, jotta se olisi sekä toistettavissa että uskottava.

Merkitysten löytämisestä kirjoittajan tai puhujan alkuperäisistä ilmauksista on kirjoittanut esimerkiksi Timo Laine (2018). Tutkittaessa yksilön omia ilmaisuja hänen kertoessaan jostain asiasta, tutkijan tehtävänä on yrittää tehdä ilmaisun pohjalta mahdollisimman tarkka tulkinta siitä, mitä ihminen ilmaisullaan tarkoittaa. Arkielämässä ihmiset toimivat hermeneuttisen esiyymmärryksen pohjalta, ikään kuin toisen ihmisen tuottamat merkitykset voisi ymmärtää sellaisinaan suoraan. Tieteellistä tutkimusta tehdessään tutkija tiedostaa oman esiyymmärryksensä, joka vaikuttaa tulkintaan. (Laine 2018, luku Hermeneuttinen ulottuvuus). Tutkija valitsee tutkimusmenetelmän sen mukaan, mikä on tutkimusasetelmaan nähden sopivin ja millä hän arvelee pystyvänsä löytämään ne merkitykset, joita puhuja tai kirjoittaja on alun perin tarkoittanut (Laine 2018, luku Metodisia ratkaisuja).

Analyysimme keskittyi poimimaan suoria mainintoja eri teknologioista ja taidoista, joten erilaisille tulkinnoille ei ollut yhtä paljon sijaa kuin toisenlaisessa tutkimuksessa olisi ollut. Yhtenäisyyden ja luotettavuuden varmistamiseksi sovimme kuitenkin selkeät kriteerit sille, miten poimimme ja tulkitimme osaamistarpeita teksteistä. Teknologioiden poimiminen aineistosta ja luokittelu kategorioihin oli yksinkertaista ja melko yksiselitteistä. Tietyn teknologian käytön maininta aineistossa kertoo siitä, että kyseiseen teknologiaan liittyvä osaaminen on ollut tarpeen. Jos kirjoittaja mainitsee tehneensä sovellukseen uuden ominaisuuden Java-koodilla Eclipse IDE:llä ja käyttäneensä SQL-kyselyjä tehdäkseen hakuja MySQL:stä, käytetyt teknologiat on helppo poimia ja luokitella. Tällöin voidaan myös yleisemmällä tasolla todeta, että kyseinen henkilö tarvitsee työssään ohjelmointiin tai tietokantoihin liittyvää osaamista.

Muiden kuin tiettyihin teknologioihin liittyvien osaamistarpeiden kohdalla (erityisesti niin kutsuttujen pehmeiden taitojen kohdalla) tulkinnan merkitys kuitenkin korostuu. Jos kirjoittaja kertoo olleensa kolmessa palaverissa päivän aikana ja kokeneensa hankalaksi ehtiä tehdä muut työtehtävät palaverien ohella, voidaanko tämän perusteella mennä niin pitkälle, että väitetään aineiston perusteella, että osaamistarpeita olivat yhteistyötaidot ja ajanhallintataidot? Pehmeiden taitojen poisjättäminen ja pelkkiin teknologioihin keskittyminen tulkinnanvaraisuuden minimoimiseksi ei kuitenkaan tuntunut oikealta, sillä aineiston alustavassa taulukoinnissa havaitsimme, että monet kirjoittajat olivat erityisesti nostaneet omiksi kehittymiskohteikseen juuri pehmeitä taitoja. Näitä taitoja pidettiin siis tärkeinä työelämässä.

Näkemyistä tukevat myös Hynnisen, Hujalan ja Knutaksen (2022) tutkimustulokset. He analysoivat valtakunnallisen uraseurantakyselyn 450 avointa vastausta selvittääkseen, mitä taitoja uransa alkuvaiheessa olevat ohjelmistoalan ammattilaiset suosittelevat opiskelijoille ja miten näitä suosituksia voidaan hyödyntää koulutusohjelmien kehittämisessä. Tutkimuksessa keskeisimmiksi suositelluiksi osaamisalueiksi nousivat pehmeät taidot, ohjelmointiosaaminen ja opintojen aikainen käytännön kokemus. Pehmeiden taitojen suosittelu oli erityisesti yhteydessä korkeaan uratyytyväisyyteen, mikä korostaa niiden merkitystä työelämässä menestymisen kannalta. (Hynninen, Hujala & Knutas 2022.)

Halusimme löytää tasapainon siihen, että emme tee aineistosta liian pitkälle meneviä tulkintoja, mutta saisimme kuitenkin poimittuja pehmeisiin taitoihin liittyviä teemoja sieltä, missä ne oli mainittu. Päädyimme linjaukseen, että jos kirjoittaja eksplisiittisesti mainitsi tietyn taidon työhönsä liittyen, otimme taidon mukaan. Emme kuitenkaan liiaksi poimineet rivien välistä pehmeitä taitoja, joita ei ollut erikseen mainittu. Tällä tavoin uskomme saavamme kiinni niistä asioista, jotka kirjoit-

taja itse on halunnut nostaa esiin juuri ammatillisen kehittymisen ja osaamistarpeiden näkökulmasta. Näin rajaten pois jäivät kirjoittajien kuvaukset, joista voisi vain epäsuorasti päätellä, että tietty taito on vaadittu.

Teknologioiden suhteen päätimme tehdä selkeän rajauksen siten, että otamme mukaan suorat maininnat teknologioista ja taidoista, ja sellaiset ilmaukset, joista voi aukottomasti päätellä tietyn teknologian olleen kirjoittajan työtehtävissä käytössä. Teknologioiden poimiminen oli kuten mainittua melko selkeä tehtävä, mutta yksittäisen havainnon luokittelu tiettyyn kategoriaan ei aina ollut täysin yksiselitteistä. Sama teknologia voi olla monessa eri käytössä, ja opinnäytetöistä löytyvät ilmaukset käytöstä eivät aina olleet täysin tarkkoja. Pyrimme teknologioiden luokittelussa siihen, että teknologia luokiteltiin ensisijaisesti sinne, mikä teknologian käyttötarkoitus tulkintamme perusteella oli kyseisessä kontekstissa. Joku muu voisi kuitenkin päätyä sisällönanalyysiä samasta aineistosta tekemällä hiukan toisenlaisiin jaotteluihin, mikä on hyvä tiedostaa tuloksia tarkastellessa.

Pyrimme lisäksi kahden tutkijan yhteistyössämme johdonmukaiseen keskinäiseen luokitteluun, jotta tutkimuksemme sisällä valinnat olisivat yhdenmukaisia. Teimme analyysia rinnakkain ja vertailimme tuloksia ja kategorioita, ja uudelleenluokittelimme analyysimme edetessä eri kategorioihin päätyneitä teknologioita niin, että päädyimme yhteiseen näkemykseen. Kirjasimme kaikki havainnot yhteiseen Excel-taulukkoon, jossa pystyimme vertailemaan tekemiämme poimintoja.

Sisällönanalyysia varten keräsimme tutkimukseen mukaan valitut opinnäytetyöt Excelliin, kunkin omalle rivilleen, ja etenimme sen jälkeen seuraavasti:

1. **Aineistoon tutustuminen:** Ennen tarkkaa analyysiä luimme opinnäytetöitä kursorisesti läpi ja muodostimme käsityksen aineiston sisällöstä.
2. **Poiminta ja koodaaminen:** Kävimme opinnäytetyöt systemaattisesti läpi ja poimimme teksteistä maininnat ohjelmistokehityksessä tarvittavista teknologioista, taidoista ja työtehtävistä. Havainnot merkittiin Excel-taulukkoon.
3. **Luokittelu ja kategorioiden muodostaminen:** Aloimme luokitella poimintoja kategorioihin ja yhdistellä niitä laajemmiksi aiheiksi. Loimme uusia sarakkeita sitä mukaa, kun analyysi eteni ja aineistosta nousi esiin uusia aiheita. Kun huomasimme, että jotkut sarakkeet sisältsivät samaan aiheeseen kuuluvia mainintoja, yhdistimme sarakkeita.
4. **Vertaisarviointi ja luokittelujen tarkistus:** Teimme analyysia rinnakkain, ja vertailimme tekemiämme poimintoja ja tarkistimme tulkintojamme. Keskustelimme vastaan tulleista rajatapauksista ja päätimme yhdessä, miten luokittelimme ne. Tarvittaessa luokittelimme tehtyjä poimintoja uudelleen ja yhdistimme jälleen samankaltaisia teknologioita tai taitoja.

5. **Analyysi ja tulosten kokoaminen:** Kun koko tekstimassa oli käyty läpi, muodostimme lopulliset aiheet Excel-tauluktoon. Yksi työkirjan rivi sisälsi aina yhdestä opinnäytetyöstä poimitut ilmaisut. Yksi sarake oli yksi aihekategoria, ja yhdessä solussa saattoi olla useampi samaan kategoriaan luokiteltava ilmaus pilkuilla eroteltuna.
6. **Määrällinen analyysi:** Kokosimme valitut tiedot määrällistä analyysia varten toiseen Excel-tauluktoon, jossa analysoimme dataa, teimme määrällisiä vertailuja ja laadimme tuloksia havainnollistavia kaavioita.
7. **Raportin kirjoittaminen:** Kirjasimme ylös yhteenvedon ja johtopäätökset tekemistämme havainnoista.

Vaikka teimme suuren osan poiminnoista opinnäytetöiden päiväkirjaosuuksista ja viikkoanalyysistä, emme rajanneet tutkimusaineistoa pelkästään näihin osioihin. Luimme kaikki opinnäytetyöt kokonaan läpi ja teimme poimintoja myös muista osista silloin, kun kirjoittaja esimerkiksi mainitsi johdannossa tai muussa tekstiosassa käyttäneensä tiettyä teknologiaa, menetelmää tai työskentelytapaa. Koska tavoitteena oli saada mahdollisimman kattava kuva kunkin kirjoittajan työtehtävissä vaaditusta osaamisesta, kaikki suorat maininnat, jotka vastasivat tutkimuskysymyksiin, otettiin huomioon riippumatta niiden sijainnista opinnäytetyössä.

Poimimme ja luokittelimme aineistosta mainintoja erilaisista teknologioista, työkaluista ja osaamisalueista. Kun saimme luokittelun valmiiksi, laskimme kunkin aiheen osalta, kuinka monessa opinnäytetyössä se mainittiin. Maininta tarkoittaa tässä yhteydessä sitä, että jos Excel-taulukon yhdellä rivillä tietyn aiheen sarakeessa (esimerkiksi "Ohjelmointikielät") oli mitä tahansa sisältöä, se laskettiin mukaan. Tästä saimme tiedon siitä, kuinka moni kirjoittaja oli käsitellyt kyseistä aihetta omassa työssään.

Tarkemmassa analyysissä otimme tämän jälkeen yksittäisen sarakkeen sisällön erilleen ja kävimme sen systemaattisesti läpi. Yhdessä solussa saattoi olla mainittuna useampi eri teknologia tai taito pilkuilla eroteltuna. Käsittelimme näistä jokaisen yksittäisen maininnan omana havaintonaan ja veimme esiintymät uuteen tauluktoon. Näin pääsimme tarkastelemaan sitä, mitä erilaisia nimettyjä teknologioita aineistossa esiintyi. Pystyimme tekemään myös tarkempaa määrällistä vertailua: esimerkiksi edellä mainittujen ohjelmointikielten kohdalla saatoimme laskea, kuinka monta kertaa kukin kieli mainittiin aineistossa, vaikka samassa opinnäytetyössä olisi mainittu useampi kuin yksi kieli. Esittelemme seuraavassa luvussa tutkimuksemme keskeiset tulokset.

4 Tutkimuksen tulokset

Tässä luvussa esittelemme analyysimme tulokset. Työn laajuuden rajaamiseksi emme käy kaikkia tekemiämme havaintoja yksityiskohtaisesti läpi, vaan keskitymme erityisesti niihin taitoihin ja teknologioihin, jotka sekä esiintyivät aineistossa yleisimmin että olivat arviomme mukaan merkityksellisiä tutkimuksen tavoitteen eli ohjelmistokehityksen osaamistarpeiden tutkimisen kannalta.

Muodostimme tekemämme luokittelun pohjalta yhteensä 39 aihetta, jotka kuvaavat aineistossa esiintyneitä ohjelmistokehityksen osaamistarpeita. Kaikki aiheet on listattu liitteessä 1. Aloitamme luvun esittelemällä yleisimmin esiin nousseet aiheet, ja jatkamme valittujen aiheiden tarkemmalla analyysillä.

4.1 Ohjelmistokehityksen keskeiset aiheet aineistossa

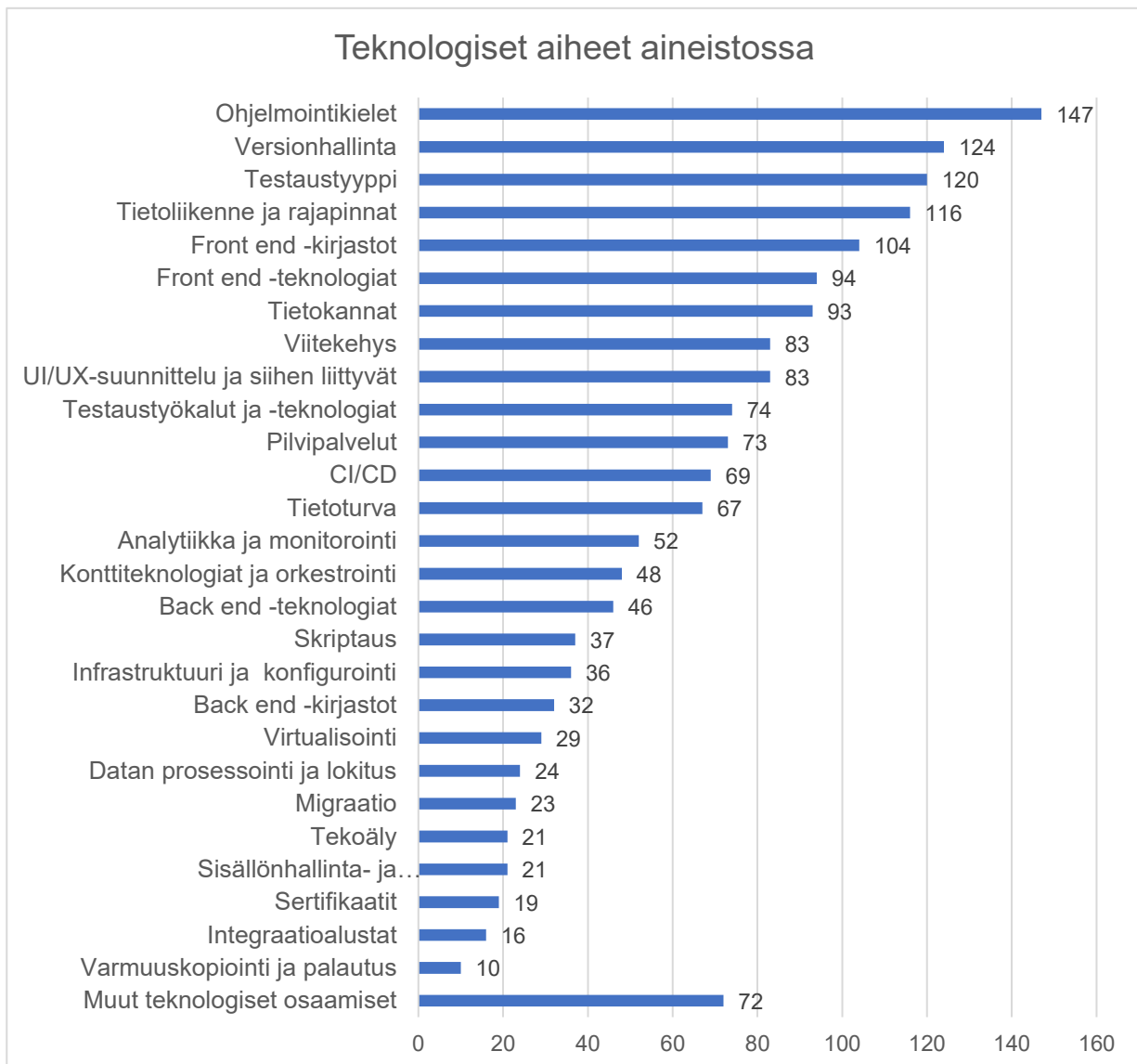
Tarkastelemme ensin niitä aiheita, jotka nousivat aineistosta esiin kaikkein useimmin. Ohjelmointikieliet olivat odotetusti yleisin mainittu aihe: peräti 147 opinnäytetyötä sisälsi mainintoja käytössä olleista kielistä, mikä kuvastaa ohjelmoinnin ilmeistä ja keskeistä roolia ohjelmistokehityksessä. Toistuvat ohjelmistokehityksen tehtävät, kuten bugien korjaus tai uusien ominaisuuksien kehittäminen, mainittiin 142 työssä. Versionhallinnan käyttö on kiinteä osa ohjelmointia ja koodin kanssa työskentelyä, ja versionhallinta mainittiinkin myös valtaosassa opinnäytetöitä (124). Useimmiten mainittu versiohallintajärjestelmä oli Git.

Pehmeät taidot (mainittiin 142 opinnäytetyössä) nousivat esiin lähes yhtä usein kuin ohjelmointikieliet. Tämä kertoo siitä, miten olennaisia pehmeät taidot ovat ohjelmistokehityksessä. Projektinhallinta- ja kommunikaatiotyökaluja, kuten Jira, Confluence ja Slack, oli mainittu niin ikään 142 työssä. Projektinhallintamenetelmiä, kuten Scrum tai yleisemmin ketterät menetelmät, oli mainittu 130 työssä, ja IDE eli kehitysympäristö ja siihen liittyviä työkaluja oli mainittu 110 työssä. Tiedonhaku, kuten dokumentaation, Stack Overflow:n tai muiden lähteiden hyödyntäminen, mainittiin 108 opinnäytetyössä.

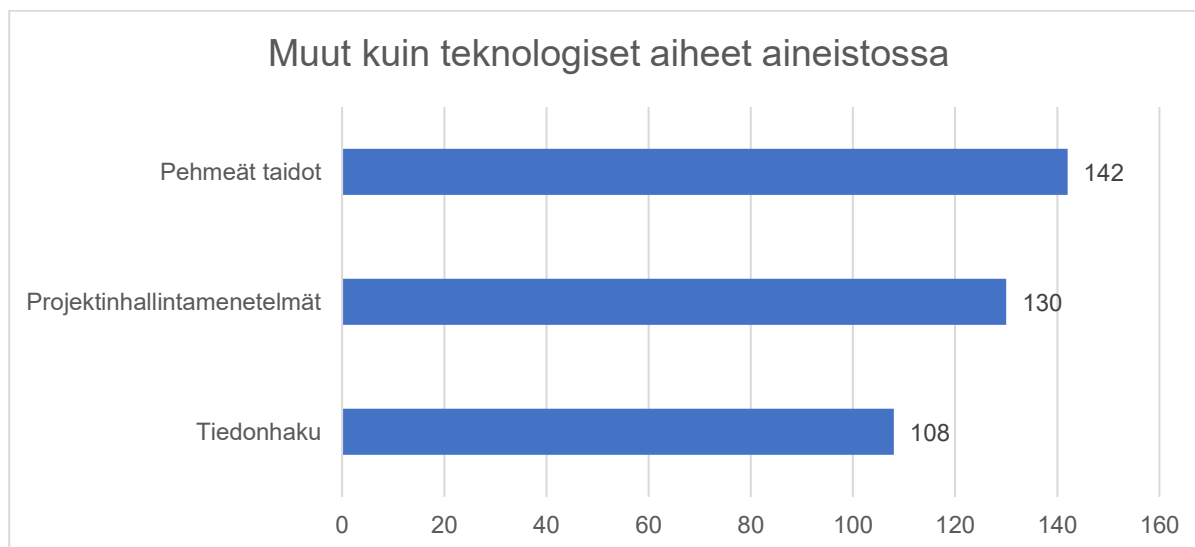
Testaaminen esiintyi aiheena aineistossa myös yleisesti: erilaisia testaustyyppisiä (esim. yksikkötestaus, automaatiotestaus) mainittiin 120 työssä, ja testaustyökaluja tai -teknologioita nimettiin 74 työssä. Tietoliikenne ja rajapinnat (116 mainintaa) olivat vahvasti esillä, erityisesti API sekä REST-rajapinnat. Front end -kirjastot (104 mainintaa, yleisin nimetty kirjasto oli React) ja front end -teknologiat kuten HTML ja CSS (94 mainintaa) olivat yleisesti mainittuja. Tietokannat mainittiin 93 opinnäytetyössä. Lisäksi UI/UX-suunnittelu ja siihen liittyvät työkalut mainittiin 83 työssä. Samoin jokin viitekehys mainittiin 83 työssä.

Pilvipalvelut (73) ja CI/CD-prosessit (69) olivat myös yleisesti mainittuja. Tietoturvaan liittyvä osaaminen tai teknologiat mainittiin 67 opinnäytetyössä. Back end -teknologiat (46 mainintaa) ja back end -kirjastot (32 mainintaa) olivat yleisesti käytettyjä, mutta kuitenkin huomattavasti harvemmassa työssä mainittuja kuin front end -osaamiset. Tämä selittyy todennäköisesti aineiston valikoituneisuudella: suuri osa kirjoittajista työskentely roolissa, jossa oli front end -painotus, ja myös full stack -kehittäjän nimikkeillä työskentelevistä useimmat mainitsivat front end -taitoja ja -teknologioita.

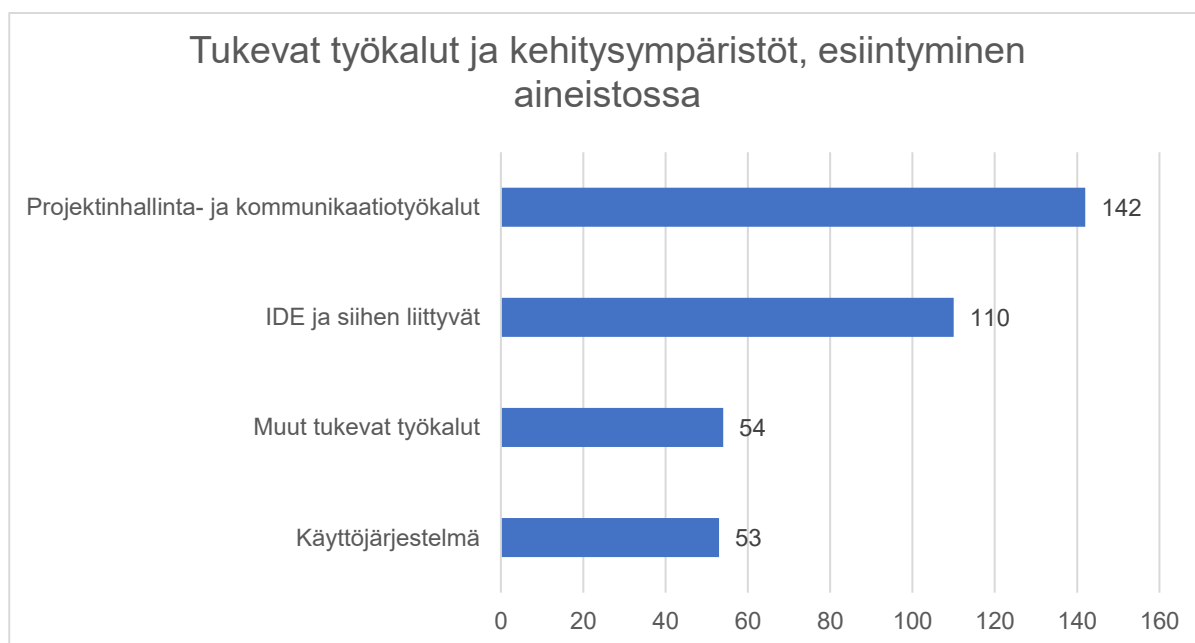
Yllä mainitut aiheiden mainintamäärät havainnollistetaan myös kuvissa 4–6. Omiin kaavioihinsa on erotettu teknologiset aiheet (kuva 4), työtapoihin liittyvät ja ei-teknologiset aiheet (kuva 5) sekä tukevat työkalut ja kehitysympäristöön liittyvät aiheet (kuva 6).



Kuva 4. Teknologiset aiheet ja teknologioiden maininnat aineistossa (N=154)



Kuva 5. Ei-teknologisiin aiheisiin liittyvät maininnat opinnäytetöissä (N=154)



Kuva 6. Tukevien työkalujen ja kehitysympäristöjen käyttöön liittyvät maininnat opinnäytetöissä (N=154)

Aineistosta löytyi lukuisia muitakin teemoja, joiden tutkiminen olisi ollut kiinnostavaa, mutta jotka eivät liittyneet osaamistarpeisiin. Toistuvasti esiin nousseita kiehtovia ilmiöitä olivat esimerkiksi huijarisyndrooma ja koronapandemian vaikutukset työhön. Pidimme kuitenkin mielessä Tuomen ja Sarajärven neuvon siirtää sivuun kaikki kiinnostava, mutta tutkimustehtävään liittymätön, mikä tulee esiin aineistoon tutustuesssa (Tuomi & Sarajärvi 2018, luku 4.1).

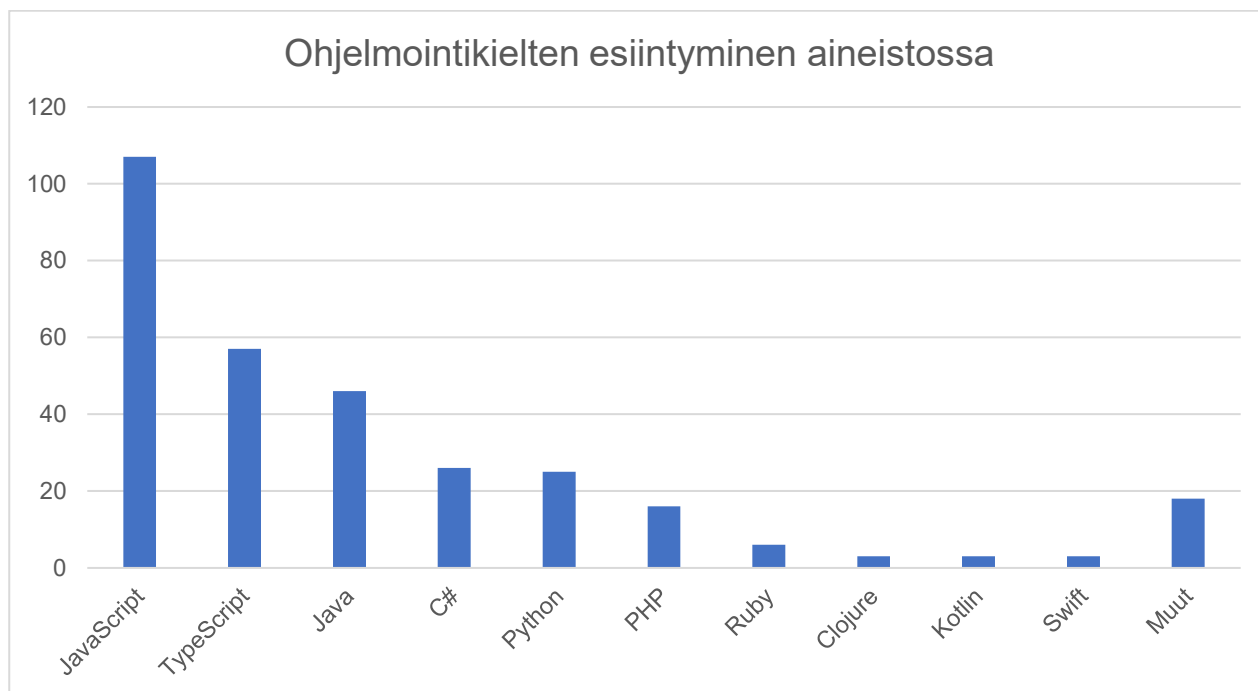
Seuraavissa luvuissa analysoimme keskeiset alakategoriat tarkemmin. Valitsimme yksityiskohtaisempaan tarkasteluun seuraavat osaamisalueet: ohjelmointikieliet, tietokannat, viitekehykset, testaus, pilvipalvelut, ohjelmistokehityksen toistuvat työtehtävät sekä pehmeät taidot.

4.2 Ohjelmointikieliet

Aineistossa esiintyi laaja kirjo ohjelmointikieliä, mutta JavaScript ja TypeScript erottuivat joukosta selvästi yleisimpinä. Kaikkein laajimmin käytetty kieli oli JavaScript, joka esiintyi 107 opinnäytetyössä. Toiseksi eniten mainintoja sai TypeScript (57 mainintaa), ja kolmanneksi sijoittui Java (46 mainintaa). Myös C# (26 mainintaa) ja Python (25 mainintaa) esiintyivät kohtuullisen usein, kuten myös PHP (16 mainintaa). Harvinaisempia mutta aineistossa esiintyneitä kieliä olivat Ruby (6 mainintaa) sekä Clojure, Kotlin ja Swift (3 mainintaa kukin). Aineistossa oli lisäksi useita yksittäisiä tai vain muutamassa opinnäytetyössä mainittuja ohjelmointikieliä, jotka on ryhmitelty "Muut"-kategoriaan (18 mainintaa). Näitä olivat muun muassa Dart, Go, Groovy, C, C++ ja Visual Basiciin pohjautuvat kielet. Ohjelmointikielten esiintyvyys aineistossa on havainnollistettu kuvassa 7.

SQL ja muut tietokantojen kyselykielet on rajattu tämän osion ulkopuolelle, vaikka ne usein rinnastetaan ohjelmointikieliin. SQL-kieli on kuitenkin tutkimuksessamme käsitelty erikseen tietokantojen yhteydessä, koska sen käyttö liittyy ensisijaisesti tiedon käsittelyyn ja kyselyihin tietokannoista, eikä se ole samalla tavalla yleiskäyttöinen ohjelmointikieli kuin muut tässä mainitut kielet. Myöskään merkintäkieliä, kuten HTML tai XML, tai tyylimäärytyksiin käytettävää CSS:ää ei ole otettu mukaan tähän ohjelmointikielten tarkasteluun.

JavaScriptin ja TypeScriptin vahva edustus aineistossa kertoo osaltaan siitä, että opinnäytetöissä kuvatuissa työtehtävissä painottui vahvasti front end- ja full stack -kehitys. TypeScript on ohjelmointikieli, joka laajentaa JavaScriptiä lisäämällä staattisen tyyppityksen ja muita ominaisuuksia (Microsoft s.a.), ja on todennäköistä, että moni JavaScript-kehittäjä käyttää myös TypeScriptiä. Java oli selkeästi yleisin perinteinen back end -ohjelmointikieli aineistossa. Se sai lähes tuplasti mainintoja verrattuna C#:iin tai Pythoniin. Java on alun perin julkaistu vuonna 1995 (Oracle 2025), ja kielen suosion on todettu olevan laskussa tietyissä yhteyksissä (Peslak & Conforti 2020, 267), mutta tämän aineiston perusteella Java on edelleen merkittävässä roolissa ohjelmistokehityksessä. Sen sijaan Python sai yllättävän vähän mainintoja (25 kappaletta) ottaen huomioon sen viime vuosina saaman media- ja muun näkyvyyden (Cass 2024; Jansen 2025).



Kuva 7. Ohjelmointikielten esiintyminen aineistossa (N=154)

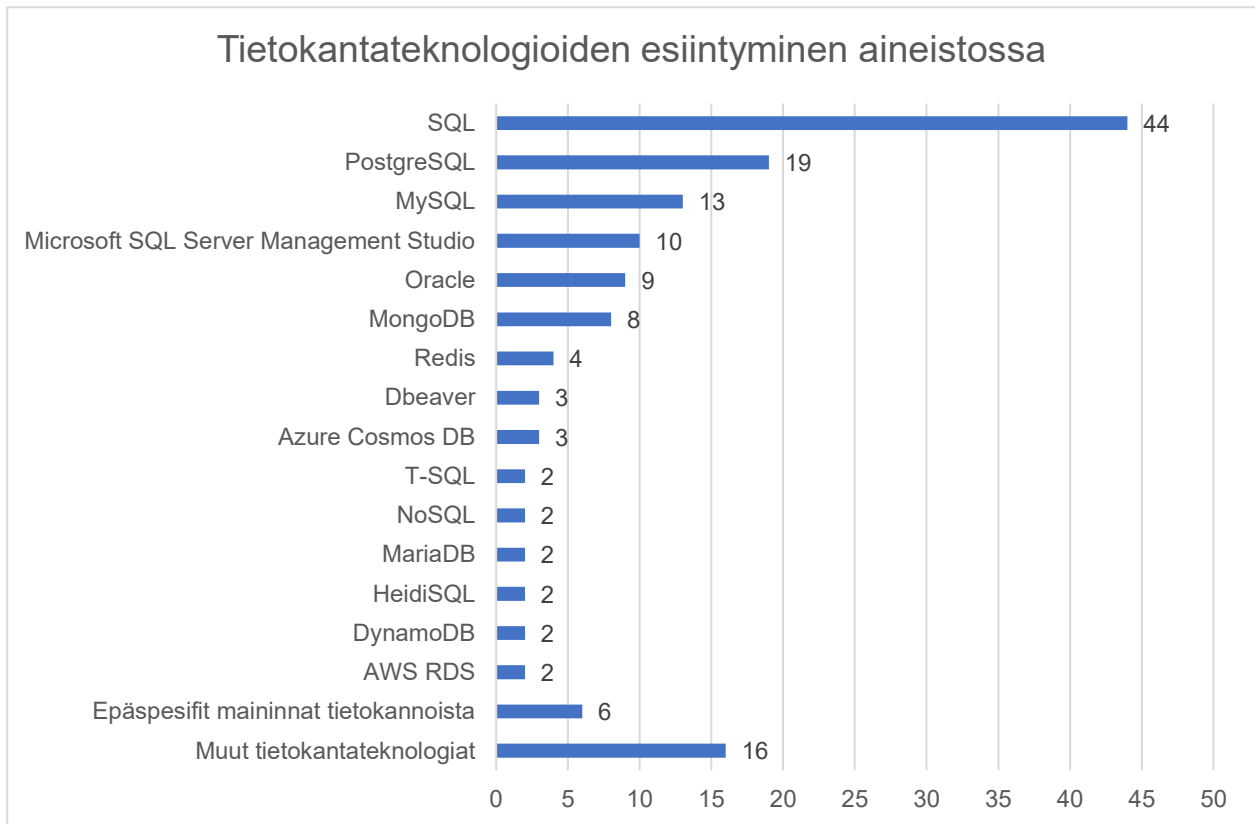
4.3 Tietokannat

Tietokantoihin liittyvistä teknologioista eniten mainintoja sai SQL-kyselykieli (44 mainintaa). SQL mainittiin usein yleisesti, ilman tarkempaa viittausta siihen, missä ympäristössä tai järjestelmässä sitä käytettiin. Laaja esiintyvyys kertoo siitä, että SQL:n hallinta (kuten tietokantaosaaminen ylipääntään) on olennainen osa ohjelmistokehittäjän perusosaamista. Yleisimmät nimeltä mainitut tietokannat aineistossa olivat PostgreSQL (19 mainintaa) ja MySQL (13 mainintaa). Lisäksi Microsoft SQL Server Management Studio (10 mainintaa) ja Oracle (9 mainintaa) olivat yleisesti mainittuja tietokantaratkaisuja. MongoDB (8 mainintaa) oli yleisin nimetty NoSQL-tietokanta. Muita useampia mainintoja saaneita tietokantaratkaisuja ja -teknologioita olivat Redis (4 mainintaa) Azure Cosmos DB (3 mainintaa), Dbeaver (3 mainintaa) ja AWS RDS (2 mainintaa).

Aineistossa oli myös 16 yksittäistä teknologian mainintaa, jotka on koottu "Muut tietokantateknologiat"-kategoriaan. Näihin kuuluvat muun muassa SQLite, Google Cloud Firestore ja AWS Aurora. Nämä yksittäiset maininnat kertovat osaltaan työtehtävissä käytettyjen tietokantateknologioiden monimuotoisuudesta. Tarkemmin määrittlemättömiä mainintoja tietokannoista oli kuusi. Nämä sisälsivät esimerkiksi mainintoja tietokannan käytöstä ilman tarkempaa teknologian nimeä tai kontekstia. Tietokantateknologioiden esiintyvyys aineistossa on esitetty kuvassa 8.

Nämä tulokset vahvistavat käsitystä siitä, että tietokannat, etenkin relaatiotietokannat (ja näistä erityisesti PostgreSQL, MySQL ja Microsoft SQL Server) ovat työelämässä yleisesti käytössä olevia

tietokantaratkaisuja. SQL:n mainintojen korkea esiintyvyyys korostaa sitä, että kielen osaaminen on ohjelmistokehityksessä perustaito ja keskeinen osaamistarve.



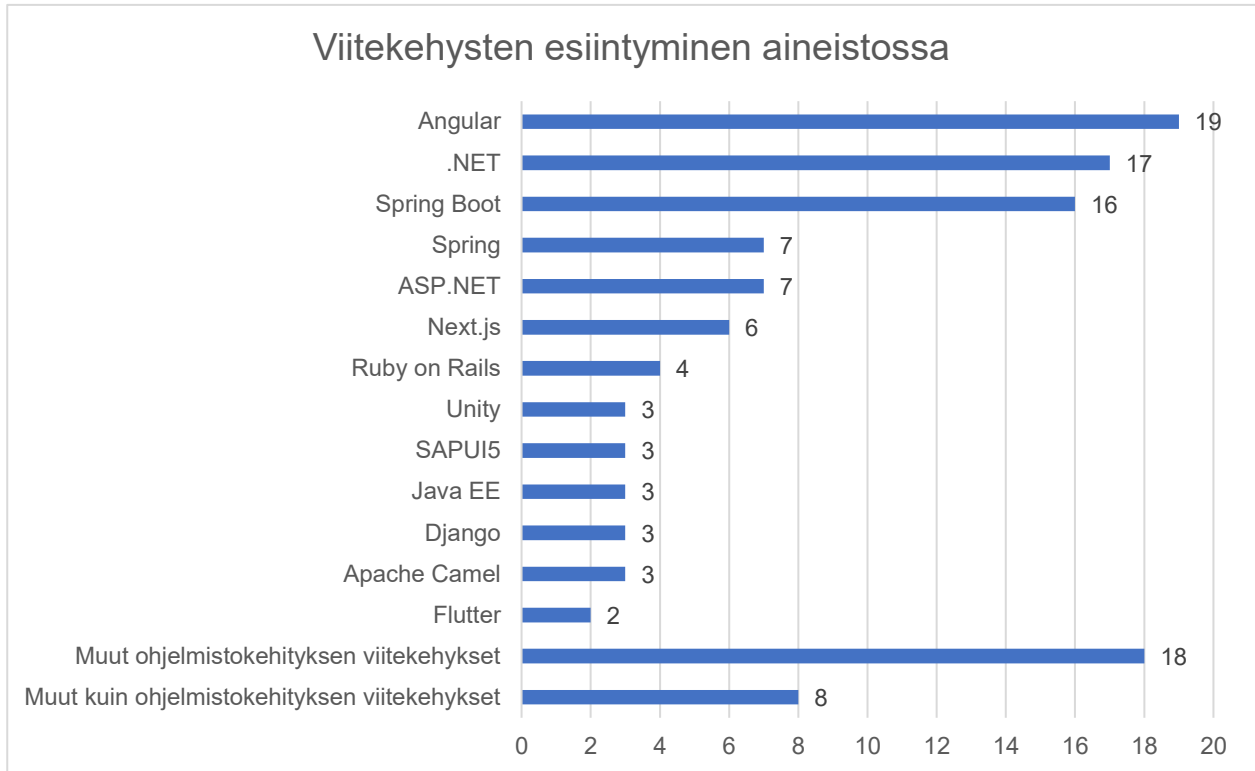
Kuva 8. Tietokantoihin ja kyselykieliin liittyvien teknologioiden esiintyminen aineistossa (N=154)

4.4 Viitekehykset

Vähintään yksi viitekehys mainittiin yhteensä 83 opinnäytetyössä (N=154), eli yli puolessa aineistosta. Tarkoitamme viitekehysillä tässä kontekstissa kehitysympäristöjä ja alustoja, joiden avulla ohjelmistoja rakennetaan systemaattisesti tietyllä tavalla. Aineistossa yleisimmin mainitut viitekehykset olivat TypeScript-kehitykseen käytettävä Angular (19 mainintaa), Microsoftin .NET (17 mainintaa) ja Java-ekosysteemissä käytettävä Spring Boot (16 mainintaa). Muita useammassa opinnäytetyössä mainittuja viitekehysiksi olivat Spring (7 mainintaa), ASP .NET (7 mainintaa) sekä Next.js (6 mainintaa). Huomionarvioista on, että luokittelimme JavaScript-kehityksessä käytettävän Reactin front end -kirjastoja joukkoon, eli se ei näy tässä osiossa.

Aineistosta löytyi myös muutamia viitekehysmainintoja, jotka eivät liity ohjelmistojen rakentamiseen, vaan esimerkiksi ohjelmistokehityksen prosesseihin. Tällaisia olivat esimerkiksi SAFe (Scaled Agile Framework) ja ITIL. Näitä mainittiin kahdeksassa opinnäytetyössä, ja ne on ryhmitelty

erilliseen kategoriaan "Muut kuin ohjelmistokehityksen viitekehykset". Muiden kuin ohjelmistokehityksen viitekehysten esiintyminen aineistossa kertoo osaltaan ohjelmistokehittäjän työn ja tarvittavan osaamisen monimuotoisuudesta.



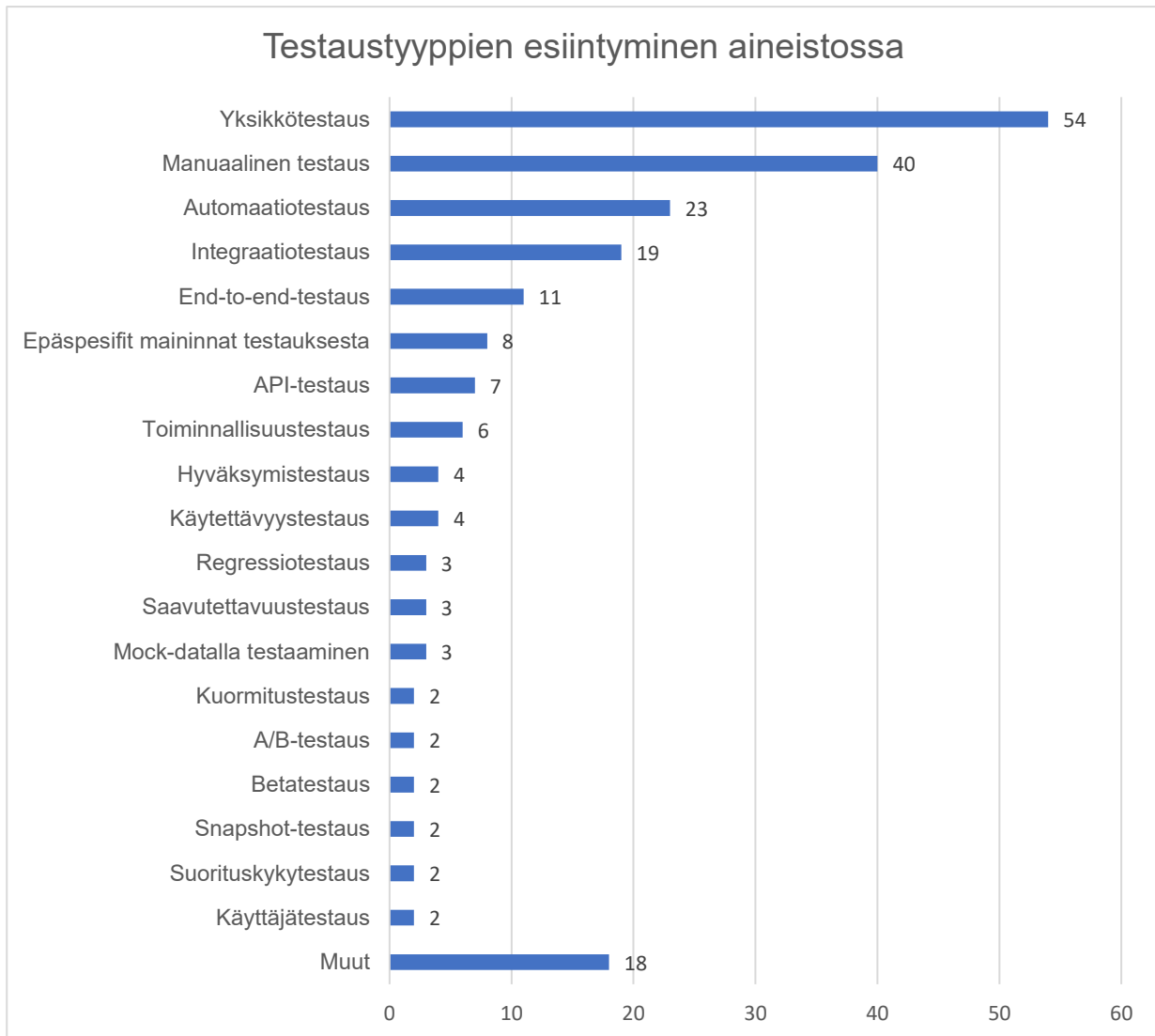
Kuva 9. Viitekehysten esiintyminen aineistossa (N=154)

4.5 Testaus ja testausteknologiat

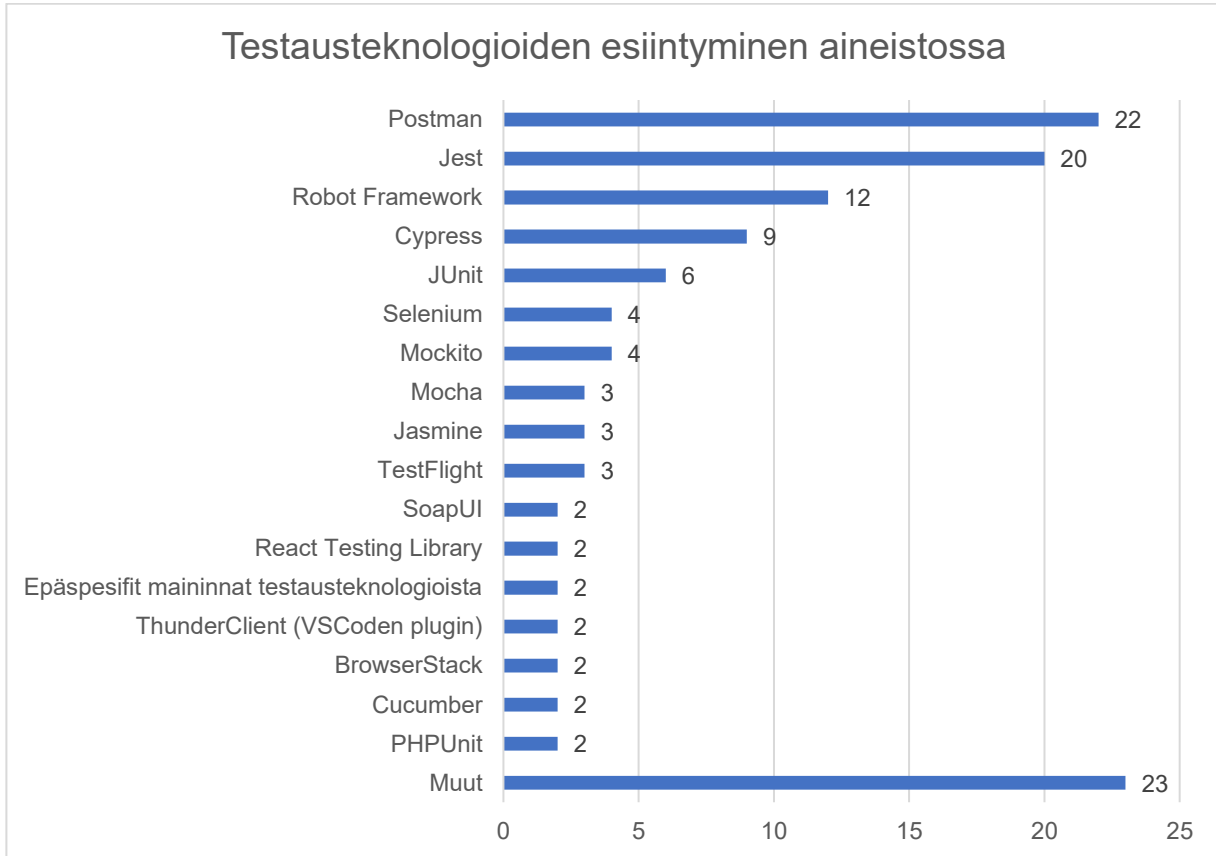
Testaus mainittiin ainakin jossain muodossa lähes kaikissa aineistoon kuuluvissa opinnäytetyöissä. Aineistossa oli kahdentasoisia mainintoja testauksesta: 1) mainintoja testaustyypeistä (mainintoja 120 opinnäytetyössä) ja 2) nimettyjä testaukseen liittyviä työkaluja ja teknologioita (mainintoja 74 opinnäytetyössä). Maininnat testaustyypeistä on eritelty kuvassa 10, ja aineistossa esiintyneet testausteknologiat on kuvattu kuvassa 11.

Tulosten perusteella yksikkötestaus (54 mainintaa) ja manuaalinen testaus (40 mainintaa) olivat yleisimmät testausmuodot, joita kirjoittajat tarvitsivat työtehtävissään. Kolmanneksi yleisin oli automaatiotestaus (23 mainintaa). Aineistosta löytyi myös laajasti erilaisia testaustyyppisiä, mistä voi päätellä, että ohjelmistokehittäjät osallistuivat testaukseen yksikkötesteistä aina end-to-end-testaukseen (11 mainintaa) ja valmiin sovelluksen hyväksymistestaukseen (4 mainintaa) asti.

Nimetyistä testausteknologioista yleisimpiä olivat Postman (22 mainintaa), Jest (20 mainintaa), Robot Framework (12 mainintaa), Cypress (9 mainintaa) ja JUnit (6 mainintaa). Testausteknologioiden Muut-kategoriassa oli peräti 23 teknologiaa, joista jokainen mainittiin vain kerran, mikä kertoo työelämässä käytettävien testausteknologioiden moninaisuudesta. Yksittäisiä mainintoja saaneita testausteknologioita olivat muun muassa Pytest, NUnit ja Fiddler.



Kuva 10. Testaustyyppien esiintyminen aineistossa (N=154)

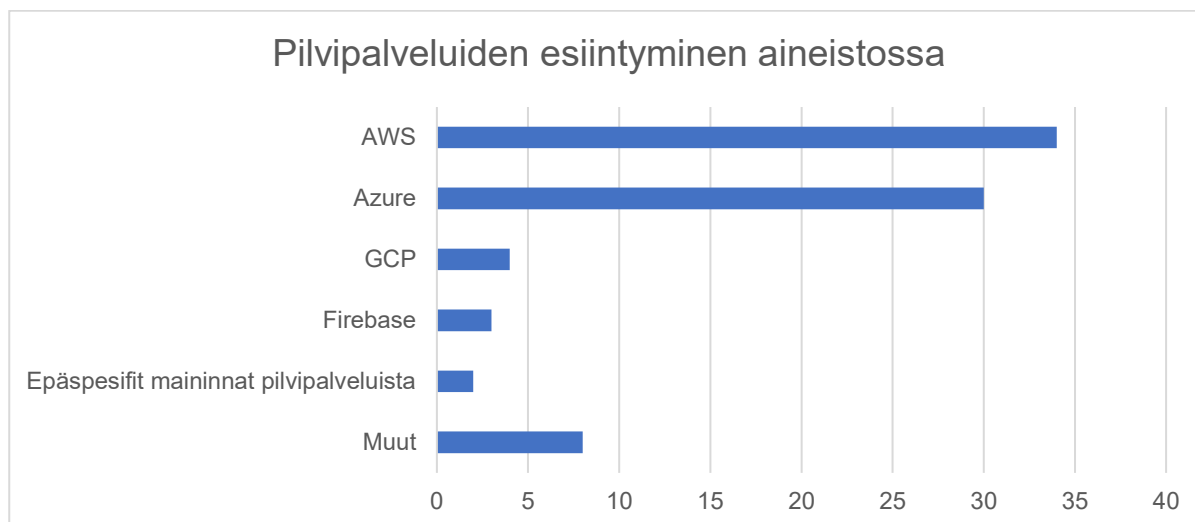


Kuva 11. Testaustyökalujen ja -teknologioiden esiintyminen aineistossa (N=154)

4.6 Pilvipalvelut

Pilvipalvelut mainittiin 73 opinnäytetyössä. Selvästi yleisimmin mainitut alustat olivat Amazon Web Services (AWS), joka esiintyi 34 opinnäytetyössä sekä Microsoftin Azure, joka mainittiin 30 opinnäytetyössä. Kolmantena tuleva Google Cloud Platform (GCP) oli selvästi harvinaisempi, saaden vain neljä mainintaa. AWS:n ja Azuren huomattavasti laajempi edustus aineistossamme kertoo siitä, että nämä kaksi suurinta pilvipalvelua ovat olleet keskeisiä kirjoittajien työssä.

Havaintojamme tukevat myös Synergy Research Groupin (2025) tuoreimmat markkina-analyysit, joiden mukaan vuoden 2025 ensimmäisellä neljänneksellä AWS hallitsi 29 % pilvi-infrastruktuuri-markkinasta, Azure 22 % ja Google Cloud 12 %. Nämä kolme suurinta toimijaa kattoivat yhteensä 68 % julkisen pilven markkinaosuudesta. (Synergy Research Group 2025.) Sekä AWS että Azure tarjoavat laajasti työkaluja mm. sovellusten julkaisuun, tietokantoihin, autentikointiin ja CI/CD-käytänteisiin.



Kuva 12. Pilvipalveluiden esiintyminen aineistossa (N=154)

4.7 Toistuvat ohjelmistokehityksen tehtävät

Aineistosta nousi esiin toistuvia käytännön työtehtäviä, jotka liittyivät ohjelmistokehitykseen, mutta jotka oli vaikeaa jaotella kuulumaan tiettyyn aiheeseen, tai jotka sopivat samanaikaisesti useaan kategoriaan. Näiden aiheiden poiminta oli myös tulkinnallisempaa kuin varsinaisten teknologioiden tai työkalujen mainintojen kirjaaminen, koska niihin liittyvät ilmaisut olivat aineistossa vaihtelevampia ja moniselitteisempiä. Halusimme kuitenkin sisällyttää nämä tehtävät tutkimukseen, koska ne havainnollistavat konkreettisesti, mistä ohjelmistokehittäjän päivittäinen työ voi koostua. Ryhmittelimme toistuvat tehtävät viiteen laajempaan aiheeseen ja esittelemme tässä ne aiheet, joihin liittyviä mainintoja oli eniten (aihe mainittu vähintään 40 opinnäytetyössä). Näitä aiheita olivat:

1. Koodin laadun varmistaminen ja ylläpito (144 mainintaa):

Tämä kattoi koodin refaktoroinnin, optimoinnin, sekä koodikatselmoinnin. Koodin laadun ylläpito oli näin kaikkein yleisin toistuva ohjelmistokehityksen tehtävä.

2. Virheiden korjaaminen ja ohjelmakoodin tutkiminen (140 mainintaa):

Bugien korjaus ja debuggaus olivat lähes jokaisen kirjoittajan arjessa esiintyviä tehtäviä. Näiden mainintojen yleisyys kertoo siitä, että ongelmanratkaisu ja virheiden korjaus ovat keskeinen osa ohjelmistokehittäjän työtä.

3. Dokumentointi ja tiedon jakaminen (47 mainintaa):

Monissa opinnäytetyöissä mainittiin teknisen dokumentaation laatiminen, tiedon kerääminen sekä tiedon jakaminen tiimin sisällä. Tämä korostaa dokumentaation merkitystä kehittäjän työtä tukevana käytäntönä.

4. Uusien ominaisuuksien kehittäminen (44 mainintaa):

Monessa työssä mainittiin uusien ominaisuuksien kehittäminen joko yksin tai parikoodauksena, sekä erilaisia kehittämisen kohteita kuten komponenttien tai mikropalvelujen kehittäminen.

5. Kehitysprosessiin osallistuminen (40 mainintaa):

Ohjelmistokehittäjät osallistuivat kehitysprosessin eri vaiheisiin ja työhön kuului muun muassa palavereihin osallistumista, retrospektiivejä ja tehtävien suunnittelua. Tämä kertoo ohjelmistokehittäjän työn monipuolisuudesta: työ ei ole pelkkää ohjelmointia, vaan työssä tarvitaan muutakin kehitysprosessiin liittyvää osaamista. Kehitysprosessin käytäntöihin ja vaiheisiin liittyvät tehtävät linkittyvät usein myös pehmeisiin taitoihin, joita käsittelemme seuraavassa luvussa.

4.8 Pehmeät taidot

Aineistosta nousi esiin runsaasti erilaisia pehmeitä taitoja, joita opinnäytetöiden kirjoittajat kuvasivat tarvinneensa ja käyttäneensä työssään. Ryhmittelimme ne 12 yläkategoriaan, joiden avulla jäsenämme ja tarkastelemme pehmeiden taitojen kokonaiskuvaa aineistossa. Pehmeisiin taitoihin viitattiin yhteensä 142 opinnäytetyössä 154:stä, eli suurimmassa osassa töitä mainittiin ainakin jokin pehmeä taito. Tämä kertoo pehmeiden taitojen tärkeydestä ja merkityksestä ohjelmistokehittäjän työssä. Useissa opinnäytetöissä esiintyi lisäksi mainintoja useammasta kuin yhdestä pehmeästä taidosta. Tätä moninaisuutta tarkastellaksemme laskimme myös yhteen jokaisen eri taidon maininnan opinnäytetyökohtaisesti, ja laskimme lopuksi yhteen nämä opinnäytetyökohtaiset eri taitojen maininnat. Tällä laskutavalla pehmeiden taitojen yhteismainintojen lukumääräksi muodostui kaiken kaikkiaan 408, eli yhdessä opinnäytetyössä mainittiin keskimäärin 2,6 pehmeää taitoa. Tämä kertoo siitä, että pehmeät taidot eivät olleet ainoastaan yleisesti mainittuja, vaan myös monipuolisesti edustettuja opinnäytetöissä.

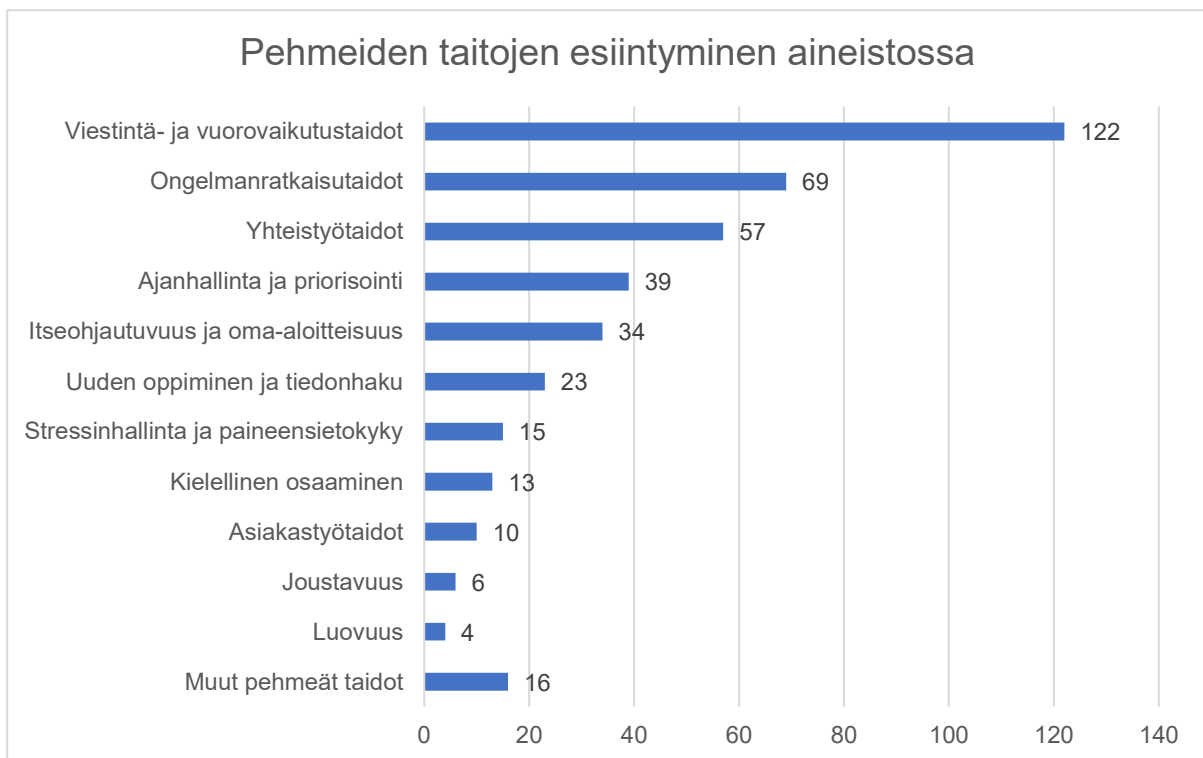
Selkeästi yleisimmät pehmeät taidot liittyivät viestintään ja vuorovaikutukseen (mainittu 122 opinnäytetyössä). Tähän kategoriaan sisältyivät muun muassa kommunikaatiotaidot, vuorovaikutustaidot, viestintä ja esiintymistaidot. Havainto on linjassa aiemman tutkimustiedon kanssa, jossa viestintätaidot on todettu keskeiseksi osaksi ohjelmistokehityksen työnkuvaa, etenkin kansainvälisissä ketterissä tiimeissä (Garbajosa, Yagüe & González 2014).

Toiseksi yleisin pehmeiden taitojen yläkategoria oli ongelmanratkaisukyky (69 mainintaa). Aineistossa mainittiin toistuvasti kyky ratkaista teknisiä ongelmia ja paikantaa virheitä esimerkiksi koodin debuggauksen yhteydessä. Ongelmanratkaisukykyä voikin pitää aivan keskeisenä ydinosana ohjelmistokehittäjän arjessa. Yhteistyötaidot (57 mainintaa) muodostivat kolmanneksi merkit-

tävimmän kokonaisuuden. Tähän luokittelimme maininnat tiimityöstä ja ryhmätyöskentelystä. Tulosten perusteella voidaan todeta, että ohjelmistokehitys ei ole yksilösuorittamista, vaan yhteistyö on keskeinen osa työn tekemisen tapaa.

Ajanhallinta ja priorisointitaidot (39 mainintaa), itseohjautuvuus ja oma-aloitteisuus (34 mainintaa) sekä uuden oppimisen ja tiedonhaun taidot (23 mainintaa) nousivat myös vahvasti esiin. Ne tukevat käsitystä siitä, että ketterässä ohjelmistokehityksessä tiimien jäseniltä edellytetään oma-aloitteisuutta, vastuullisuutta ja kykyä itsenäiseen työskentelyyn. Itsensä johtaminen ja jatkuva oppiminen ovat keskeisiä tekijöitä, jotka tukevat tätä työtettä. (Masood, Hoda & Blincoe 2020, 4963, 4981.)

Muita merkityksellisiä, vaikkakin pienempiä kategorioita olivat esimerkiksi stressinhallinta ja paineensietokyky (15 mainintaa), kielellinen osaaminen (13 mainintaa) sekä asiakastyötaidot (10 mainintaa). Myös joustavuus (6 mainintaa) ja luovuus (4 mainintaa) saivat useampia mainintoja. Muut pehmeät taidot -kategoriaan sijoitimme sellaiset yksittäisiä mainintoja saaneet taidot, jotka eivät selkeästi asettuneet muiden yläkategorioiden alle. Näitä olivat muun muassa keskittymiskyky, motivaatio, positiivinen asenne, tunnetaidot ja rohkeus.



Kuva 13. Pehmeisiin taitoihin liittyvien mainintojen esiintyminen aineistossa (N=154)

5 Pohdinta

Tässä luvussa peilaamme tutkimustamme ja sen tuloksia opinnäytetyömme tietoperustaan. Tarkastelemme, miten analyysimme tulokset sopivat osaamisen luokitteluihin, alan viitekehyksiin, aiempaan tutkimustietoon aiheesta sekä muihin tietoperustamme lähteisiin. Tavoitteena on arvioida, miten hyvin aineistosta löytyvät osaamistarpeet asettuvat osaksi laajempaa kuvaa ohjelmistokehityksen osaamistarpeista. Lopuksi luomme katsauksen tuloksissa näkyviin ajankohtaisiin ja ajallisiin ilmiöihin, tarkastelemme tutkimuksen luotettavuuden varmistamista ja pohdimme mahdollisia aiheita jatkotutkimukselle.

Kävimme läpi tietoperustan lähteitä, kirjoitimme osiota työn alkuvaiheessa ja jatkoimme osion työstämistä koko analyysiprosessin ajan. Koska tutkimuksemme pohjautui induktiiviseen sisällönanalyysiin, pyrimme parhaamme mukaan siihen, ettei tietoperusta ohjannut liiaksi havaintojen tekemistä. Halusimme antaa aineiston puhua ja tehdä luokittelumme aidosti siitä lähtökohdasta, mitä opinnäytetöistä nousi esiin. Tietoperustan rooli korostuu kuitenkin tulosten tulkintavaiheessa: jäsenämme ja arvioimme sen avulla havaintojamme ja pohdimme sitä, millaisia johtopäätöksiä voimme tehdä tutkimuksemme tuloksista.

5.1 Tulokset ja tietoperusta

5.1.1 Tulosten vertailu: Kvalifikaatioluokitus ja osaamisen typologia

Kuten tietoperustassa esitettiin, osaamista voidaan tarkastella useista toisiaan täydentävistä näkökulmista: kompetenssina, kvalifikaationa ja ammattitaitona (Hanhinen 2010, 96–98). Tässä tutkimuksessa tarkastelun kohteena ovat erityisesti kvalifikaatiot eli työn asettamat osaamisvaatimukset. Analyysin tukena käytimme kahta eri luokittelumallia: Opetushallituksen kolmitasoista kvalifikaatioluokitusta (Leveälahti & Nyyssölä 2019, 18–20) sekä Euroopan unionin osaamisen typologiaa (Balcar ym. 2011, 22). Aineistolähtöinen jaottelumme poikkeaa osin näiden mallien jaottelusta. Esimerkiksi kielitaito on analyysissämme luokiteltu pehmeäksi taidoksi, vaikka EU:n raportissa se kuuluu koviin taitoihin. Vertailu luokitusten ja aineistolähtöisen jaottelumme välillä on kuitenkin perusteltua, sillä se syventää ymmärrystä siitä, miten opinnäytetöissä esiin nousevat osaamiset suhteutuvat teoreettisiin malleihin. Samalla vertailu tuo esiin mahdollisia eroja teorian ja käytännön välillä.

Aineistossa toistuvimmin mainitut, ohjelmistokehittäjien rooleja yhdistävät teknologiset osaamisalueet liittyivät ohjelmointikieliin (147 mainintaa), versionhallintaan (124 mainintaa), testauksen eri tyypeihin (120 mainintaa), tietoliikenteeseen ja rajapintoihin (116 mainintaa), kehitysympäristöihin (110 mainintaa) sekä tietokantoihin (93 mainintaa). Nämä osaamiset voidaan luokitella osaamisen

typologiassa spesifeiksi koviksi taidoiksi, ja ne sijoittuvat Opetushallituksen luokituksessa ammattialakohtaisiin osaamisiin. Mainintojen määrä kertoo, että nämä teknologiset osaamisalueet muodostavat aineiston keskeisen ohjelmistokehitystyön osaamis pohjan.

Aineistossa esiintyi myös runsaasti mainintoja, jotka liittyvät siirrettäviin ja vuorovaikutuksellisiin taitoihin. Esimerkiksi projektinhallintamenetelmät (130) sekä tiedonhaku (108) voidaan luokitella yleiseksi työelämäosaamisiksi kvalifikaatioluokituksen mukaan sekä pehmeiksi taidoiksi EU:n typologiassa. Maininnoista nousee esiin myös viitteitä geneerisistä osaamisista, jotka eivät ole sidoksissa tiettyyn teknologiaan tai työtehtävään. Esimerkiksi ongelmanratkaisukyky (69 mainintaa) ja jatkuva oppiminen sekä tiedonhaku (23 mainintaa) edustavat siirrettäviä taitoja, joita voidaan hyödyntää laajasti eri ammatillisissa tilanteissa, ja tukevat yksilön työmarkkinakelpoisuutta (Leveälähti & Nyysölä 2019, 18–20).

Ohjelmistokehittäjän rooli näyttää aineistomme perusteella kattavan kaikki keskeiset osaamisalueet, joita kvalifikaatioluokitus ja osaamisen typologia määrittelevät. Kovat ja pehmeät taidot, sekä geneeriset, yleiset työelämäosaamiset ja ammattialakohtaiset osaamiset näyttävät kytkeytyvän aineistossamme vahvasti yhteen, toisiansa täydentäen. Tämä tukee tutkimustiedon mukaista käsitystä ohjelmistokehityksestä monialaisena asiantuntijatyönä, jossa eri osaamisalueet ovat tiiviissä vuorovaikutuksessa keskenään (Assyne, Ghanbari & Pulkkinen 2022, 9–10).

5.1.2 Tulosten vertailu: Raportit ja viitekehykset

Tässä alaluvussa vertaamme tutkimuksemme tuloksia (ks. tarkemmin luku 4) valitsemiimme viitekehyksiin. Tavoitteena on asettaa tutkimuksemme tulokset kansainväliseen kontekstiin ja arvioida, miltä osin tulokset tukevat viitekehysten määritelmiä, ja tarkastella myös mahdollisia eroavaisuuksia. Vertailukohtina ovat seuraavat neljä valitsemaamme kansainvälistä viitekehystä (ks. tarkemmin luku 2.5):

- IT2017 (Association for Computing Machinery & IEEE Computer Society 2017)
- SE2014 (IEEE Computer Society & Association for Computing Machinery 2015)
- SWEBOK V4.0 (IEEE Computer Society 2024)
- European ICT Professional Role Profiles (Breyer ym. 2018).

Ohjelmistokehityksen osaamiset vaikuttivat aineistomme tulosten perusteella olevan pääasiassa linjassa kansainvälisten viitekehysten kanssa. Mainintojen kanssa selkeästi eniten yhteneväisyyksiä löytyi European ICT Professional Role Profiles-viitekehyyksessä. Käytännössä kaikista tässä viitekehyyksessä esitetyistä työtehtävistä, osaamisalueista ja tuotoksista löytyi aineistossamme mainintoja. Ainoa poikkeus oli tuotoksiin kuuluva käyttäjäkokemussuunnitelma, jota ei aineistossamme esiintynyt suoraan. Tätä voi selittää se, että suuri osa opinnäytetöiden kirjoittajista työskentelivät jo

käytössä olevien ohjelmistojen parissa tai pitkälle edenneissä kehitysprojekteissa, joissa keskeiset käytettävyyteen liittyvät ratkaisut oli todennäköisesti tehty kehitysprosessin aiemmissa vaiheissa.

Yhteneväisyyksiä tutkimusaineiston ja viitekehysten välillä havaittiin erityisesti ohjelmoinnin perusosaamisen lisäksi viestintä- ja tiimityötaitoissa. Viestintä- ja tiimityötaitoja korostavat IT2017, SE2014 sekä SWEBOK: aineistossamme viestintä- ja vuorovaikutustaidot mainittiin 122 kertaa ja yhteistyö- sekä tiimityötaidot 57 kertaa. Edellä mainittujen lisäksi myös ajanhallinta- ja priorisointiosaamisen (39 mainintaa) voidaan tulkita sisältyvän SE2014:n Tiimityötaitot-osaamisalueeseen.

Ohjelmointikielet, ohjelmiston kehittäminen ja toteutus ovat osa-alueita, jotka mainitaan keskeisinä kaikissa viitekehyksissä, ja ne ovat vahvasti esillä myös analysoimassamme aineistossa. Näitä osa-alueita on kuitenkin haastavaa erottaa toisistaan, sillä ne kytkeytyvät tiiviisti moniin kirjoittajien kuvaamiin työtehtäviin ja voivat tulkinnan mukaan sisältää monia osaamisia itsessään. Vertailu voi kuitenkin havainnollistaa, että aineistomme havainnot viitekehysten mainintoihin ohjelmoinnin perusosaamisesta ovat linjassa. Esimerkiksi ohjelmointikieliin viitattiin 147 kertaa, ohjelmakoodin laadun varmistamiseen ja ylläpitoon viitattiin 144 kertaa, uusien ominaisuuksien kehittämiseen 44 kertaa, virheiden korjaamiseen ja ohjelmakoodin tutkimiseen 140 kertaa sekä ohjelmointikieliin 147 kertaa.

Testaus (124 mainintaa) ja dokumentointi (47 mainintaa) olivat aineistossamme myös laajasti edustettuina, vastaten muun muassa SWEBOK:in Ohjelmiston rakennus- ja Ohjelmistotestaus-osaamisalueita, SE2014:n teknisiä osaamisia sekä European ICT Professional Role Profiles -viitekehysten pääasiallisia työtehtäviä ja osaamisia. Tietoturvaosaaminen (67 mainintaa) näkyi autentikoinnin ja tietoturvatestauksen muodossa ja vastasi SWEBOK:in Tietoturva-osaamisaluetta sekä European ICT Professional Role Profiles-viitekehysten pääasiallisia työtehtäviä. Itsehjautuvuuden ja oma-aloitteisuuden (34 mainintaa) korostuminen tukee puolestaan sekä IT2017:n että SWEBOK:in elinikäisen oppimisen painotusta.

Monia osaamisalueita sivuttiin epäsuorasti, mutta ei esimerkiksi käsitelty järjestelmällisesti. Tämä saattaa liittyä aineiston kontekstiin ja tekstilajin erityispiirteisiin: päiväkirjamuotoiset opinnäytetyöt tuovat esiin ennen kaikkea käytännön kokemuksia ja henkilökohtaista oppimista, eivätkä välttämättä muodollisten viitekehysten tai ammatillisten standardien systemaattista esittämistä. Havainto voi kertoa siitä, että uran alkuvaiheen ohjelmistokehittäjien työ on käytännönläheistä ja painottuu soveltaviin ohjelmointitehtäviin enemmän kuin teoreettiseen mallintamiseen tai laskennalliseen analyysiin.

Osittaisia yhteneväisyyksiä havaittiin ratkaisujen suunnittelun ja käyttäjätarpeiden huomioimisen osalta. Vaikka IT2017 ja SE2014 korostavat järjestelmällistä ratkaisujen suunnittelua vaatimusten

pohjalta, aineistossamme suunnittelu liittyi lähinnä käyttöliittymien (UI/UX) ja saavutettavuuden kehittämiseen. Sidosryhmäviestintä, jota IT2017 ja SE2014 pitävät tärkeänä, sisältyi aineistossamme laajempaan viestinnän käsitteeseen ilman erillistä tarkastelua.

Yllättävää oli, että aineistomme maininnoista puuttuivat kokonaan järjestelmällinen vaatimusmäärittely ja vaatimusten analysointi, joita erityisesti SWEBOK ja SE2014 painottavat. Samoin käyttäjätarpeiden tunnistaminen esiintyi lähinnä epäsuorasti. Tämä saattaa selittyä sillä, että vaatimusmäärittely ja käyttäjätarpeiden tunnistaminen oli jo tehty ennen opinnäytetyön kirjoittajan osallistumista projektiin. Toinen mahdollinen selitys liittyy kokemustasoon: vaatimusmäärittely voi olla työtehtävä, joka jätetään kokeneemmille työntekijöille tai esimerkiksi liiketoimintapuolen asiantuntijoille, erityisesti kun siihen liittyy asiakkaiden tai muiden sidosryhmien kanssa käytävää vuorovaikutusta. Monesti vaatimusmäärittelyyn osallistuu ohjelmistokehittäjän lisäksi myös muut ohjelmistoprojektin sidosryhmät. (ISO/IEC/IEEE 29148:2018, 9.) Tämä voi johtaa tilanteeseen, jossa uransa alkuvaiheessa oleva ohjelmistokehittäjä ei ole mukana määrittelytyössä tai ei tunnista osallistuneensa siihen, vaikka olisi ollut esimerkiksi teknisen ratkaisun keskustelussa mukana.

Tulkinnanvaraisiksi osaamisalueiksi voidaan katsoa ongelmien analysointi, kompromissien tekeminen ja projektinhallinta. IT2017:n edellyttämä monimutkaisten ongelmien laskennallinen analysointi puuttui, vaikka toisaalta ongelmanratkaisukyky (69 mainintaa) oli vahvasti esillä. SE2014:n määrittelemä kyky tasapainottaa ristiriitaisia vaatimuksia näkyi epäsuorasti ajanhallinnan, stressinhallinnan ja joustavuuden maininnoissa. Projektinhallinnan osalta aineistossa mainittiin projektinhallinta- ja kommunikaatiotyökalujen käyttö (130 mainintaa) sekä projektinhallintamenetelmät (130 mainintaa), mutta prosessimallien arviointi ja kehittäminen, joita SWEBOK korostaa, jäi puuttumaan.

Merkittäviä puutteita havaittiin eettisen, oikeudellisen ja yhteiskunnallisen harkinnan alueilla, joita IT2017, SE2014 ja SWEBOK V4.0 painottavat. Nämä osaamisalueet puuttuivat aineistostamme kokonaan. Samoin taloudelliset näkökulmat, matemaattiset perusteet, insinööritieteiden perusteet sekä tietorakenteet ja algoritmit, jotka sisältyvät SWEBOK:in ja IT2017:n osaamisalueisiin, jäivät aineistossa vähäiselle huomiolle tai puuttuivat kokonaan.

Aineistostamme ei pääsääntöisesti löytynyt sellaisia mainintoja, jotka olisivat täysin puuttuneet tarkastelluista viitekehyksistä. Tähän liittyvä merkittävin ero näyttäisi olevan se, että aineistossamme esiintyy runsaasti tarkasti nimettyjä teknologioita ja työkaluja, kun taas viitekehykset kuvaavat näitä osaamisalueita pääosin yleisellä tasolla ilman viittauksia yksittäisiin sovelluksiin, alustoihin tai kirjastoihin. Tämä ero on todennäköisesti viitekehysten näkökulmasta kuitenkin tarkoituksenmukainen: niiden tavoitteena on esittää ohjelmistokehityksen osaaminen kestäväällä ja ajattomalla tavalla, joka ei ole sidoksissa nopeasti muuttuvaan teknologiaympäristöön.

5.1.3 Tulosten vertailu: Aiemmat tutkimukset

Vertaamme tässä luvussa tutkimuksemme tuloksia tietoperustamme kahden tieteellisen artikkelin tuloksiin. Tutkimusartikkelit on esitelty tarkemmin luvussa 2.7. Lethbridgen (1999) tutkimuksessa käsiteltiin 75 osaamisaihetta, ja Ala-Mutka & Puhakka (2009) olivat puolestaan jakaneet osaamisen 72 aiheeseen. Omassa tutkimuksessamme muodostimme aineistosta 39 aihetta.

Tutkimusasetelmien erojen vuoksi suoraa vertailua on mahdotonta tehdä, mutta vertailemme tutkimuksissa esiintyneitä aiheita siten, että vertailemme aiempien tutkimusten tärkeimmiksi tai hyödyllisimmiksi arvioituja aiheita omassa tutkimuksessamme yleisimmin esiin nousseisiin aiheisiin. Tarkastelemme, missä määrin aiheet ovat yhteneväisiä, ja mitkä ovat olennaisimmat erot. Otimme vertailuun meidän tutkimuksemme 25 useimmin esiintynyttä, sekä aiemmista tutkimuksista Lethbridgeltä 25 tärkeimmäksi arvioitua, ja Ala-Mutka & Puhakalta 25 työssä hyödyllisimmäksi arvioidua aihetta. Nämä on esitelty taulukoissa 8–10.

Taulukko 8. Lethbridge, 25 tärkeintä taitoa (mukaillen Lethbridge 1999, 33)

	Aihe	Alkup. englanninkielinen aihe	Tärkeys
1	Tietyt ohjelmointikieliset	Specific Programming Languages	3,8
2	Tietorakenteet	Data Structures	3,6
3	Ohjelmistosuunnittelu ja mallit	Software Design and Patterns	3,5
4	Ohjelmistoarkkitehtuuri	Software Architecture	3,4
5	Vaatimusmäärittely	Requirements Gathering & Analysis	3,4
6	HCI ja käyttöliittymät	HCI / User Interfaces	3,3
7	Olio-ohjelmointiin liittyvät aiheet	Object Oriented Concepts & Tech.	3,3
8	Eettisyys ja ammatillisuus	Ethics and Professionalism	3,3
9	Analyysi- ja suunnittelumenetelmät	Analysis and Design Methods	3,3
10	Presentaatiotaidot	Giving Presentations to an Audience	3,3
11	Projektinhallinta	Project Management	3,3
12	Testaus, verifiointi ja laadunvarmistus	Testing, Verif. & Quality Assurance	3,2
13	Algoritmien suunnittelu	Design of Algorithms	3,2
14	Tekninen kirjoittaminen	Technical Writing	3,1
15	Käyttöjärjestelmät	Operating Systems	3,1
16	Tietokannat	Databases	3,1
17	Johtajuus	Leadership	3,0
18	Konfiguraatio- ja julkaisuhallinta	Configuration & Release Management	3,0
19	Tiedonsiirto ja verkot	Data Transmission and Networks	3,0
20	Johtaminen	Management	2,9

	Aihe	Alkup. englanninkielinen aihe	Tärkeys
21	Tiedostonhallinta	File Management	2,8
22	Luotettavuus ja viankestävyys	Software Reliability & Fault Tolerance	2,8
23	Järjestelmäohjelmointi	Systems Programming	2,8
24	Verkkoarkkitehtuuri ja tiedonsiirto	Network Architecture & Data Trans.	2,8
25	Neuvottelutaidot	Negotiation	2,8

Taulukko 9. Ala-Mutka & Puhakka, 25 työelämässä hyödyllisintä taitoa (mukaillen Ala-Mutka & Puhakka 2009, 3)

	Aihe	Alkup. englanninkielinen aihe	Hyödyll.
1	Ohjelmoinnin perustaidot	Basic Programming Skills	4,1
2	Tietty ohjelmointikieli	A Programming Language	4,0
3	Tietokannat	Databases	3,9
4	Vaatimusmäärittely	Requirements Gathering and Analysis	3,9
5	Dokumentointi	Documentation	3,9
6	Testaus, verifiointi ja validointi	Testing, Verification and Validation	3,7
7	Ylläpito ja uudelleensuunnittelu	Softw. Maintenance & Reengineering	3,7
8	Vieraat kielet	Foreign Languages	3,7
9	Presentaatio- ja neuvottelutaidot	Giving Presentations and Negotiation	3,6
10	Tietorakenteet	Data Structures	3,5
11	Projektinhallinta	Project Management	3,5
12	Käytettävyys ja käyttöliittymät	Usability and User Interfaces	3,4
13	Ohjelmistoprosessit	Software Processes	3,3
14	Tekninen kirjoittaminen	Technical Writing	3,3
15	Käyttöjärjestelmät	Operating Systems	3,2
16	Määrittely- ja suunnittelumenetelmät	Specification and Design Methods	3,2
17	Internet ja WWW	Internet and World Wide Web	3,2
18	Olio-ohjelmointi	Object-Oriented Programming	3,1
19	Luotettavuus ja viankestävyys	Reliability and Fault Tolerance	3,1
20	Ohjelmistoarkkitehtuuri	Software Architecture	3,0
21	Tietoturva ja kryptografia	Security and Cryptography	3,0
22	Algoritmien suunnittelu	Algorithm Design	2,9
23	Ohjelmistosuunnittelu ja mallit	Software Design and Patterns	2,9
24	Konfiguraatio- ja julkaisuhallinta	Config. & Release Management	2,9

	Aihe	Alkup. englanninkielinen aihe	Hyödyll.
25	Ohjelmistojen kustannusarviointi	Software Cost Estimation	2,9

Taulukko 10. Tutkimuksessaamme 25 eniten mainintoja saanutta aihealuetta

	Aihe	Kpl
1	Ohjelmointikielet	147
2	Toistuvat ohjelmistokehityksen tehtävät	142
3	Pehmeät taidot	142
4	Projektinhallinta- ja kommunikaatiotyökalut	142
5	Projektinhallintamenetelmät	130
6	Versionhallinta	124
7	Testaustyyppi	120
8	Tietoliikenne ja rajapinnat	116
9	IDE ja siihen liittyvät	110
10	Tiedonhaku	108
11	Front end -kirjastot	104
12	Front end -teknologiat	94
13	Tietokannat	93
14	Viitekehykset	83
15	UI/UX-suunnittelu ja siihen liittyvät	83
16	Testaustyökalut ja -teknologiat	74
17	Pilvipalvelut	73
18	Muut teknologiset osaamiset	72
19	CI/CD	69
20	Tietoturva	67
21	Muut tukevat työkalut	55
22	Käyttöjärjestelmät	53
23	Analytiikka ja monitorointi	52
24	Konttiteknologiat ja orkestrointi	48
25	Back end -teknologiat	46

Useat aihealueet olivat esillä ja keskeisiä kaikissa kolmessa tutkimuksessa. Ohjelmointikielet tai ohjelmoinnin perustaidot olivat ensimmäisellä sijalla kaikissa tutkimuksissa. Tietokannat ja tietokantaosaaminen näyttäytyivät tärkeinä kaikissa, joskin Ala-Mutkalla ja Puhakalla ne sijoituivat listalla huomattavasti korkeammalle kuin muissa tutkimuksissa. Myös testausosaamisen tärkeys korostui kaikissa tutkimuksissa. Projektinhallinnan merkitys näkyi myös kaikissa kolmessa tutkimuksessa. Käyttöliittymäsuunnittelu ja käytettävyys tulivat myös esiin kaikkien tutkimusten top 25 -aiheiden listoissa. Ohjelmointitaitojen, tietokantaosaamisen, testauksen ja projektinhallinnan taitojen sekä käyttöliittymien suunnittelun kaltaisten perustaitojen voi ajatella olevan ohjelmistokehityksen ydintaitoja, jotka ovat säilyneet tärkeinä jo 1990-luvulta saakka, ja jotka todennäköisesti säilyvät tärkeinä jatkossakin. Näiden taitojen merkitys näkyi selvästi sekä Lethbridgen että Ala-Mutkan ja Puhakan tutkimuksissa ja toistui myös meidän aineistossamme.

Jotkut aiheet esiintyivät myös kaikissa kolmessa tutkimuksessa, mutta hieman eri tavoin nimettyinä tai luokiteltuina. Dokumentointi esiintyi kaikissa kolmessa tutkimuksessa (Lethbridgellä dokumentointi sijoittui otsikon Technical Writing alle, meillä toistuviin ohjelmistokehityksen tehtäviin). Konfiguraatio- ja julkaisuhallinta oli tämännimisenä mukana aiemmissa tutkimuksissa, mutta meidän aineistostamme muodostui aiheeseen liittyen kategoriat Versionhallinta ja CI/CD. Tuloksissamme esiintyvä Tietoliikenne ja rajapinnat -aihe voidaan nähdä osittain vastaavana Ala-Mutka & Puhakan Internet and World Wide Web -aiheen kanssa. Lethbridgellä tähän linkittyviä aiheita olivat Tiedonsiirto ja verkot sekä Verkkoarkkitehtuuri ja tiedonsiirto.

Olennaisimpia aiheita, jotka löytyivät aiemmista tutkimuksista keskeisimpien aiheiden joukosta, mutta eivät nousseet meidän tutkimuksessamme esille, olivat matemaattinen ja teoreettinen osaaaminen, liiketoiminta, ja vaatimusmäärittely. Esimerkiksi tietorakenteet ja olio-ohjelmointi teoreettisina käsitteinä sekä algoritmit puuttuivat meidän tuloksistamme kokonaan. Teoreettisten käsitteiden puute tuloksissamme johtuu todennäköisesti osin analyysimenetelmästäämme, jossa keskityimme poimimaan aineistosta mainintoja nimenomaan kirjoittajien käyttämistä työkaluista, teknologioista ja taidoista. Tästä syystä meidän tutkimuksemme tuotti runsaasti nimettyjä mainintoja tietystä teknologioista, kun taas aiemmissa tutkimuksissa laajemmat aiheet korostuivat enemmän. Tulosten käytäntöön painottumista lisäsi todennäköisesti osaltaan myös se, että aineisto koostuu ammattikorkeakouluopiskelijoiden opinnäytetöistä, ja ammattikorkeakouluissa opiskelu painottuu käytäntöön.

Tulosten painottuminen meidän tutkimuksessamme käytännön taitoihin johtuu todennäköisesti osaksi myös siitä, että opinnäytetyön kirjoittajat ovat raportoineet konkreettisista työtehtävistään, mutta eivät ole kaikkien tehtävien yhteydessä pohtineet syvällisemmin niihin liittyviä, taustalla olevia teoreettisempia käsitteitä. Kirjoittajat ovat valinneet syvällisempään reflektointiin vain rajatun

määrän teemoja työn alkuvaiheessa määrittelemiensä tavoitteiden mukaisesti. Jos kirjoittajilta olisi kysytty, oliko heidän työssään eteen tullessiin käytännön tehtäviin liittyvä teoreettisempi osaamisen tausta tärkeä, vastaus olisi saattanut olla kyllä, mutta analyysimenetelmällämme tämä tieto ei noussut esiin aineistosta.

Erot meidän tuloksissamme aiempiin tutkimuksiin verrattuna saattavat kertoa osaksi myös siitä, että ohjelmistokehitys on muuttunut käytännönläheisempään suuntaan. Tutkimuskirjallisuudessa on esitetty, että teoreettiset aiheet kuten matematiikka ja insinööritieteet, nähdään nykyään vähemmän keskeisinä ohjelmistokehittäjän työn kannalta (Garousi, Giray, Tüzün, Catal & Felderer 2019, 13–14). Lisäksi tutkimuksemme aineiston valikoituneisuus ja painottuminen uran alkuvaiheessa oleviin kehittäjiin vaikuttaa todennäköisesti siihen, että tietyt aiheet, kuten liiketoimintaosaaminen ja johtaminen, puuttuivat kokonaan meiltä mutta esiintyivät aiemmissä tutkimuksissa. Liiketoimintaosaaminen ja johtamistehtävät saattavat tulla mukaan vasta urallaan pidemmälle ehtineiden ohjelmistokehittäjien työnkuviin.

Yhteenvedona voidaan siis mainita, että erona aiempiin tutkimuksiin aineistossamme korostui käytännön tekeminen enemmän kuin teoreettiset lähtökohdat. Lisäksi käytettyjen työkalujen ja teknologioiden tarkempi nimeäminen oli tutkimusasetelmamme takia tuloksissamme tyypillistä, kun aiemmissä tutkimuksissa keskityttiin enemmän osaamisen yleisiin alueisiin. Joissain aiheissa oli nähtävissä myös työtapojen ja teknologioiden ajallista trendiä ja siirtymää, joka voi kuvastaa ohjelmistokehityksen suuntaa yleisesti. Ohjelmistokehityksen ydinosaamiset näyttäytyivät meidän tuloksissamme kuitenkin pääpiirteittäin yhtenevinä aikaisempien tutkimuksien kanssa.

5.1.4 Tulosten vertailu: Aiemmat opinnäytetyöt ja kyselyt

Tässä luvussa vertailemme tutkimuksemme tuloksia Teerimäen ja Kallioston opinnäytetöihin sekä Anu Nivan artikkeliin, joka perustuu Oulun ammattikorkeakoulussa tehtyyn sidosryhmäkyselyyn. Näihin vertailu on kiinnostavaa, sillä tutkimuksissa on käsitelty nimettyjä yksittäisiä teknologioita kuten meidänkin tutkimuksessamme, mikä tekee vertailusta selkeämpää. Tutkimukset ovat hieman erilaisista näkökulmista toteutettuja, mutta ne kaikki käsittelevät ohjelmistokehityksen osaamistarpeita työelämässä.

Aiemmissa opinnäytetöissä, Oulun sidosryhmäkyselyssä ja meidän tuloksissamme kaikissa ohjelmointi näyttäytyi tärkeänä taitona. Ohjelmointikielten painotukset vaihtelivat hieman: Teerimäellä korostuivat SQL, Java ja PHP (Teerimäki 2012, 27), Kalliostolla Java, JavaScript ja C# (Kalliosto 2018, 18), ja Nivalla Python, JavaScript ja Java (Niva 2024). Meidän tutkimuksessamme kolme mainituinta ohjelmointikieltä olivat JavaScript, TypeScript ja Java. Java oli siis ainoa kieli, joka oli kaikissa tutkimuksissa kaikkein tärkeimpien tai yleisimpien ohjelmointikielten joukossa.

Myös tietokannat oli mainittu kaikissa tutkimuksissa tärkeiden osaamisten joukossa. Teerimäellä tietokantoihin liittyen korostui erityisesti SQL-kielen osaaminen (Teerimäki 2012, 27), ja Kalliosto mainitsi tärkeimpinä tietokantateknologioina erityisesti PostgreSQL:n ja MySQL:n (Kalliosto 2018, 19). Myös Niva mainitsee tietokannat sidosryhmäkyselyssä useita mainintoja saaneena osaamisalueena (Niva 2024), ja myös meillä tietokannat ja tietokantateknologiat näyttäytyivät keskeisenä aiheena.

Kallioston työ nosti esiin Dockerin (Kalliosto 2018, 19) ja uusien teknologioiden osiossa myös pilvipalvelut (Kalliosto 2018, s. 23). Konttitekniikat ja pilvipalvelut näkyivät myös omassa aineistossamme ja Nivan artikkelissa (Niva 2024). Teerimäen tutkimus on hiukan vanhempi, mikä selittänee sitä, että nämä aiheet eivät tulleet hänen tutkimuksessaan samalla tavalla esiin. Samoin Teerimäen tutkimuksesta puuttuu mutta myöhemmistä löytyy tärkeiden osaamistarpeiden joukosta ketterä kehitys (Kalliosto 2018, 21) tai ketterät menetelmät (Niva 2024), mikä kertoo osaltaan työtapojen nopeasta kehityksestä IT-alalla.

Pehmeistä taidoista Teerimäki nosti esiin viestintä- ja neuvottelutaidot, ajanhallinnan ja presentaatitaidot (Teerimäki 2012, 28). Kalliostolla pehmeistä taidoista tärkeimpinä näyttäytyivät asiakaspalvelutaidot ja myös esimerkiksi englannin taito ja sosiaaliset taidot mainittiin (Kalliosto 2018, 20). Nivan aineistossa puolestaan nousee esiin paljon pehmeitä taitoja kuten viestintä, tiimityö, asiakaslähtöisyys, muutoskyky ja itseohjautuvuus (Niva 2024). Meidän tutkimuksemme tulokset ovat linjassa erityisesti Nivan kanssa, sillä pehmeät taidot kuten tiimityöskentely, itseohjautuvuus ja viestintä esiintyvät aineistossamme useasti.

Versionhallinta mainittiin meidän aineistossamme varsin usein: se mainittiin yhteensä 124 opinnäytetyössä (N = 154). Tämä oli mielenkiintoinen ero aiempiin opinnäytetöihin, joissa sen merkitys ei korostunut samalla tavalla. Kalliosto tosin nosti aiheen erikseen esille ja piti yllättävänä sitä, että hänen kyselynsä vastaajat eivät yleensä olleet maininneet versiohallintaa juuri lainkaan. Hän pohti, että versionhallinta saattaa olla niin rutiininomainen osa ohjelmistokehitystä, että sitä ei ole tullut vastaajille edes mieleen mainita. (Kalliosto 2018, 30.) Päiväkirjamuotoisten opinnäytetöiden analyysissä nousivat ehkä selkeämmin esiin myös tällaiset päivittäisiin rutiineihin liittyvät asiat, jotka eivät ehkä kyselyiden tai haastattelujen vastauksissa esiinny. Tähän viittaa myös se, että meidän aineistossamme kirjoittajat mainitsivat nimeltä muun muassa käyttämiänsä pikaviestimiä sekä erilaisia projektin- ja tehtävähallinnan työkaluja. Versionhallinnan ja työkalujen osaamisen tärkeys näkyi kuitenkin Oulun ammattikorkeakoulun sidosryhmäkyselyssä, missä ne oli nostettu esille teknisinä osaamistarpeina (Niva 2024).

5.2 Tulokset ja ajankohtaiset trendit

Teimme myös joitakin kiinnostavia havaintoja ohjelmistokehityksen teknologioiden ja työtapojen muutoksesta ajan myötä. Meidän tutkimuksessamme nousi esiin useita DevOps-aiheisia uusia osaamisalueita, joita ei aiemmissa tutkimuksissa vielä tunnistettu. DevOps-menetelmään liittyvät pilvipalvelut, konttitekniologiat, CI/CD-käytännöt eli jatkuva integraatio ja jatkuva toimitus, sekä analytiikka ja monitorointi (Jabbari ym. 2016). Lucy Ellen Lwakatare on pohtinut väitöskirjassaan (2017) DevOpsia ja sen käyttöä ohjelmistokehityksessä. Lwakataren mukaan termi DevOps sai alkunsa vuonna 2009, jonka jälkeen sen käyttö ohjelmistokehityksessä on yleistynyt. Vuonna 2013 siitä julkaistiin ensimmäinen tilanneraportti ja vuosi sen jälkeen ensimmäinen tieteellinen katsaus. (Lwakatare 2017, 16–17, 25, 31.) Tämä viittaa siihen, että DevOps-käytännöt eivät olleet juurikaan käytössä vielä Lethbridgen tutkimuksen aikaan 1990-luvulla, eivätkä ne vielä Ala-Mutkan ja Puhakan (2009) tutkimuksenkaan aikaan olleet ohjelmistokehityksen valtavirtaa.

Tietoliikenteeseen liittyvissä aiheissa on myös nähtävissä mielenkiintoinen ero aiempiin tutkimuksiin verrattuna. Meidän tutkimuksessamme rajapinnat (API) nousivat esiin laajasti mainittuna aiheena. API-rajapinnat ovat keskeisessä asemassa nykyaikaisessa ohjelmistokehityksessä, erityisesti mikropalveluarkkitehtuurien kontekstissa (Lercher, Glock, Macho & Pinzger 2024, 1). Rajapintoja ei ollut omana aiheenaan mukana aiemmissa tutkimuksissa, mikä viittaa siihen, että aihe on mahdollisesti noussut esiin keskeisempänä vasta myöhemmin.

Ehkä suurin yllätys meille oli Pythonin mainintojen suhteellisen pieni määrä aineistossa. Tähän voi vaikuttaa aineistomme, joka pohjautuu Haaga-Helia ammattikorkeakoulun tietojenkäsittelyn opiskelijoiden vuosina 2020–2024 julkaistuihin opinnäytetöihin, ja kyseisinä vuosina Haaga-Helian tietojenkäsittelyn ohjelmistokehityksen kurssit painottuivat vahvasti Javaan ja JavaScriptiin (Haaga-Helia ammattikorkeakoulu s.a.). On mahdollista, että Python olisi noussut vahvemmin esiin, jos mukana olisi ollut myös data-analyttikoiden, tekoälykehittäjien tai automaatiotehtävissä työskentelevien henkilöiden opinnäytetyöt ajanjaksolta, koska tämän tyyppisissä rooleissa Pythonin käyttö on yleistä (Raschka, Patterson & Nolet 2020, 2–3).

Tekoälyyn ja erityisesti suuriin kielimalleihin (Large Language Models, LLM) liittyvät maininnat alkoivat erottua aineistossamme vasta vuoden 2023 opinnäytetöistä alkaen ja johdonmukaisemmin vuoden 2024 opinnäytetöissä. Siitä huolimatta mainintojen kokonaismäärä jäi yllättävän vähäiseksi. Tämä on kiinnostavaa suhteessa viimeaikaiseen uutisointiin (Helakallio 18.11.2024) sekä muun muassa Houn ja kumppaneiden (2024) tutkimukseen, jossa LLM-tekniologian kasvava merkitys ohjelmistokehityksessä on ollut vahvasti esillä (Hou ym. 2024, 2–4). Tutkimuksessa toteutettiin

395 tutkimusartikkelin kirjallisuuskatsaus vuosilta 2017–2024, jonka tavoitteena oli muodostaa kokonaisvaltainen ymmärrys LLM-tekniikan soveltamisesta ohjelmistokehityksessä (Hou ym. 2024, 1).

Tulosten mukaan kielimalleja on hyödynnetty erityisesti ohjelmakoodin generoinnissa ja korjaamisessa, vaatimustenmäärittelyn tukena (Hou ym. 2024, 27) sekä ohjelmistotestauksen automatisoinnissa (Hou ym. 2024, 35). Tekniikan on nähty madaltavan kynnystä uusien työkalujen omaksumiseen ja tukevan erityisesti aloittelevia kehittäjiä. Katsauksessa tuotiin kuitenkin esiin myös haasteita, kuten mallien epäluotettavuutta, virhealttiutta, vaikeuksia käsitellä monimutkaisia rakenteita ja selittää tuotettuja ratkaisuja (Hou ym. 2024, 42–43).

Kielimallien ja tekoälyn vähäinen näkyvyys opinnäytetöissä voi viitata siihen, että niiden käyttöönotto työelämässä on edelleen alkuvaiheessa, eikä niitä välttämättä vielä hyödynnetä uransa alkuvaiheessa olevien kehittäjien arjessa. Toinen mahdollinen selitys on, ettei tekoälypohjaisia työkaluja vielä koeta olennaiseksi osaksi omaa asiantuntijuutta, tai niiden käyttö jää mainitsematta, mikäli se on satunnaista tai rooli jää vähäiseksi. Myös tekniikan käyttöön liittyvä epävarmuus ja kriittisyys voivat osaltaan vähentää sen näkyvyyttä opinnäytetöissä.

5.3 Tutkimuksen luotettavuuden varmistaminen

Laadullisen tutkimuksen luotettavuuden kannalta on olennaista kuvata käytetyt menetelmät mahdollisimman tarkasti ja avata, miten aineiston pohjalta on tehty tulkintoja (Aaltio & Puusa 2020, 188). On tärkeää, että lukija ymmärtää, millaisen prosessin myötä tutkija on päätenyt juuri niihin johtopäätöksiin, jotka hän tutkimuksessaan esittää (Aaltio & Puusa 2020, 178; Hirsjärvi ym. 2016, 232–233).

Reliabiliteetti on keskeinen käsite tieteellisen tutkimuksen luotettavuuden arvioinnissa. Reliabiliteetilla tarkoitetaan sitä, että jos tutkimus toistettaisiin samalla tavalla, pitäisi tulostenkin olla samat. (Tuomi & Sarajärvi 2018, luku 6.2.) Laadullista tutkimusta ei kuitenkaan koskaan voi toistaa täysin samanlaisena, sillä vaikka tutkimusprosessi olisi kuvattu yksityiskohtaisesti, eri tutkijat voisivat yksilöllisistä lähtökohdistaan silti päätyä hieman erilaisiin tuloksiin (Vilka 2015, 197). Olemme tässä opinnäytetyössämme pyrkineet kuvaamaan tutkimusprosessimme niin, että se avautuu lukijalle mahdollisimman hyvin. Olemme pyrkineet systemaattiseen ja yhdenmukaiseen aineiston läpikäyntiin ja dokumentointiin tutkimuksen kulun, jotta lukijalle on selkeää, miten olemme toteuttaneet tutkimuksemme. Aineiston analyysi tehtiin induktiivisen sisällönanalyysin keinoin, jolloin teemat nousivat suoraan aineistosta. Saimme lisäksi tuotettua tutkimukseen triangulaatiota toteuttamalla analyysin kahden tutkijan yhteistyönä, mikä lisäsi tutkimuksen luotettavuutta.

Tiedostamme myös tutkimuksemme puutteet, ja pyrimme käsittelemään niitä avoimesti. Tärkeimmät havaintomme mahdollisista vinoumista liittyvät toisaalta aineiston valikoituneisuuteen, ja toisaalta opinnäytetyöhön tekstilajina ja tutkimuksen kohteena. Tarkastelemme seuraavaksi molempia ja pohdimme, miten ne saattavat vaikuttaa tutkimuksemme tuloksiin ja luotettavuuteen.

Tärkeä huomio on aineiston rajausta ja sen vaikutukset tulosten yleistettävyyteen. Tutkimuksen validiteetilla tarkoitetaan sitä, tuottaako tutkimus tietoa juuri siitä asiasta, mitä halutaan tutkia (Hirsjärvi ym. 2016, 231). Meidän tutkimuksemme kannalta keskeinen on erityisesti ulkoisen validiuden käsite, joka viittaa siihen, voiko tutkimustulosten perusteella tehdä yleistyksiä, jotka kertovat myös muista kuin suoraan tutkimuksen kohteina olevista tapauksista (Aaltio & Pusa 2020, 179).

Aineistomme ei muodosta kattavaa otosta ohjelmistokehityksen rooleista, koska se ei ole satunnaisesti valikoitu, vaan koostuu ainoastaan Haaga-Helia ammattikorkeakoulun ohjelmistokehityksen päiväkirjamuotoisista opinnäytetöistä. Kirjoittajat ovat todennäköisesti vasta uransa alussa, mikä vaikuttaa siihen, millaisia asioita heidän työtehtäviinsä sisältyy ja mistä näkökulmasta he raportoivat tehtävistään. Lisäksi opiskelijoiden päätyminen juuri tiettyihin yrityksiin ja edelleen tiettyihin tehtäviin näissä yrityksissä ei ole sattumanvaraista, ja käytössä olleet teknologiat saattavat poiketa huomattavasti siitä, että tarkasteltaisiin tasapuolisesti kaikkia suomalaisia ohjelmistotalan yrityksiä. On myös mahdollista, että opiskelijat eivät ole roolissaan käyttäneet kaikkia organisaatiossaan käytössä olevia teknologioita, eikä niitä ole näin ollen mainittu opinnäytetyöraporteissa. Erityisesti Haaga-Helian opetuksen painotukset vaikuttavat todennäköisesti opiskelijoiden työllistymiseen ja roolien painottamiseen. Haaga-Heliassa tarjottavat ohjelmoinnin peruskurssit keskittyvät tällä hetkellä pääasiassa Javaan, JavaScriptiin ja Pythoniin (Haaga-Helia ammattikorkeakoulu s.a.), joten on todennäköistä, että opiskelijat työllistyvät rooleihin, joissa käytetään juuri näitä ohjelmointikieliä. On tärkeää tiedostaa, että tämä rajausta heikentää tutkimuksemme ulkoista validiteettia ja vaikuttaa näin tutkimustulosten yleistettävyyteen.

Lisäksi opinnäytetyö on oma erityinen tekstilajinsa, mikä vaikuttaa siihen, millaisia asioita induktiivinen sisällönanalyysi voi nostaa esiin aineistosta. Opinnäytetyö tekstilajina ohjaa kirjoittajan valintoja paitsi yksittäisessä viestintätilanteessa, myös laajemmassa sosiokulttuurisessa kontekstissa (Vuorijärvi 2013, 213). Päiväkirjamuotoisissa opinnäytetöissä kirjoittamista ohjaa opinnäytetyön ohjeistus ja tietyt opinnäytetyökontekstiin liittyvät odotukset, jotka vaikuttavat siihen, mitä teksteissä korostetaan. Haaga-Helian opinnäytetyöohjeistus on ohjannut opinnäytetyön kirjoittajaa tarkastelemaan työtään opinnäytetyöhön valitun tietoperustan näkökulmasta, jolloin kulloinkin tietoperustassa käytetty kirjallisuus ohjaa kirjoittajaa nostamaan tiettyjä asioita esiin työstään. Kirjoittajia on myös ohjeistuksessa erikseen pyydetty raportoimaan tietyistä teemoista, esimerkiksi viikoittaisista

vuorovaikutustilanteista sekä niiden vaikutuksesta omaan osaamiseen. Tämä johtaa todennäköisesti siihen, että ohjeistuksessa erikseen mainittuihin teemoihin liittyvät osaamistarpeet saavat suhteessa suuremman painoarvon kuin työtehtävien todellinen sisältö edellyttäisi.

Päiväkirjamuotoisten opinnäytetöiden käyttö aineistona voidaan kuitenkin nähdä myös tutkimuksen vahvuutena. Toisin kuin yksittäinen haastattelu, päiväkirjat sisältävät pidemmältä ajanjaksolta kerättyä ja konkreettista dokumentaatiota tehdyistä työtehtävistä, ja tehtäviin palataan opinnäytetyössä usein uudelleen viikkoanalyysien muodossa. Tutkimustuloksia voidaan hyödyntää ohjelmistokehityksen osaamistarpeiden tarkastelussa, kun otetaan huomioon aineistoon ja menetelmään liittyvät rajoitteet.

5.4 Jatkotutkimus

Päiväkirjamuotoisissa opinnäytetöissä on runsaasti potentiaalia jatkotutkimukselle. Ne muodostavat rikkaan ja autenttisen aineiston, jonka laadullinen analyysi voisi tuottaa tietoa esimerkiksi työelämän ilmiöistä, opiskelijoiden kokemuksista ja ammatillisesta kehitymisestä. Ohjelmistokehityksen opinnäytetöitä voisi edelleen hyödyntää ohjelmistokehityksen osaamistarpeiden tutkimiseen: esimerkiksi ottamalla mukaan opinnäytetöitä pidemmältä ajalta ja keskittymällä ajallisiin muutoksiin voisi löytyä kiinnostavaa tietoa osaamistarpeiden trendeistä. Kirjoittajien ilmauksia eri tutkimusmenetelmillä analysoimalla voisi olla mahdollista tavoittaa syvällisempää tietoa kirjoittajien kokemuksista johonkin tiettyyn osaamiseen liittyen. Myös muiden kuin ohjelmistokehityksen suuntauksen opinnäytetyöt voisivat tarjota tietoa esimerkiksi ICT-infrastruktuurin tai digitaalisten palveluiden aihepiireistä. Haaga-Heliassa on myös muiden alojen kuin tietojenkäsittelyn opiskelijoita, ja esimerkiksi liiketalouden alan suuntautumisten päiväkirjamuotoisista opinnäytetöistä voisi löytyä kiinnostavaa tutkittavaa.

Olisi myös erittäin mielenkiintoista toteuttaa tämän tutkimuksen jälkeen ohjelmistokehityksen osaamistarpeita kartoittava kysely- tai haastattelututkimus. Esimerkiksi Haaga-Helian alumneille tai laajemmin alalla työskenteleville suunnattu kysely osaamistarpeista tarjoaisi ajankohtaista ja vertailukelpoista tietoa siitä, mitä ajankohtaisia osaamistarpeita alalla on. IT-ala muuttuu nopeasti ja useat aikaisemmat kyselyt on toteutettu jo useampia vuosia sitten, joten tällaisen kyselyn toistaminen olisi mielestämme jälleen ajankohtaista. Kyselyn suunnittelussa voisi mahdollisesti hyödyntää tämän tutkimuksen tuloksia ja havaintoja.

Lähteet

Aaltio, I. & Puusa, A. 2020. Mitä laadullisen tutkimuksen arvioinnissa tulisi ottaa huomioon? Teoksessa Puusa, A. & Juuti, P. (toim.). Laadullisen tutkimuksen näkökulmat ja menetelmät. Gaudeamus. Tallinna.

Ala-Mutka, K. & Puhakka, A. 2009. Survey on the Knowledge and Education Needs of Software Professionals. Tampereen teknillinen yliopisto, ohjelmistotekniikan laitos. Luettavissa: <https://citereerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=f0a4e83dab64e3fdc9310ac905435ddf19bc61a5>. Luettu: 24.11.2024.

Association for Computing Machinery & IEEE Computer Society 2017. Information Technology Curricula 2017: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology. Computing Curricula Series. Association for Computing Machinery & IEEE Computer Society. New York. Luettavissa: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf>. Luettu: 29.3.2025.

Association for Computing Machinery & IEEE Computer Society 2020. Computing Curricula 2020: Paradigms for Global Computing Education. A Computing Curricula Series Report. Association for Computing Machinery & IEEE Computer Society. New York. Luettavissa: <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/cc2020.pdf>. Luettu: 29.3.2025.

Assyne, N., Ghanbari, H. & Pulkkinen, M. 2022. The state of research on software engineering competencies: A systematic mapping study. Journal of Systems and Software, 185, 111183, s. 1–18.

Balcar, J., Blažičková, J., Braňka, J., Czesaná, V., Gavenda, M., Grygerová, Š., Havlena, J., Havlíčková, V., Hladká, M., Homolová, E., Janoš, J., Karásek, Z., Karásková, J., Kavan, L., Kotíková, J., Larsen, C., Micková, E., Rand, S., Schmid, A., Soukup, T., Suciú, D., Trokanová, K., Vondráková, M., Vychová, H. & Zacharová, L. 2011. Transferability of Skills across Economic Sectors: Role and Importance for Employment at European Level. Publications Office of the European Union. Luxembourg. Luettavissa: <https://data.europa.eu/doi/10.2767/40404>. Luettu: 19.2.2025.

Breyer, J., Defina, R., Hook, T., Lau, F., Scquizzato, R. & Thornley, C. 2018. European ICT professional role profiles – Part 1: 30 ICT profiles. CWA 16458-1:2018. CEN – European Committee for Standardization. Bryssel. Luettavissa: https://www.cencenelec.eu/media/CEN-CENELEC/AreasOfWork/CEN%20sectors/Digital%20Society/CWA%20Download%20Area/ICT_SkillsWS/16458-1.pdf. Luettu: 5.4.2025.

Cass, S. 2024. Top Programming Languages 2024. Luettavissa: <https://spectrum.ieee.org/top-programming-languages-2024>. Luettu: 3.5.2025.

Cedefop 2010. Terminology of European education and training policy. Glossary. Luettavissa: <https://www.cedefop.europa.eu/en/tools/vet-glossary/glossary>. Luettu: 19.4.2025.

Ek, J. 2020. Ohjelmistoalan toimialaraportti 2020. Työ- ja elinkeinoministeriön julkaisuja 2020:6. Työ- ja elinkeinoministeriö. Helsinki. Luettavissa: <http://urn.fi/URN:ISBN:978-952-327-493-8>. Luettu: 8.1.2025.

Garbajosa, J., Yagüe, A. & González, E. 2014. Communication in Agile Global Software Development: An Exploratory Study. On the Move to Meaningful Internet Systems: OTM 2014 Workshops, Amantea, s. 408–417.

Garousi, V., Giray, G., Tüzün, E., Catal, C. & Felderer, M. 2019. Aligning software engineering education with industrial needs: A meta-analysis. Journal of Systems and Software, Volume 156.

Haaga-Helia ammattikorkeakoulu s.a. TRATI22 Tietojenkäsittelyn koulutus, päivä ja monimuoto, Pasila. Opinto-opas. Luettavissa: <https://opinto-opas.haaga-helia.fi/fi/131320/fi/131354/TRATI22/801/year/2024>. Luettu: 30.4.2025.

Hanhinen, T. 2010. Työelämäosaaminen: Kvalifikaatioiden luokitusjärjestelmän konstruointi. Väitöskirja. Tampereen yliopisto, kasvatustieteellinen tiedekunta. Luettavissa: <https://urn.fi/urn:isbn:978-951-44-8290-8>. Luettu: 22.11.2024.

Helakallio, A. 18.11.2024. Tekoäly nostaa Tietoevryn tehokkuutta: vaikutus +23 %. Tivi. Luettavissa: <https://www.tivi.fi/uutiset/tekoaly-nostaa-tietoevryn-tehokkuutta-23-prosentin-kasvu/dfc9b31a-4951-4a8d-b9b5-692efae55d40>. Luettu: 3.5.2025.

Hirsjärvi, S., Remes, P. & Sajavaara, P. 2016. Tutki ja kirjoita. 21. painos. Kustannusosakeyhtiö Tammi. Porvoo.

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J. & Wang, H. 2024. Large Language Models for Software Engineering: A Systematic Literature Review. ACM Transactions on Software Engineering and Methodology, 33, 8, s. 1–79.

Humalamäki, A. 2025. Helsingin yliopisto avaa yrityksille uuden tutkimusyhteistyön kiihdyttämön – ”Tiede voi auttaa liiketoiminnan kehittämisessä”. Luettavissa: <https://www.helsinki.fi/fi/uutiset/innovaatiot/helsingin-yliopisto-avaa-yrityksille-uuden-tutkimusyhteistyon-kiihdyttamon-tiede-voi-auttaa-liiketoiminnan-kehittamisessa>. Luettu: 15.1.2025.

Hynninen, T., Hujala, M. & Knutas, A. 2022. What can we learn from recommendations of early-career engineers? Assessing computing and software engineering education using a career monitoring survey. The United Kingdom and Ireland Computing Education Research (UKICER) Conference, Dublin, s. 1–7.

IEEE Computer Society 2024. Guide to the Software Engineering Body of Knowledge V4.0. IEEE Computer Society. Washington, DC. Luettavissa: <https://www.computer.org/education/bodies-of-knowledge/software-engineering>. Luettu: 11.4.2025.

IEEE Computer Society & Association for Computing Machinery 2015. Software Engineering 2014: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. Computing Curricula Series. IEEE Computer Society & Association for Computing Machinery. Washington, DC. Luettavissa: <https://www.acm.org/binaries/content/assets/education/se2014.pdf>. Luettu: 29.3.2025.

ISO/IEC/IEEE 24765:2017. Systems and software engineering — Vocabulary. International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE). Geneve.

ISO/IEC/IEEE 29148:2018. Systems and software engineering — Life cycle processes — Requirements engineering. International Organization for Standardization (ISO), International Electrotechnical Commission (IEC), Institute of Electrical and Electronics Engineers (IEEE). Geneve.

Jabbari, R., Ali, N. b., Petersen, K. & Tanveer, B. 2016. What is DevOps? A Systematic Mapping Study on Definitions and Practices. Scientific Workshop Proceedings of XP2016, Edinburgh, s. 1–11.

Jansen, P. 2025. TIOBE Index for April 2025. Luettavissa: <https://www.tiobe.com/tiobe-index/>. Luettu: 3.5.2025.

Kalliosto, J. 2018. Ohjelmistokehittäjän työelämän osaamistarpeet. AMK-opinnäytetyö. Haaga-Helia ammattikorkeakoulu, tietojenkäsittelyn koulutusohjelma. Luettavissa: <https://urn.fi/URN:NBN:fi:amk-201805229375>. Luettu: 19.4.2025.

Laine, T. 2018. Miten kokemusta voidaan tutkia? Fenomenologinen näkökulma. Kirjassa Valli, R. (toim.) Ikkunoita tutkimusmetodeihin II. Näkökulmia aloittavalle tutkijalle tutkimuksen teoreettisiin lähtökohtiin ja analyysimenetelmiin. Uudistettu ja täydennetty painos. PS-kustannus. Jyväskylä. E-kirja. Luettu: 6.1.2025.

Lamri, J. & Lubart, T. 2023. Reconciling hard skills and soft skills in a common framework: The generic skills component approach. *Journal of Intelligence*, 11(6), 107.

Lercher, A., Glock, J., Macho, C. & Pinzger, M. 2024. Microservice API Evolution in Practice: A Study on Strategies and Challenges. *Journal of Systems and Software*, Volume 215. Luettavissa: <https://doi.org/10.1016/j.jss.2024.112110>. Luettu: 1.5.2025.

Leveälahti, S., Nyyssölä, K. 2019. Osaaminen 2035. Osaamisen ennakointifoorumin ensimmäisiä ennakointituloksia. Raportit ja selvitykset 2019:3. Opetushallitus. Helsinki. Luettavissa: https://www.oph.fi/sites/default/files/documents/osaaminen_2035.pdf. Luettu: 6.1.2025.

Lethbridge, T. 1999. The Relevance of Education to Software Practitioners: Data from the 1998 Survey. University of Ottawa Computer Science Technical Report TR99-06. University of Ottawa. Luettavissa: <http://www.site.uottawa.ca/~tcl/edrel/EdrelTechReport.pdf>. Luettu: 22.11.2024.

Lwakatare, L.E. 2017. DevOps adoption and implementation in software development practice: Concept, practices, benefits and challenges. Väitöskirja. Oulun yliopisto, Tieto- ja sähkötekniikan tiedekunta. Luettavissa: <https://urn.fi/URN:ISBN:978-952-62-1711-6>. Luettu: 4.5.2025.

Masood, Z., Hoda, R. & Blincoe, K. 2020. How agile teams make self-assignment work: a grounded theory study. *Empirical Software Engineering*, 25, s. 4962–5005.

Mell, P. & Grance, T. 2011. The NIST Definition of Cloud Computing. NIST Special Publication 800–145. National Institute of Standards and Technology. Gaithersburg. Luettavissa: <https://doi.org/10.6028/NIST.SP.800-145>. Luettu: 30.4.2025.

Microsoft s.a. The Basics. Luettavissa: <https://www.typescriptlang.org/docs/handbook/2/basic-types.html>. Luettu: 3.5.2025.

Mäntysaari, J. 2022. IT-alan osaajapula: Työmarkkinoiden kohtaanto-ongelma tietojenkäsittelyn tradenomien ja tietojärjestelmätieteen KTM-tutkinnon suorittaneiden kannalta. Pro gradu -tutkielma. Vaasan yliopisto, Tekniikan ja innovaatiojohtamisen akateeminen yksikkö, tietojärjestelmätiede. Luettavissa: <https://urn.fi/URN:NBN:fi-fe2022051635683>. Luettu: 5.3.2025.

Niiniluoto, I. 1996. Informaatio, tieto ja yhteiskunta: Filosofinen käsiteanalyysi. 5., täydennetty painos. Edita. Helsinki.

Niva, A. 2024. ICT-alan tutkinto-ohjelmasta valmistuneelta odotetaan erityisesti ohjelmointi- ja kommunikaatiotaitoja. Oulun ammattikorkeakoulun julkaisuja 115/2024. Oulun ammattikorkeakoulu. Oulu. Luettavissa: <https://urn.fi/URN:NBN:fi-fe20241213102330>. Luettu: 19.4.2025.

Nyysölä, K. 2022. Koulutus tulevaisuudessa: Ennakointinäkökulmia koulunkäyntiin, kehittämiseen ja osaamiseen. Raportit ja selvitykset 2022:1. Opetushallitus. Helsinki. Luettavissa:

https://www.oph.fi/sites/default/files/documents/Koulutus_tulevaisuudessa.pdf. Luettu: 24.11.2024.

Opetushallinnon tilastopalvelu Vipunen 2025. Opetushallinnon ja Tilastokeskuksen tietopalvelusopimuksen aineisto 2.9: Ammattikorkeakouluissa suoritettut tutkinnot. Luettavissa: <https://vipunen.fi/fi-fi/layouts/15/xlviewer.aspx?id=fi-fi/Raportit/Ammattikorkeakoulujen%20tutkinnot%20-%20amk.xlsb>. Luettu: 9.4.2025.

Opinnäytetyökoordinaattorit 2022. Ohje päiväkirjaopinnäytetyölle. Haaga-Helia ammattikorkeakoulu. Helsinki. Luettavissa: <https://www.haaga-helia.fi/sites/default/files/file/2024-01/paivakirja-opinnaytetyo-ohje.pdf>. Luettu: 22.11.2024.

Oracle, 2025. What is Java technology and why do I need it? Luettavissa:

https://www.java.com/en/download/help/whatis_java.html. Luettu: 16.4.2025.

Peslak, A. & Conforti, M. 2020. Computer programming languages in 2020: What we use, who uses them, and how they impact job satisfaction. Issues in Information Systems, Volume 21, Issue 2. Luettavissa: https://doi.org/10.48009/2_iis_2020_259-269. Luettu: 1.5.2025.

Raschka, S., Patterson, J. & Nolet, C. 2020. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and Artificial Intelligence. Information, 11, 4, artikkeli 193.

Salminen, A. 2011. Mikä kirjallisuuskatsaus? Johdatus kirjallisuuskatsauksen tyypeihin ja hallintotieteellisiin sovelluksiin. Vaasan yliopiston julkaisuja. Vaasa. Luettavissa: https://www.uwasa.fi/materiaali/pdf/isbn_978-952-476-349-3.pdf. Luettu: 10.1.2025.

Shanahan, M. 2024. Talking about Large Language Models. Communications of the ACM, 67(2), 68–79.

Soanes, C. & Stevenson, A. (toim.) 2005. The Oxford dictionary of English. 2. painos. Oxford University Press. Oxford.

Statista 2024. Revenue of the software market in Finland from 2020 to 2029 (in million U.S. dollars). Statista. Luettu: 9.4.2025.

Synergy Research Group 2025. AI Helps Cloud Market Growth Rate Jump to Almost 25 % in Q1. Luettavissa: <https://www.srgresearch.com/articles/ai-helps-cloud-market-growth-rate-jump-back-to-almost-25-in-q1>. Luettu: 3.5.2025.

Teerimäki, J. 2012. IT-tradenomikoulutuksen vastaavuus työelämän vaatimuksiin. AMK-opinnäytetyö. Haaga-Helia ammattikorkeakoulu, tietojenkäsittelyn koulutusohjelma. Luettavissa: <https://urn.fi/URN:NBN:fi:amk-2012091313612>. Luettu: 19.4.2025.

Teknologiateollisuus ry 2025. Tietotekniikka-alan uudet tilaukset Suomessa (Talousnäkymät helmikuu 2025). Luettavissa: <https://teknologiateollisuus.fi/graafit/tietotekniikka-alan-uudet-tilaukset-suomessa-talousnakymat-helmikuu-2025/>. Luettu: 3.3.2025.

Tilastokeskus 2024a. Suomen virallinen tilasto (SVT): Työssäkäynti. Luettavissa: <https://stat.fi/tilasto/tyokay>. Luettu: 9.4.2025.

Tilastokeskus 2024b. Suomen virallinen tilasto (SVT): Sijoittuminen koulutuksen jälkeen. Luettavissa: <https://stat.fi/tilasto/sijk>. Luettu: 4.3.2025.

Tuomi, J. & Sarajärvi, A. 2018. Laadullinen tutkimus ja sisällönanalyysi. Uudistettu laitos. Kustannusosakeyhtiö Tammi. Helsinki. E-kirja. Luettu: 6.1.2025.

Vilkkä, H. 2015. Tutki ja kehitä. PS-kustannus. Jyväskylä.

Vuorijärvi, A. 2013. Tekstilaji ja yhteisö: Ammattikorkeakoulun opinnäytetyön diskussio tekstinä. Väitöskirja. Helsingin yliopisto, humanistinen tiedekunta. Luettavissa: <https://helda.helsinki.fi/items/e6e34e3e-d2ec-4a72-be71-39e57dee9d10>. Luettu: 30.4.2025.

Warr, M., Driskell, S., Langran, E., Mouza, C. & Schmidt-Crawford, D. 2023. Curriculum design for technology infusion: A continuous collaborative process. *Contemporary Issues in Technology and Teacher Education*, 23(1), s. 124–150.

Liitteet

Liite 1. Aineistosta sisällönanalyysissä muodostetut aiheet

Tässä on esitetty muodostamamme 39 aihetta, joihin jaottelimme tutkimusaineistosta poimimme havainnot.

Aineistosta muodostetut pääkategoriat:

1. Projektinhallinta- ja kommunikaatiotyökalut
2. IDE ja siihen liittyvät
3. Ohjelmointikielet
4. Front end -teknologiat
5. Front end -kirjastot
6. Viitekehykset
7. Virtualisointi
8. Konttiteknologiat ja orkestrointi
9. Pilvipalvelut
10. Back end -teknologiat
11. Back end -kirjastot
12. Tietokannat
13. Testaustyyppi
14. Testaustyökalut ja -teknologiat
15. CI/CD
16. Versionhallinta
17. Analytiikka ja monitorointi
18. UI/UX-suunnittelu ja siihen liittyvät
19. Projektinhallintamenetelmät
20. Toistuvat ohjelmistokehityksen tehtävät
21. Tiedonhaku
22. Pehmeät taidot
23. Tietoliikenne ja rajapinnat
24. Skriptaust
25. Tietoturva
26. Migraatio
27. Muut teknologiset osaamiset
28. Sisällönhallinta- ja asiakkuudenhallintajärjestelmät
29. Integraatioalustat

- 30. Käyttöjärjestelmät
- 31. Etä-/lähityö
- 32. Sertifikaatit
- 33. Tukevat työkalut
- 34. Tekoäly
- 35. Infrastruktuuri ja konfigurointi
- 36. Datat prosessointi ja lokitus
- 37. Varmuuskopiointi ja palautus
- 38. Muut teemat
- 39. Keskeiset huomiot