



jamk

Tekoäly osana ohjelmistokehitystä

Integratiivinen kirjallisuuskatsaus

Jarkko Tervakoski

Opinnäytetyö, ylempi AMK

Toukokuu 2024

Teknologiaosaamisen johtaminen

Tervakoski Jarkko

Tekoäly osana ohjelmistokehitystä

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2024, 79 sivua.

Teknologiaosaamisen johtamisen tutkinto-ohjelma. Opinnäytetyö ylempi AMK.

Julkaisun kieli: Suomi

Julkaisulupa avoimessa verkossa: Kyllä

Tiivistelmä

Tekoäly on viime vuosina ollut termi, jonka lähes jokainen meistä on kuullut. Vuoden 2022 loppupuolella tehty julkaisu uudesta tekoälyohjelman versiosta mahdollisti käyttäjäkokemukseltaan täysin uudenlaisen lähestymistavan tekoälyn hyödyntämiseen. Tekoälyn hyödyntäminen etenkin ohjelmistokehityksen parissa on ollut erittäin ajankohtainen aihe ja siitä tuotetaan jatkuvasti uutta tutkimusmateriaalia.

Integratiivisen kirjallisuuskatsauksen avulla tuotiin yhteen uusimpia tutkimuksia tekoälyn hyödyntämisestä ohjelmoinnissa. Katsaukseen valittu aineisto hankittiin ProQuest, ScienceDirect ja IEEE Xplore tietokantoja hyödyntäen. Tietokantojen haut suoritettiin ennalta-asetettujen hakusanojen avulla. Hakutulokset käsiteltiin manuaalisesti ja lopulliset katsaukseen valitut tutkimukset valittiin laatuarvioinnin perusteella.

Kirjallisuuskatsauksen tuotoksena muodostui hyvin kattava aineisto, jonka pohjalta pystyttiin muodostamaan luotettavia tuloksia tämänhetkisestä tekoälyn hyödyntämisen valmiudesta ohjelmistokehityksessä. Kattavan aineiston avulla tutkimusten välille pystyttiin luomaan synergioita, mikä vahvisti tuloksien luotettavuutta.

Ohjelmistokehitys on selkeässä murrostilassa tekoälypohjaisten työkalujen takia. Katsauksessa läpikäytyjen tulosten avulla voitiin selkeästi päätellä, että tekoälyn hyödyntäminen ohjelmointitehtävissä nostaa kehittäjien tuottavuutta, koodin laatua sekä työntekijäkokemusta. Tämän lisäksi tekoälyn todettiin parantavan ohjelmistojen laatua.

Yritysten on suositeltavaa lähteä perehtymään tekoälyn tarjoamiin etuihin omalla toimialallaan erityisesti ohjelmistokehityksen osalta. Useat kehittäjät hyödyntävät tekoälyä jo päivittäisissä tehtävissään ja etenkin yrityksen tietoturvan osalta on tärkeää, että yritys on mukana murroksessa tukemassa työntekijöitä ja luomassa yhteisiä pelisääntöjä. Tekoälyn hyödyntämisessä epäonnistuvat yritykset tulevat väistämättä jäämään kehityksessä jälkeen yrityksistä, jotka pystyvät valjastamaan tekoälyn potentiaalin edukseen.

Avainsanat (asiasanat)

Integratiivinen kirjallisuuskatsaus, tekoäly, tietokoneohjelmointi, ohjelmistotekniikka, tietokoneohjelmien kehitystyökalut

Tervakoski Jarkko

AI in software development

Jyväskylä: JAMK University of Applied Sciences, May 2024, 79 pages.

Degree Programme in Technology Competence Management. Master's thesis.

Permission for open access publication: Yes

Language of publication: Finnish

Abstract

During the past few years, artificial intelligence (AI) has been a word which almost everyone has heard. AI application version which was released at the end of the year 2022 made possible a totally new user experience while utilizing AI. Utilizing AI in software development has been extremely current topic and new research materials are continuously created.

Integrative literature view collected latest researches about utilizing AI in programming. Material used in review was collected from ProQuest, ScienceDirect and IEEE Xplore databases. Database searches were conducted by predefined keywords. Search results were analyzed, and final results were selected by quality review.

Literature review managed to form a very comprehensive material which was the base for creating reliable results about the current status of AI's utilization maturity in software development. With the comprehensive results, it was possible to create synergies between researches which confirmed reliability of the results.

Software development is clearly in transition because of AI based tools. The results, which literature review provided, clearly indicated that utilizing AI in programming increases developers' productivity, quality of the code and employee experience. Additionally, it was noted that using AI will also increase a quality of the software.

It is highly recommended that companies start to familiarize themselves with the benefits which AI can provide in their own industries and especially in software development. Most developers are already using AI in their daily tasks and just because of information security it is crucial that companies start to get involved by supporting them and creating common rules. Companies who fail to adapt the AI will inevitably fall behind in development with the companies which are able to harness the potential of AI for their advantages.

Keywords/tags (subjects)

Integrative literature review, artificial intelligence, computer programming, software engineering, software development tools

Sisältö

1	Johdanto	8
2	Tutkimuksen tausta	10
2.1	Toimeksiantajan tarve.....	10
2.2	Ohjelmistokehityksen haasteet.....	12
2.3	Tekoälyn nopea kehitys.....	15
3	Tutkimuksen tarkoitus	16
3.1	Tutkimuksen tarkoitus ja tavoitteet.....	16
3.2	Tutkimuskysymykset	17
3.3	Tutkimuksen rajaukset	18
3.4	Aikaisemmat tutkimukset	19
4	Tekoälyn käsitteet	20
4.1	Tekoälyn määritelmä.....	20
4.2	Koneoppimisen ymmärtäminen.....	21
4.3	Prediktiivinen tekoäly.....	23
4.4	Generatiivinen tekoäly	24
5	Tekoälyn korkean tason toimintamalli	25
5.1	Tekoälyn peruskomponentit	25
5.2	Koneoppimisen hyödyntäminen	26
5.3	Keinotekoisten neuroverkkojen toiminta	28
5.4	Kielimallien käyttäminen.....	31
6	Kirjallisuuskatsauksen toteutus.....	33
6.1	Integratiivinen kirjallisuuskatsaus tutkimusmenetelmänä	33
6.2	Aineiston hankinta	34
6.3	Haun tulokset ja valinnat	38
6.4	Aineiston laadun arviointi	40
6.5	Aineiston analysointi	41
7	Tulokset.....	42
7.1	Tekoälyn kypsyys ohjelmoinnissa	42
7.2	Koulutuksen tehostaminen	47
7.3	Tekoäly ohjelmoijan tukena	49
7.4	Testauksen kehittäminen.....	55
7.5	Turvallisuuden ja luotettavuuden huomioiminen	57

8 Pohdinta	58
8.1 Tutkimuksen luotettavuus	58
8.2 Tutkimuksen eettisyys.....	59
8.3 Johtopäätökset.....	60
8.4 Jatkotutkimusehdotukset.....	64
Lähteet	66
Liitteet	70
Liite 1. Kirjallisuuskatsauksessa käytetyt lähteet	70
Liite 2. Valittujen aineistojen laadunarviointi	72

Kuviot

Kuvio 1. Tekoälyn hyödyntäminen eri liiketoiminta-alueilla	9
Kuvio 2. Gartnerin hypekäyrä AI sovelluksille.....	16
Kuvio 3. Esimerkkejä tekoälyn käyttötapauksista.....	24
Kuvio 4. Prediktiivisen ja generatiivisen tekoälyn erot.....	25
Kuvio 5. Tekoälyn peruskomponentit	26
Kuvio 6. Esimerkki koneoppimisen päätöspuusta	27
Kuvio 7. Vahvistusoppimisen prosessi	28
Kuvio 8. Yksinkertainen neuroverkko	29
Kuvio 9. Syvä neuroverkko useammalla piilokerroksella.....	30
Kuvio 10. Luonnollisen kielen käyttötapauksia.....	31
Kuvio 11. Kirjallisuuskatsauksen vaiheet	34
Kuvio 12. Aineiston valinta PRISMA 2020 -diagrammin mukaisesti.....	39
Kuvio 13. Tekoälyn avulla parantaminen eri osa-alueilla	43
Kuvio 14. Tekoälytyökalujen tärkeimpiä käyttötapauksia	44
Kuvio 15. Tekoälytyökalujen käyttötapauksia muun muassa toteutusvaiheessa	45
Kuvio 16. Eri pilottiryhmien erot tehtävien ratkaisuisa	46
Kuvio 17. Tekotyökalun haasteet.....	46
Kuvio 18. Tekoälyn tilanne yrityksissä käyttökohteiden perusteella jaettuna	47
Kuvio 19. Perusteluita työkalun käytön puolesta ja vastaan	50
Kuvio 20. Miten vastaajat kokevat tekoälyn uhkana ammatilleen.....	50
Kuvio 21. Tekoälytyökalujen käyttöaste	51
Kuvio 22. Tekoälyn ongelmanratkaisukyky ongelman vaikeutuessa.....	52

Kuvio 23. Kehittäjien huolia tekoälyn tuottamasta koodista	57
Kuvio 24. Huolestuttavimmat alueet tekoälyn käytössä	58

Taulukot

Taulukko 1. Tunnettujen ohjelmistojen koodirivien lukumääriä miljoonissa	13
Taulukko 2. Ohjelmointikielien yleisyys	14
Taulukko 3. Ensimmäisen vaiheen hakusanat	36
Taulukko 4. Tietokantahakujen lopulliset hakusanat	37

Lyhenteet

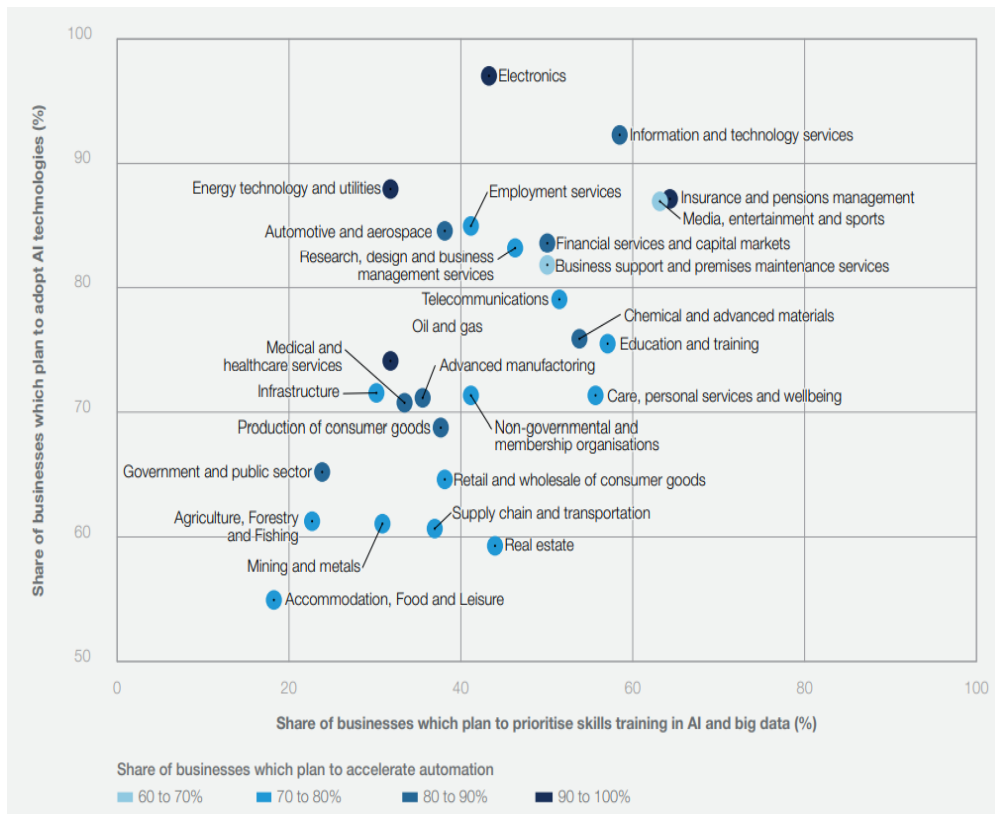
Lyhenne	Selite	Käännös
AI	Artificial Intelligence	Tekoäly
ANN	Artificial Neural Network	Keinotekoinen neuroverkko
ASR	Automatic Speech Recognition	Automaattinen puheentunnistus
DL	Deep Learning	Syväoppiminen
IOT	Internet Of Things	Esineiden internet
LLM	Large Language Model	Suuri kielimalli
NLP	Natural Language Processing	Luonnollisen kielen käsittely
ML	Machine Learning	Koneoppiminen
POC	Proof of Concept	Konseptitodistus
RL	Reinforcement Learning	Vahvistusoppiminen
SL	Supervised Learning	Ohjattu oppiminen
UL	Unsupervised learning	Ohjaamaton oppiminen

1 Johdanto

Vuoden 2022 marraskuussa OpenAI niminen yhtiö julkaisi kehittämänsä ChatGPT-3.5 version avoimeksi kaikille. Tämän julkaisun myötä tekoäly tuli saataville yhä suuremmille massoille ja tekoälyn hyödyntäminen sekä sen potentiaalin kartoittaminen on tullut ajankohtaiseksi niin yksityishenkilöille kuin myös yrityksille. Tekoälyn avulla yritykset voivat toimialasta riippumatta kehittää omaa toimintaansa tehokkaammaksi, vapauttaen työntekijöitään triviaaleimmista tehtävistä niiden tehtävien pariin, joihin tekoäly ei vielä tänä päivänä kykene.

Jopa 75 % yrityksistä suunnittelee ottavan tekoälyn käyttöönsä seuraavan viiden vuoden aikana. Merkittävä osa yrityksistä uskoo, että tekoäly tuo mukanaan muutosta työmarkkinoille. Muutos aiheuttaa esimerkiksi työtehtävien murroksen yrityksissä. Yli 50 % yrityksistä kuitenkin uskoo, että tekoälyn mahdollistamana yrityksissä voidaan kasvattaa työpaikkojen lukumäärä. Yhtenä nopeitten kasvavana tarpeena työmarkkinoilla nähdään nimenomaan tekoälyyn ja koneoppimiseen liittyvien erityisasiantuntijoiden rooli. (Future of Jobs Report 2023, 4–5.)

Kuvion 1 perusteella voidaan todeta, että tekoälyn hyödyntäminen nähdään hyödyllisempänä aloilla, joissa tehdään selkeästi dataan perustuvia päätöksiä, kuten vakuutus- ja finanssialalla. Myös tietotekniikan ja teknologiapalveluiden alueella tekoälyn hyödyntämisen uskotaan olevan oleellisessa osassa ratkaisuja ja ratkaisuiden tuottamista. Tekoälyn nähdään vähiten hyödyttävän liiketoiminta-alueita, joiden keskiössä on ihminen, kuten asumisessa, ruuassa ja vapaa-ajassa. Kuitenkin yli 50 % yrityksistä uskoo tekoälyn palvelevan myös edellä mainittuja toimialoja, mutta henkilöstön koulutusta tekoälyn hyödyntämiselle ei nähdä olennaiseksi. Myös elektroniikan valmistamiseen tekoälytoteutuksien uskotaan tuovan merkittäviä hyötyjä esimerkiksi valmistusprosesseissa, laadunhallinnassa ja monissa muissa elektroniikan valmistukseen kuuluvissa prosesseissa (Panchal 2023). Mikään ala ei siis kokonaan pysty välttämään tekoälyä, vaan tekoälyn nähdään antavan hyötyjä liiketoimintaan erinäisten toteutuksien avulla.



Kuvio 1. Tekoälyn hyödyntäminen eri liiketoiminta-alueilla (Future of Jobs report 2023, 46)

Tekoälyn ja siihen liittyvien toimintamallien sekä teknologioiden kehitys on huimassa kasvussa ja uutta tutkimusta julkaistaan hyvin nopealla tahdilla. Tässä tutkimuksessa toteutetaan kirjallisuuskatsaus alan uusimpiin julkaisuihin, joiden avulla muodostetaan kuva tämänhetkisistä tekoälyn hyödyntämismahdollisuuksista ohjelmistotuotannossa. Ymmärrys tekoälystä ja sen käyttömahdollisuuksista kasvaa jatkuvasti. Tutkimuksessa tarkastellaan myös, mitä mahdollisia ongelmia tekoälyn hyödyntäminen ohjelmiston kehittämisessä saattaa aiheuttaa. Yrityksen on tärkeää ymmärtää riskit, jotka uuden teknologian käytöstä aiheutuvat, jotta nämä osataan ottaa huomioon alusta alkaen, jolloin myös riskien mitigointisuunnitelma kannattaa tehdä.

Ennen tutkimuksen kirjallisuuskatsauksen osiota tutkimuksessa käydään lävitse tekoälyn liittyviä ajankohtaisia asioita, joiden avulla lukijalle pyritään antamaan hyvä ymmärrys nykypäivän tekoälyn toiminnasta. Taustatietojen perusteella on helpompi ymmärtää kirjallisuuskatsauksessa esiin nousevat asiat ja miten tuloksien perusteella voidaan luoda ymmärrys tekoälyn nykytilasta ohjelmoinnissa ja potentiaalisista jatkotutkimusehdotuksista.

Tutkimuksen tavoitteena on vaikuttaa siihen, miten tekoälyä tullaan hyödyntämään toimeksiantajayrityksen ohjelmointikehityksessä. Uusimpien tekoälyjulkaisujen myötä useat yritykset ovat julkaisseet omia tekoälytyökalujaan, jotka pohjautuvat aikaisempiin tekoälyratkaisuihin, kuten OpenAI-yrityksen julkaisemaan kielimalliin. Tutkimuksessa tarkastellaan kirjallisuuskatsauksen avulla, miten tekoälyn hyötyjä on tutkittu ohjelmistokehityksessä. Näiden tulosten perusteella pyritään löytämään tapoja tehostaa ohjelmistokehitystä ja vapauttamaan enemmän aikaa suunnittelulle ja niiden ongelmien ratkaisuihin, joihin tekoäly ei vielä tänä päivänä kykene.

Tutkimuksen lopputuloksella voi olla potentiaalisesti hyvinkin merkittävä rooli siihen, miten ohjelmistoja tulevaisuudessa toteutetaan. Tekoälyn on arveltu korvaavan yksinkertaiset ja itseään toistavat työtehtävät lähitulevaisuudessa, mutta myös tehostavan monia eri työtehtäviä, jotka ovat nykypäivän kehittäjälle jo hyvin rutiininomaisia. Ohjelmistokehittäjien työtehtävät muodostuvat myös monista muista tehtävistä kuin itse ohjelmoinnista ja tästä syystä aihetta on hyvä tarkastella myös hieman ohjelmointia laajemmin, kuten esimerkiksi dokumentoinnin ja testaamisen näkökulmasta.

Tutkimuksen lopputuloksena on yhteenveto tekoälyn tämänhetkisestä tilasta, jonka perusteella toimeksiantaja pystyy tekemään päätöksensä tekoälyn hyödyntämisen suhteen. On hyvin todennäköistä, että kilpailukykyisenä pysyvän ohjelmistotalon on pystyttävä investoimaan tekoälyyn oikeaan aikaan. Myös eri liiketoiminta-alueilla toimivilla ohjelmistotaloilla saattaa olla eroja etenkin ohjelmistokehityksen eri vaiheissa, mutta työssä keskitytään tarkastelemaan ohjelmistokehitystä hieman yleisemmin, jotta se palvelisi mahdollisimman montaa henkilöä tai yritystä, jotka pohtivat tekoälyn hyödyntämismahdollisuuksia tulevaisuudessa ohjelmistokehityksen parissa.

2 Tutkimuksen tausta

2.1 Toimeksiantajan tarve

Opinnäytetyön toimeksiantajana toimii Oy Samlink Ab, joka on nykyään osa amerikkalaisomisteista Kyndryl Oy:ta. Kyndryl on vuonna 2021 IBM:stä irtaantunut yritys, jolla on hyvin vahva osaaminen tietotekniikan infrastruktuurin puolelta. Samlinkilla on yli 30 vuoden kokemus finanssialan ohjelmistokehityksestä. Ohjelmistokehitystä tehdään Samlinkissa hyvin kattavasti aina asiakastarpeesta loppukäyttäjän tukemiseen saakka.

Vuosien saatossa digitalisoinnin tarve on vain kiihtynyt ja järjestelmien monimutkaisuus on samalla moninkertaistunut. Tämä tuo mukanaan kompleksisuutta, minkä takia ohjelmistokehityksen prosesseja ja toimintatapoja on tärkeää jatkuvasti kehittää. Prosessien lisäksi, tämä tarkoittaa myös ohjelmistokehittäjien arjen sujuvoittamista ja tukemista ohjelmistojen kehityksessä.

Toimeksiantaja toimii tietotekniikan toimialalla toimittaen ratkaisujaan pääasiassa finanssialan yrityksille. Näiden molempien toimialojen ennustetaan olevan tekoälyn todennäköisimmästä ja eniten käytetyimmistä toimialoista niin ratkaisuiden kuin toteuttamisen osalta. Tietotekniikan aloista jopa yli 90 prosenttia kuvion 1 perusteella tulee ottamaan tekoälypohjaisen teknologian käyttöönsä tehostamaan toimintaansa. Myös finanssialalla vastaava luku on yli 80 prosenttia. Myös joka toinen yritys kyseisillä toimialoilla aikoo kouluttaa työntekijänsä tekoälyohjelmistojen hyödyntämiseen. Näiden kahden toimialan tekoälyn käyttöönottoennusteiden perusteella tekoälyn omaksuminen hyvin monipuolisesti, tulee olemaan hyvin oleellista toimeksiantajan yrityksessä kilpailukyvyyn ylläpitämiseksi, alati kehittyvässä kilpailussa.

Biswas, Carson, Chung, Singh ja Thomas (2020) olivat tutkineet, että tekoälysovellusten hyödyntäminen finanssialalla voi kasvattaa pankkien arvoa jopa yli miljardin verran. Edellä mainittu luku perustuu siihen, että yritykset pystyvät kasvattamaan liikevaihtoa lisäämällä personalisoituja ratkaisuja asiakkailleen, vähentämällä kustannuksia muun muassa automaatioiden sekä virheiden vähentämisen avulla sekä löytämällä uusia mahdollisuuksia etenkin datasta jalostettujen ratkaisujen avulla. Biswas ja muut (2020) myös uskovat pankkien, jotka eivät omaksu tekoälypohjaisia ratkaisuja mukaan liiketoimintaansa, asettavan itsensä vaaraan tulevaisuudessa. Joko toinen tekoälyratkaisut omaksunut pankki voi vallata yrityksen tai asiakkaat valitsevat pankin, joka pystyy tekoälyratkaisujen avulla tarjoamaan tehokasta ja monipuolista, mutta kuitenkin personalisoitua palvelua asiakkailleen. Koronapandemian vuodet kiihdyttivät digitaalisten palveluiden kehitystä, sillä esimerkiksi käynnit pankkien konttoreilla olivat haastavia. Monet palvelut haluttiin tästä syystä siirtää internetiin. Moni ratkaisu tarvitsee tekoälyratkaisuja toimiakseen tehokkaasti ja asiakaskokemusta vahvistaen. Yleisimmät tekoälyä hyödyntävät ratkaisut monelle loppukäyttäjälle ovat chatbotit. Taustalla kuitenkin pyörii myös monia muita päätöksiä vauhdittavia ratkaisuja, joissa hyödynnetään tekoälyä ohjelmistorobotiikan avulla. Näissä tekoälyn avulla suoritetaan rutiininomaisia tehtäviä ja erilaisia koneoppimISRatkaisuja mm. petoksien havaitsemiseksi.

Toimeksiantajayrityksen on siis pystyttävä luomaan tekoälypohjaisia ratkaisuja asiakkailleen, mikä on aikaisempien tutkimusten mukaan oletus toiminnan jatkumiselle. Ei riitä, että ymmärretään asiakkaan tarve ja tahtotila, vaan tämän lisäksi on tärkeää pystyä tarjoamaan tekoälynpohjautuvia ratkaisuja, jotka voivat mahdollistaa asiakkaan toiminnan alati muuttuvassa ja kehittyvässä maailmassa. Vaikka innovointi ja uudet teknologiat avaavat uusia ovia, niin ne myös saattavat monimutkaistaa sovelluksia ja ennen kaikkea niiden luominen vie aikaa. Rutiininomaisia ohjelmointitehtäviä tehostamalla voidaan ohjata myös ohjelmistokehittäjien aikaa paremmin luomaan uusia innovatiivisia ratkaisuja tekoälyä hyödyntäen asiakkaalle.

Tarve ohjelmistojen nopeammalle ja laadukkaammalle kehittämiselle luo hyvän pohjan tämän opinnäytetyön aiheelle, jossa tarkastellaan, miten tämän hetken tekoälytoteutuksia voitaisiin hyödyntää ohjelmistokehityksessä. ”Tekoälyn käyttö on tulevaisuuden työelämätaito, käyttäväthän kaikki nyt jo tietokoneita, nettiä ja ainakin taskulaskimia” (Kivimäki 2023, 57).

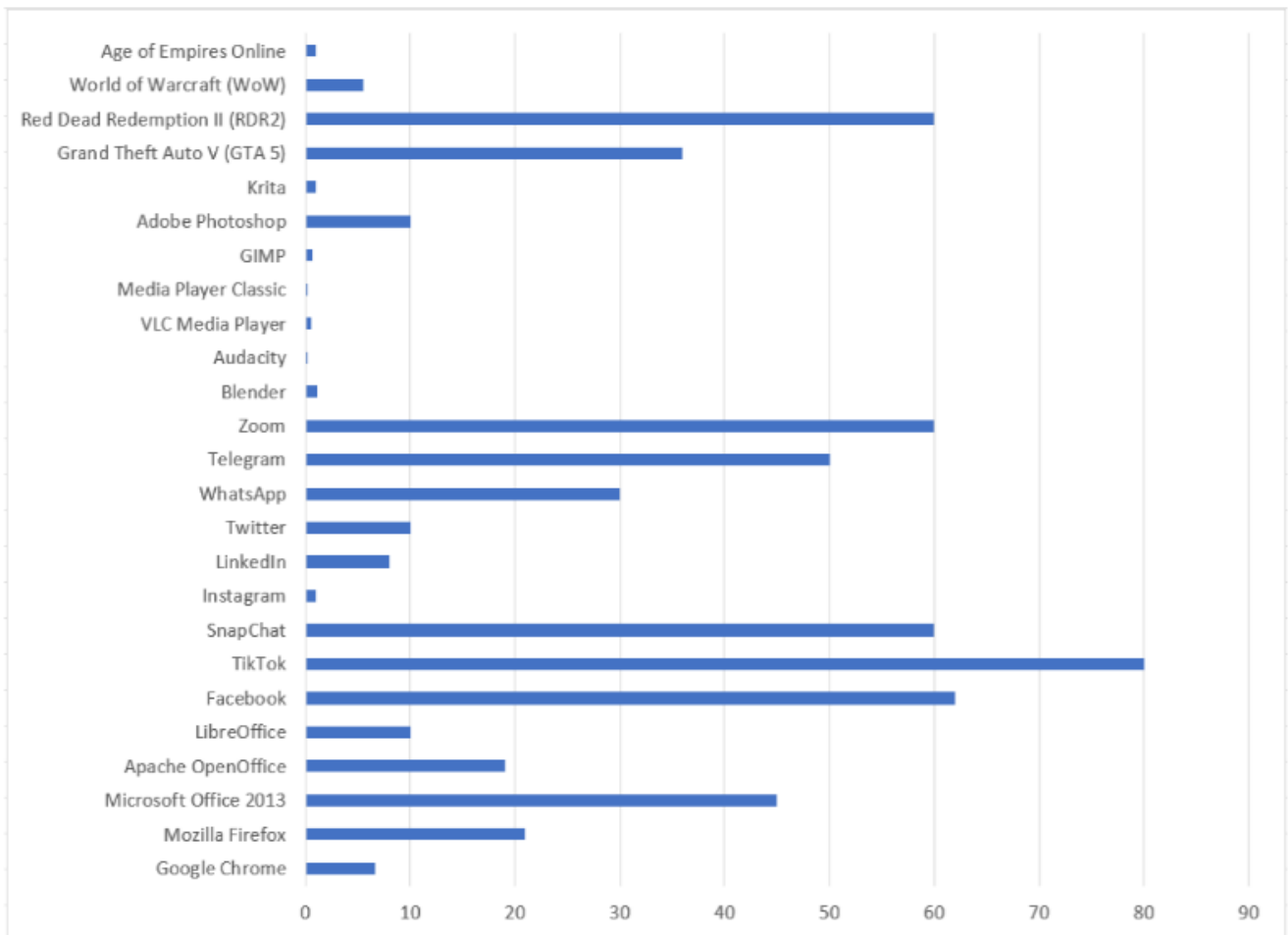
2.2 Ohjelmistokehityksen haasteet

Nykypäivän ohjelmistokehityksen haasteena ovat muun muassa ohjelmistojen laajuus, olemassa olevien järjestelmien teknologioiden vanhuus sekä nopeus, jolla ohjelmistoja ja ohjelmistojen ominaisuuksia on luotava. Näiden lisäksi on pidettävä huolta vaatimuksista, jotka eivät toimiessaan näy käyttäjälle, kuten sovelluksen nopeus ja tietoturva. Mikään näistä haasteista ei ole häviämässä, vaan päinvastoin ne monimutkaistuvat vuosi vuodelta. Näiden haasteiden takia tarvitaan uusia prosesseja ja työkaluja kehittämään ja ylläpitämään nykyisiä sekä tulevia ohjelmistoja.

Ohjelmistojen laajuus

Nykypäivän ohjelmistojen laajuutta voi hyvin avata siten, että vuonna 1969 laukaistu avaruusraketti Apollo 11 toteutettiin 145 000 koodirivin avulla. Tänä päivänä Boeing 787 -lentokone käyttää 7 miljoonaa riviä koodia ja modernit autot sisältävät arviolta jopa 10–100 miljoonaa koodiriviä. Esimerkkien avulla voidaan päätellä, että tällaiset määrät koodia lisäävät ohjelmistojen kompleksisuutta ja hallittavuutta, tuoden mukanaan lukuisia prosesseja ja kasvavia työmääriä. (Algaze 2017.) Lisää esimerkkejä nykypäivän ohjelmistojen suuruudesta voidaan tarkastella taulukosta 1.

Taulukko 1. Tunnettujen ohjelmistojen koodirivien lukumääriä miljoonissa (Fernández 2023)



Teknologioiden vanhuus

Teknologian vanhuudesta COBOL-ohjelmointikieli on edelleen laajasti käytössä oleva esimerkki. COBOL on vuonna 1959 kehitetty ohjelmointikieli ja yli 90 % Fortune 500 -listan yrityksistä hyödyntää COBOL:lla ohjelmoituja järjestelmiä. Tämä on hyvin yleistä esimerkiksi finanssialan yrityksissä (Vaughan-Nichols 2019). Haasteeksi tämä kohta eläkeiän saavuttanut ohjelmointikieli luo sen, että sitä osaavat kehittäjät vähenevät. Esimerkiksi koronaviruksen alussa COBOL-koodareista oli huu-tava pula Yhdysvalloissa (Kangasniemi 2020). On siis selvää, että myös vanhojen ohjelmointikielien omaksumiseen kuin migratoimiseen uudempiin ohjelmointikieliin tarvitaan uusia työkaluja ja ta-poja toimia, jotta olemassa olevia järjestelmiä voidaan ylläpitää ja jatkokehittää tulevina vuosina, kunnes ne voidaan lopullisesti korvata moderneimmilla järjestelmillä uusia ohjelmistoteknologioita hyödyntäen. Ohjelmistokielen kirjo on tänä päivänä valtava, kuten taulukosta 2 voi havaita.

Taulukko 2. Ohjelmointikielien yleisyys (Programming Languages Products n.d)

Ohjelmointikieli	Yritysten lukumäärä
PHP	7632341
Java	448975
Python	250716
Perl	124006
PL/SQL	62039
Kotlin	37376
COBOL	32572
ABAP	30465
XPath	15072
Fortran	12026
Microsoft Visual FoxPro	5733
SAS Base	5720
Jython	3231
Scala	2855
Informix 4GL	1196

Ohjelmistotuotannon nopeus

Digitalisoituminen, korkean tason ohjelmointikielien, kehitystyökalujen saatavuus sekä jakelukanavien kehittyminen ovat vauhdittaneet ohjelmistojen kehitystä ja kasvattaneet kilpailua, mutta samalla laajentaneet markkinoita uusien sovelluksien avulla. Applen sovelluskauppaan julkaistaan päivittäin keskiarvolta melkein 1000 eri sovellusta (iOS Apple App Store Statistics and Trends 2024). Myös Googlen sovelluskauppaan julkaistaan keskiarvolta melkein 3800 sovellusta päivittäin (Sharma 2024). Tämä tarkoittaa yhteensä melkein 5000 uutta mobiilisovellusta päivittäin niin vapaa-ajalle kuin työelämään.

Myös suurempien sovellusten tuotantoon vieni saattaa tapahtua useita kertoja päivässä. Esimerkiksi Stack Overflow niminen yritys kertoi jo vuonna 2016 julkaisevansa päivässä 5–10 uutta versiota tuotantoon (Stack Overflow: How We Do Deployment – 2016 Edition 2016). Tämä tarkoittaa hyvin automatisoitua jatkuvaa integrointi- ja julkaisuprosessia. Tämän lisäksi ohjelmistojen kehittäjillä tulee olla oikeat prosessit ja työkalut ohjelmoidakseen toimintoja tehokkaasti valmiiksi.

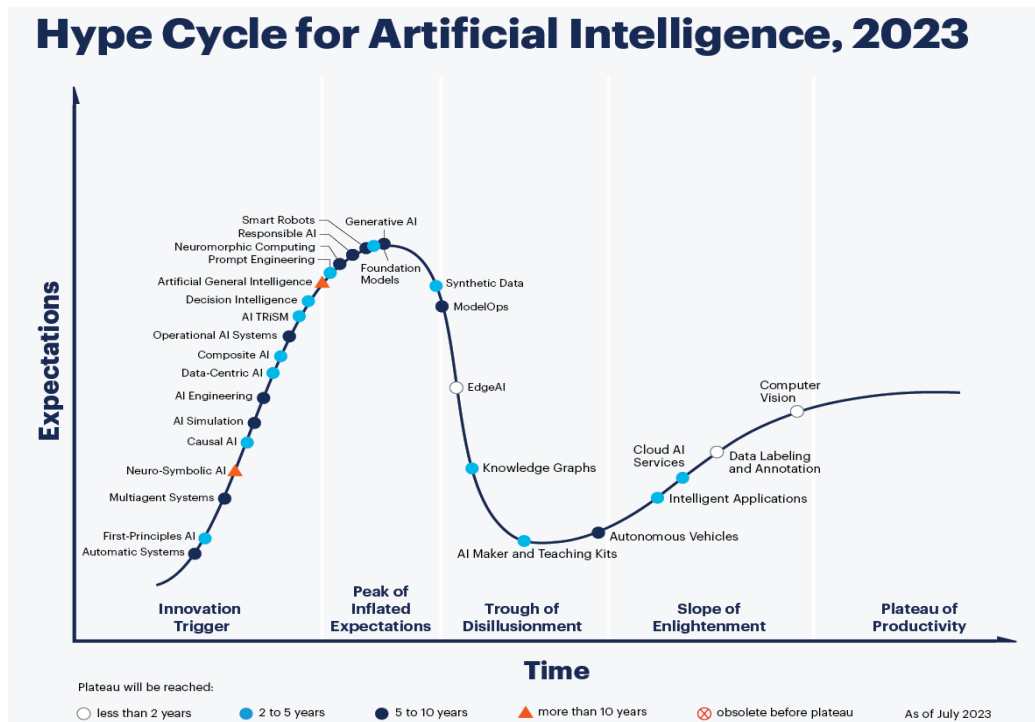
Nopeus ja tietoturva

Ohjelmistojen nopeus on käyttäjäkokemuksen kannalta yksi oleellisimpia asioita. Nykyajan kiireisellä ihmisellä ei ole aikaa eikä kärsivällisyyttä odottaa sivuston lataamista kovin pitkään, koska muitakin vaihtoehtoja on. Digital-sivuston vuonna 2022 teettämän kyselyn perusteella yli 50 % käyttäjistä lähtee verkkosivulta, jos sivuston lataaminen kestää yli 6 sekuntia (Website Speed Stats: Does Your Site Load in 3 Seconds? 2022).

Järjestelmien monimutkaisuus haastavine arkkitehtuureineen, kolmannen osapuolen integraatiot sekä järjestelmiin lisättävät kirjastot lisäävät järjestelmiin tietoturva-aukkoja, joiden havainnointi voi olla hyvinkin haastavaa. Tämän lisäksi järjestelmät vanhenevat, eikä niihin välttämättä tule enää päivityksiä, mikä luo haasteen etenkin vanhojen järjestelmien ylläpitämiselle. Tämän lisäksi elämme hyvin aktiivista IoT-aikakautta, jossa laitteita liitetään yhä enemmän ja enemmän internettiin. (Challenges Faced by Software Security Engineers 2024.) Näiden kokonaisuuksien huomioiminen myös tietoturvan osalta haastaa ohjelmistokehittäjiä päivittäin ohjelmistojen kehittämisessä.

2.3 Tekoälyn nopea kehitys

Tekoäly ei ole keksintönä uusi ja sitä on hyödynnetty jo vuosikymmenien ajan erinäköisissä ohjelmistoissa. Pitkään tekoälyratkaisut olivat kuitenkin tarkkaan määriteltäviä ratkaisuja ja kaupalliset kohteet, joissa tekoälyä hyödynnettiin, olivat lähinnä tiedon keräämiseen ja analysointiin perustuvia ratkaisuja. Viimeisimmän 10 vuoden aikana teknologia on kuitenkin kehittynyt merkittävästi tukemaan juuri tekoälyyn pohjautuvia ratkaisuja. Tätä ovat vauhdittaneet pilviratkaisut, isojen datamäärien hallitseminen ja syväoppiminen neuroverkkojen avulla. Viimeisin tekoälyn läpimurto oli kehittynyt kielimalli, joka mahdollisti sujuvan tekoälyratkaisujen käyttämisen ilman syvällisempää ymmärrystä tekoälyn toiminnasta. Viime vuodet tekoäly on ollut yksi seuratuimmista teknologioista tuottaen lukuisia eri sovelluksia niin yksityis- kuin yrityskäyttöön. Myös Gartner on tunnistanut useita eri potentiaalisia tekoälysovelluksia lähitulevaisuuteen kuten kuvioista 2 käy ilmi.



Kuvio 2. Gartnerin hypekäyrä AI sovelluksille (What's New in Artificial Intelligence from the 2023 Gartner Hype Cycle 2023)

3 Tutkimuksen tarkoitus

3.1 Tutkimuksen tarkoitus ja tavoitteet

Tutkimuksen tarkoituksena on perehtyä integratiivisen kirjallisuuskatsauksen avulla tekoälyn hyödyntämiseen ohjelmistokehityksessä. Tutkimuksessa on tarkoitus käyttää mahdollisimman uusia julkaisuja, sillä tekoälyn kehitys on viime vuosina ollut todella nopeaa, minkä takia informaation luotettavuuden riski kasvasi merkittävästi vanhempien julkaisuiden myötä.

Tutkimuksen tavoitteena on saavuttaa ymmärrys tekoälytyökalujen tämänhetkisestä maturiteetista ja hyödyntämismahdollisuuksista ohjelmistokehityksessä. Lopputuloksesta on tärkeää käydä ilmi, voidaanko tekoälypohjaisten työkalujen avulla tehostaa ohjelmointia ja parantaa laatua ohjelmistokehityksessä. Lopputuloksesta on myös hyvä käydä ilmi, mitä yrityksen tulee ottaa huomioon, jos he päätyvät tekoälypohjaisia työkaluja käyttämään ja tuoko tekoäly mukanaan joitain rajoitteita. Yrityksellä on myös tulosten pohjalta ymmärrys siitä, tuleeko tekoälytyökalujen takia ottaa huomioon jotain esimerkiksi ohjelmointiprosessien aikana. Tavoitteena on, että lopputuloksesta käy ilmi, kannattaako tekoälypohjaisia työkaluja ottaa vielä käyttöön tukemaan ohjelmointia.

3.2 Tutkimuskysymykset

Tekoälyn kehitys on viimeisien vuosien aikana ollut todella vauhdikasta ja usein julkaistu materiaali vanhenee hyvinkin nopeasti. Mielenkiinto tekoälyä kohtaan on levittäytynyt tutkijoiden keskuudesta aina yritysmaailmaan ja ihmisten arkeen. Tekoälyn potentiaalia pidetään valtavana ja konkreettisia hyötyjä on jo nähty esimerkiksi tekstin tuottamiseen vaadituissa tehtävissä. Työn tavoitteena on parantaa ymmärrystä tekoälyn hyödyntämisestä ohjelmistokehityksessä.

Myös toimeksiantaja on viimeisien vuosien aikana huomannut tekoälyn vauhdikkaan kehityksen ja potentiaalin, minkä takia tekoälyn hyödyntämistä on tutkittu jo useammasta eri kulmasta. Yritys on jo vuosien ajan kehittänyt ohjelmistorobotiikan avulla erilaisia sovelluksia, joiden avulla loppukäyttäjän työkuormaa on voitu keventää merkittävästi. Tekoäly on siis toimeksiantajalle luontainen seuraava vaihe viedä ohjelmistoautomaatioita seuraavalle tasolle.

Toimeksiantajayritys on myös tarkasti perehtynyt tekoälytoteutuksien tietoturvaan ja datan käyttöön, sillä yritys toimii pankkisektorilla, jossa käsitellään paljon myös arkaluonteista asiakasdataa. Tällaisen tiedon vuotaminen julkiseksi on estettävä ja tämä on otettava huomioon, kun erilaisia tekoälytoteutuksia tutkitaan. Myös muun arkaluontoisen datan, kuten yrityssalaisuuksien vuotaminen julkiseksi on estettävä. Valitettavasti vastuu on kuitenkin usein käyttäjällä, sillä suurin osa generatiivisista tekoälyistä ja etenkin tekoälypohjaisista työkaluista vaatii edelleen syötteen, joka useimmiten tulee käyttäjältä. Tästä syystä on käytännössä mahdotonta luoda täysin aukottomia ohjeistuksia ja prosesseja, mutta jatkuvalla kouluttamisella riskit voidaan minimoida.

Toimeksiantajan on pyrittävä jatkuvasti osakeyhtiönä parempaan tulokseen. Yksinkertaistettuna tämä tarkoittaa myynnin kasvattamista ja kulujen vähentämistä. Tekoälyn käytön oletetaan auttavan kummassakin edellä mainituista asioista. Jos ohjelmointitehtäviä voidaan tehdä nopeammin ja laadukkaammin, niin kuluja saadaan pienennettyä. Myös kehittäjiä voidaan siirtää nopeammin uusien projektien pariin ja myyntiä voidaan suorittaa enemmän. Kehittäjien aikaa voidaan käyttää myös enemmän uusien asioiden innovointiin niin myynnin kuin toiminnan kehittämisen puolella. Täten opinnäytetyön kannalta tärkeimmät kysymykset, joihin haetaan vastausta ovat seuraavat:

1. Voidaanko tekoälyn avulla tehostaa ohjelmistokehitystä?
2. Voidaanko tekoälyn avulla parantaa ohjelmistokehityksen laadukkuutta?

Näihin kysymyksiin pyritään löytämään vastaukset integratiivisen kirjallisuuskatsauksen avulla, jossa katsauksen kohteena ovat aiheesta jo olemassa olevat tutkimukset. Molemmat kysymykset ovat lopputuloksen kannalta olennaisia, sillä menestyvän ohjelmistotalon on pystyttävät parantamaan niin ohjelmistokehityksen tehokkuutta kuin ohjelmistojensa laadukkuutta.

3.3 Tutkimuksen rajaukset

Ohjelmistokehitysprosessin eri vaiheet voidaan päätasolla karkeasti jakaa seitsemään eri vaiheeseen:

1. Asiakastarpeen ymmärtäminen
2. Ohjelmiston toiminallisuuden määrittelemine
3. Toteutuksen suunnittelu
4. Ohjelmiston ohjelmointi
5. Ohjelmiston testaaminen
6. Ohjelmiston julkaiseminen
7. Ohjelmiston ylläpito

Näiden päävaiheiden alle mahtuu lukuisia eri alavaiheita sisältäen tehtäviä laidasta laitaan. Tekoälyn hyödyn kartoittaminen näissä kaikissa eri vaiheissa ei ole toimeksiantajan kannalta tehokkain tapa lähteä tutkimaan tekoälyn mahdollisuuksia. On myös hyvin mahdollista, että eri vaiheisiin on omat tekoälysovelluksensa, mikä myös laajentaisi toimeksiantajan alustavaa tarpeen selvittämistä merkittävästi. Tämän lisäksi tekoäly on jatkuvasti kehittyvä tieteenala, minkä takia on aluksi tärkeää keskittyä yrityksen kannalta oleellisimpiin alueisiin hyödyn optimoimiseksi. Tästä syystä työ rajattiin aiheeseen ja siihen liittyviin tehtäviin, joissa suurin osa yrityksen henkilöstöstä toimii päivittäin eli ohjelmointiin. Ohjelmointi sivuaa myös automaatiotestausta, joten tuloksia voidaan mahdollisesti hyödyntää myös sillä alueella.

Tutkimuksessa ei oteta huomioon eri toimialojen eroavaisuuksia tekoälyn suhteen eikä myöskään toimeksiantajan pääasiallista liiketoiminta-aluetta. Toimialalla ei pitäisi olla suoranaista yhteyttä ohjelmointitehtäviin sen jälkeen, kun määritelmät ja tekniset suunnitelmat ovat toteutettuina. Tämän lisäksi tutkimuksesta halutaan tehdä mahdollisimman geneerinen, jotta sitä voidaan hyödyntää laajasti eri toimialoilla ohjelmistokehityksen parissa.

Tutkimuksen kirjallisuuskatsauksen ei myöskään ole tarkoitus rajoittaa tiettyihin teknologioihin tai toimintatapoihin vaan tarkastella tekoälyn hyödyntämistä yleisesti ohjelmoinnin parissa. On loogista olettaa, että tekoälysovellusten tuki on parempi ohjelmointikielille, jotka ovat yleisimmin käytössä kahdesta syystä. Ensimmäisenä syynä on tekoälyn tuki suuremmille massoille, jos tuki tarjotaan yleisesti käytössä oleville ohjelmointikielille. Toiseksi tekoälyille, jotka hyödyntävät koneoppimista on olemassa enemmän opetusmateriaalia silloin, kun kyseessä on yleisesti käytössä oleva ohjelmointikieli. Näiden edellä mainittujen syiden perusteella tutkimuksen kannalta ei ole oleellista ottaa huomioon tekoälyn hyödyntämistä eri ohjelmointikielien parissa.

3.4 Aikaisemmat tutkimukset

Tekoälyyn liittyvien tutkimusten hakeminen Theseus-tietokannasta osoittaa selkeästi, että aiheeseen liittyvien tutkimusten määrä on ylemmässä ammattikorkeakoulussa nousujohteinen. Vuonna 2019 tekoälyyn liittyviä opinnäytetöitä löytyy vain 5 kappaletta, kun taas vastaava luku on vuonna 2023 ollut 58. Tekoälyn käyttö nousee esille monessa eri koulutusalan työssä tavalla tai toisella. Tekoälyn hyödyntämiseen ohjelmistokehityksessä liittyviä tutkimuksia ei kuitenkaan löydy. Lähimpänä oleva opinnäytetyö keskittyy ohjelmistorobotiikkaan, joka hieman sivuaa varsinaista ohjelmistokehitystä. Tekoälyn tutkiminen eri koulutusaloilla on vielä yleisempää ammattikorkeakoulun tasolla, mutta tämä selittyy todennäköisesti osittain myös sillä, että opinnäytetöitä tuotetaan enemmän. Vastaava opinnäytetöiden lukumäärä vuoden 2023 osalta on 270 kappaletta.

Integratiivisen kirjallisuuskatsauksen tekeminen on myös vielä toistaiseksi melko harvinaista tekniikan alalla. Theseuksen kannasta tekniikan alalle ei juurikaan löydy uudehkoja opinnäytetöitä, joissa tutkimusmenetelmänä on kirjallisuuskatsaus. Tähän todennäköisesti vaikuttaa muun muassa se, että tekniikan töiden tekeminen on yleensä käytännöntasolla yksinkertaisempaa ja toimeksiantajalle kannattavampaa. Toisin kuin esimerkiksi hoitoalalla, missä asioiden käytäntöön ottaminen ei aina ole yhtä yksinkertaista, jolloin kirjallisuuskatsaus voi olla tehokkaampi tapa tuottaa uutta tietoa alalle.

Tekoälyn tutkiminen ja hyödyntäminen on kuitenkin viime vuosina kasvanut räjähdysmäisesti eri käyttötarkoituksiin. Esimerkiksi haut ScienceDirect ja IEEE Xplore tietokantoihin tekoäly-termillä tuottaa satojatuhansia tuloksia, joista kymmeniätuhansia on julkaistu viime vuosien aikana. Näi-

den lisäksi alaa johtavat yritykset kuten Gartner ja Stack OverFlow ovat teettäneet aiheesta tutkimuksia. Tämä mahdollistaa hyvin kattavan kirjallisuuskatsauksen aiheesta, sillä uusia tutkimuksia teetetään jatkuvasti. Tämän lisäksi uusia tekoälytyökaluja tulee markkinoille yritysten ja kuluttajien saataville hyvin nopeasti. Tämän opinnäytetyön integratiivinen kirjallisuuskatsaus osuu tekoälyn kannalta erittäin hyvään kohtaan, sillä se pääsee tarkastelemaan jo tutkittua tietoa ja nitomaan tämänhetkistä tilannetta yhteen useasta eri näkökulmasta.

4 Tekoälyn käsitteet

Opinnäytetyössä tarkastellaan ohjelmoinnin kehittämistä liiketoiminnan kehittämisen näkökulmasta. Toimiakseen nykypäivän ohjelmistomarkkinoilla yrityksen on jatkuvasti pyrittävä kehittämään ohjelmistokehitysprosessejaan ja työkaluja, jotta ne voivat pysyä kilpailukykyisenä. Opinnäytetyössä tarkastellaan nimenomaan ohjelmointivaiheen tehostamista erilaisissa tehtävissä tekoälyä hyödyntäen. Tekoälyn toiminnan ymmärtämiseksi ohjelmoinnissa, tässä kappaleessa esitellään yleisimmät käsitteet, jotka linkittyvät aiheeseen.

4.1 Tekoälyn määritelmä

Tekoälystä (artificial intelligence) on luotu useampia eri määritelmiä. Vaikka määritelmissä on usein eroavaisuuksia, niin yhtenä määritelmänä voidaan pitää kykyä oppia ja suorittaa sopivia tekniikoita ongelmien ratkaisemiseksi ja tavoitteiden saavuttamiseksi jatkuvasti vaihtuvassa maailmassa (Manning 2020). Nämä edellä mainitut asiat toistuvat tavalla tai toisella lähes kaikissa määritelmissä.

Euroopan komissio on määritellyt tekoälyn seuraavanlaisesti: Tekoäly (AI) viittaa järjestelmiin, jotka osoittavat älykästä käyttäytymistä analysoimalla ympäristöään ja toteuttamalla toimenpiteitä osittaisella autonomialla saavuttaakseen tavoitteensa. Tämän lisäksi määritelmästä käy ilmi, että AI-pohjainen järjestelmä voi toimia täysin virtuaalisesti, kuten hakukoneena, puheen tai kasvojentunnistus ohjelmistoissa tai tekoäly voidaan sulauttaa fyysisiin järjestelmiin, kuten kehittyneisiin robotteihin tai itseohjautuviin autoihin. (A Definition of AI: Main Capabilities and Disciplines 2019.)

Toivonen (2023, 4) kirjoittaa tekoälystä hieman eri kulmasta, sillä hänen mukaansa kyseessä on metafora, missä tekoäly tarkoittaa tietokoneen toimintaa, mihin rinnastetaan ihmisen kaltaista älykkyyttä. Tekoälyn määritelmä tarkentuu mahdollisesti lähivuosina, kun eri tekoälysovelluksia hyödynnetään yhä laajemmissa mittakaavoissa.

4.2 Koneoppimisen ymmärtäminen

Kaikki koneoppimisen ratkaisut ovat tekoälyratkaisuja, mutta kaikki tekoälyratkaisut eivät pohjaudu koneoppimiseen. Koneoppimisesta on kuitenkin tullut tärkeä osa nykypäivän tekoälyä. Koneoppimisen avulla tekoälyä voidaan hyödyntää yhä monimutkaisempien asioiden ratkaisemiseen. Koneoppimisen avulla tekoäly pystyy oppimaan muun muassa virheistään ja tunnistamaan erilaisia kaavoja. Koneoppimista voidaan tehostaa esimerkiksi erilaisilla datalähteillä. (Artificial Intelligence (AI) vs. Machine Learning n.d.) Ymmärtääkseen koneoppimista vielä hieman paremmin on syytä tarkastella käsitteitä ohjattu oppiminen, ohjaamaton oppiminen, vahvistusoppiminen sekä syväoppiminen. Näiden lisäksi käsitteet keinotekoiset neuroverkot, luonnollisen kielen käsittely ja suuret kielimallit ovat olennaisia koneoppimisen osalta.

Ohjattu oppiminen (Supervised Learning, SL) tarkoittaa koneoppimisessa tilannetta, jossa tekoälyä opetetaan materiaalin avulla, minkä pohjalta tekoäly toteuttaa toimintojaan. Tekoäly ei siis itsenäisesti kehity vaan toteuttaa ennalta asetettuja sääntöjä datan pohjalta. (Toivonen 2023, 35; Tuominen ja Neittaanmäki 2019, 12.) Sekä Toivonen (2023) että Tuominen ja Neittaanmäki (2019) molemmat nostavat esimerkiksi ohjatusta ohjaamisesta sähköpostien automaattisen lajittelun eri kansioihin.

Ohjaamattoman oppimisen (Unsupervised Learning, UL) tapauksessa tekoälylle myös opetetaan malleja ja sääntöjä materiaalin avulla, mutta ohjaamattoman oppimisen tapauksessa tekoäly pyrkii löytämään samankaltaisuuksia aikaisemmin syötetyn ja uuden datan väliltä. Tämän perusteella tekoäly pystyy ryhmittelemään uutta dataa ilman, että se on täysin samanlaista kuin alkuperäinen. (Toivonen 2023, 35.)

Vahvistusoppiminen (Reinforcement Learning, RL) on jälleen askel kehittyneempään tapaan muodostaa tekoäly. Tekoäly kehittyy saamansa palautteen avulla ja pystyy näin oppimaan, mikä on oikein ja mikä ei (Tuominen ja Neittaanmäki 2019, 13–14). Esimerkiksi robotin pitää palauttaa pullo

koriin, mutta koreja on kaksi. Robotti valitsee toisen ja saa tästä käyttäjältä palautetta. Jos kori on väärä, robotti osaa ensi kerralla valita toisen korin. Jos taas palaute on positiivista, niin robotti palauttaa pullon samaan koriin, johon edellinenkin palautus tehtiin.

Syväoppiminen (Deep Learning, DL) on koneoppimisen seuraava, syvempi muoto. Syväoppiminen tarvitsee huomattavasti suurempia datamääriä, mutta lopputuloksena se myös kykenee entistä monimutkaisemman datan käsittelyyn ja tarkempiin lopputuloksiin (Haziqa 2023). Syväoppimisen perustana on neuroverkot. Neuroverkko on ollut käsitteenä olemassa jo kymmeniä vuosia, mutta ennen vuotta 2010 neuroverkot olivat hyvin alkeellisia. Vuoden 2010 jälkeen suuremman datan määrän hallinta on ottanut harppauksia eteenpäin, mikä on mahdollistanut myös neuroverkkojen kehittymisen yhä monimutkaisemmiksi. (Tuominen ja Neittaanmäki 2019, 2.)

Keinotekoiset neuroverkot (Artificial Neural Network, ANN) eli usein myös pelkästään neuroverkot, ovat tekoälyn hyödyntämisen syväoppimisen ydin. Nykypäivän laskentatehon kasvu ja datan hallinnan kehittyminen ovat mahdollistaneet entistä kehittyneempien neuroverkkojen rakentamisen. Neuroverkot koostuvat korkealla tasolla kolmesta eri kerroksesta, jotka ovat syöte-, piilo- ja ulostulokerros. Nämä kerrokset ottavat syötteen vastaan, käsittelevät sen neuronien avulla erilaisia laskelmia sekä päätelmiä hyväksikäyttäen ja välittävät lopputuloksen eteenpäin. (Tuominen ja Neittaanmäki 2019, 23–24.)

Luonnollisen kielen käsittely (Natural Language Processing, NLP) on yksi tekoälyn osajoukoista. Tämän avulla tekoäly pystyy vastaanottamaan käyttäjän kirjoittamaa tekstiä, prosessoimaan sitä ja tuottamaan käyttäjälle ymmärrettävän lopputuloksen. (Gruetzemacher 2022.) Luonnollisen kielen käsittely ja siitä jakautuneet mallit ovat olleet suuressa roolissa tekoälyn yleistymisessä viime vuosina, sillä juuri luonnollisen kielen käsittely on mahdollistanut sen, että tekoäly on tullut käytettävämmiksi suuremmille massoille.

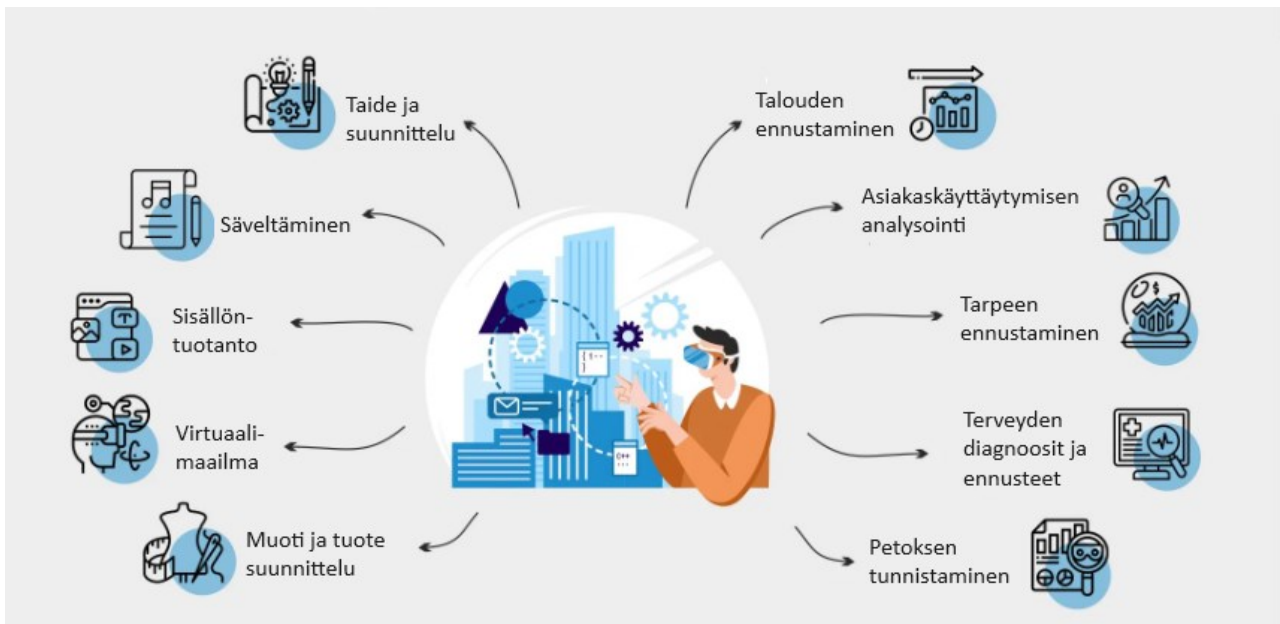
Suuri kielimalli (Large Language Model, LLM) on yksi luonnollisen kielen käyttämistä syväoppimisen algoritmeista, joka pystyy suorittamaan lukuisia eri luonnolliseen kieleen liittyviä tehtäviä. Tämä algoritmi käyttää transformer-mallia ja se koulutetaan suurella määrällä dataa, jonka avulla

se voi suorittaa useita eri tehtäviä, kuten esimerkiksi tekstin tunnistamista, kääntämistä ja generoimista. (What are Large Language Models (LLM) n.d.) Yksi tunnetuimmista suurta kielimallia hyödyntävistä tekoälyistä tänä päivänä on ChatGPT.

4.3 Prediktiivinen tekoäly

Prediktiivinen eli ennakoiva tekoäly on ollut olemassa jo vuosikymmeniä. Vasta viimeisten vuosikymmenten aikana sen potentiaalia on kuitenkin vasta pystytty kunnolla hyödyntämään. Tähän ovat vaikuttaneet laskentatehon kasvaminen sekä datan varastointimahdollisuudet. Eli niin sanottu massadatan (big data) hyödyntäminen on nostanut prediktiivisen tekoälyn hyödyntämisen seuraavalle tasolle. Tämän lisäksi kehittyneemmällä koneoppimisella on merkittävä rooli nykypäivän prediktiivisen tekoälyn käytössä.

Prediktiivinen tekoäly eroaa muista tekoälyistä siten, että se keskittyy nimenomaan analysoimaan sille ajettua informaatiota ja muodostamaan niiden pohjalta haluttuja ennusteita. Se ei siis voi tietää mitä seuraavaksi tapahtuu, mutta nimensä mukaisesti ennustaa tai ennakoii tulevaa historiadatan pohjalta. Prediktiivinen tekoäly käsittelee ja muodostaa päätelmiä koneoppimisen sekä statististen algoritmien pohjalta. Tekoälyä voi hyödyntää vähäiselläkin datalla, jos tarve on kohdistettu hyvin tarkkaan. Yleisimmät käyttötapaukset tämän tekoälyn käytössä ovat olleet jo pitkään yritysten käytössä. Tekoälyä hyödynnetään muun muassa rakentamaan ennusteita talouden kehityksestä ja asiakaskäyttäytymisestä. (Abdullahi 2024.) Kuviossa 3 on esitelty esimerkkejä potentiaalisista tekoälyn käyttötapauksista.

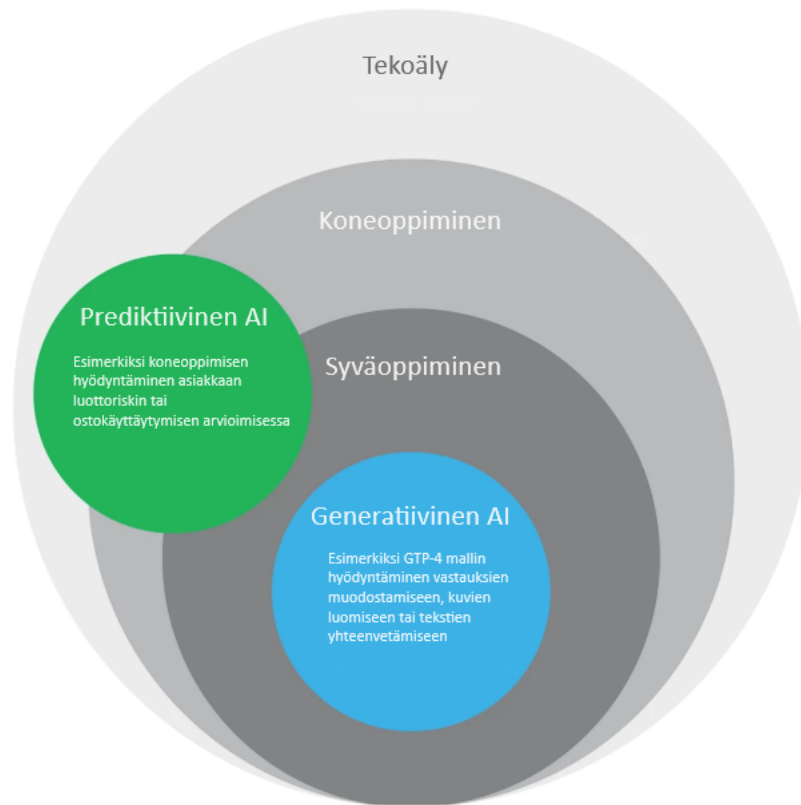


Kuvio 3. Esimerkkejä tekoälyn käyttötapauksista (Generative AI vs. Predictive AI: Unraveling the Distinctions and Applications n.d, muokattu)

4.4 Generatiivinen tekoäly

Tänä päivänä moni, joka puhuu tekoälystä tarkoittaa juuri generatiivista tekoälyä. Generatiivinen tekoäly on ollut jo pitkään olemassa, mutta sen käyttö on ollut hyvin vajaata suppeiden ominaisuuksien ja epätarkkuuksien takia. OpenAI-yrityksen julkaisema ChatGPT on nimenomaan generatiivinen tekoäly, joka hyödyntää muun muassa OpenAI:n rakentamaa suurta kielimallia. Tämän kielimallin ja tekoälyn avulla voidaan tuottaa käyttäjille hyvin realistisia vastauksia ja kuvia.

Generatiivisen tekoälyn tarkoituksena on nimenomaan generoida, eli luoda uutta sisältöä. Tekoäly voi luoda esimerkiksi uusia kuvia tai tekstiä saamansa syötteen perusteella. Luodut tuotokset perustuvat tekoälylle koulutettuun dataan, jota tekoäly voi koneoppimisen avulla hyödyntää tehokkaasti. Generatiivisen tekoälyn kouluttaminen vaatii huomattavasti enemmän dataa kuin prediktivisen tekoälyn, jotta se pystyy luomaan uutta käyttäjän syötteen sekä oppimansa tiedon perusteella. (Abdullahi 2024.) Kuviossa 4 on esitelty tiivistettynä prediktivisen ja generatiivisen tekoälyn erot.



Kuvio 4. Prediktiivisen ja generatiivisen tekoälyn erot (Riemer, S., Strauß, M., Rabener, E., Kwong Bickford, J., Hilbers, P., Kalra, N., Kapoor, A., King, J., Palumbo, S., Pardasani, N., Pauly, M., Rulf, K., Widowitz, M. 2023, muokattu)

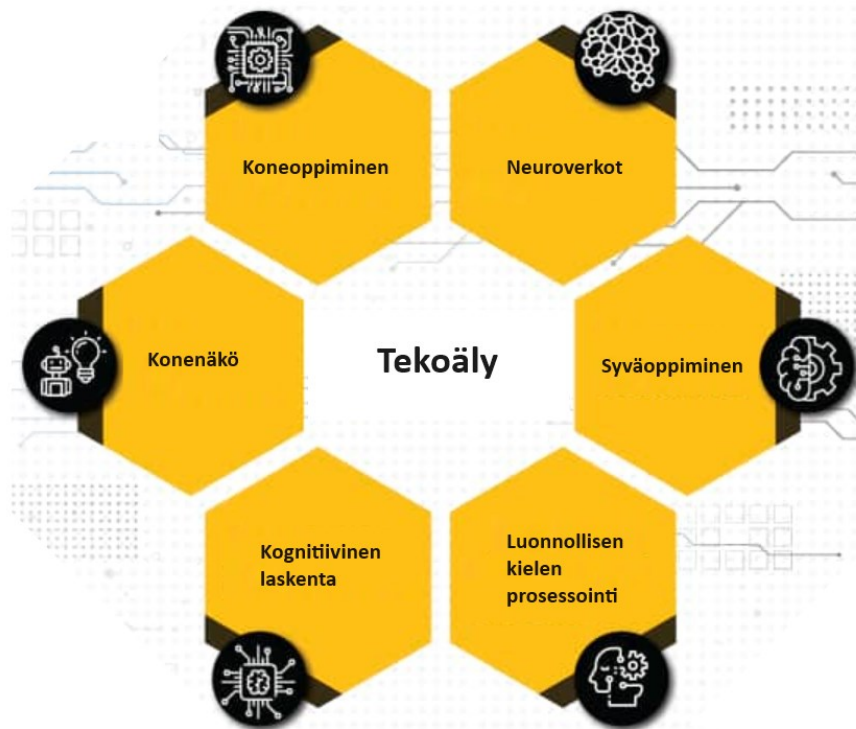
5 Tekoälyn korkean tason toimintamalli

5.1 Tekoälyn peruskomponentit

Erilaisia tekoälymalleja on useita, mutta tässä luvussa pyritään tarkastelemaan lähemmin prediktii- vistä ja generatiivista tekoälyä, sillä niiden toiminta perustuu lähtökohtaisesti vastaaviin tarpeisiin kuin mitä ohjelmoinnissa pystytään hyödyntämään.

Tekoälyn toiminta voidaan korkealla tasolla jakaa kuuteen eri kokonaisuuteen kuten kuviossa 5 on esitelty. Poikkeuksia näihin komponentteihin on myös olemassa, sillä tekoälymalleja on todella monia, mutta näiden kuuden avulla saadaan parannettua ymmärrystä yleisimpien tekoälyjen perustoiminnasta. Tämän luvun alaluvuissa tullaan käymään tarkemmin lävitse eri komponenttien

toiminta, pois lukien konenäkö, jonka ei odoteta merkittävästi auttavan ohjelmointitehtävissä.

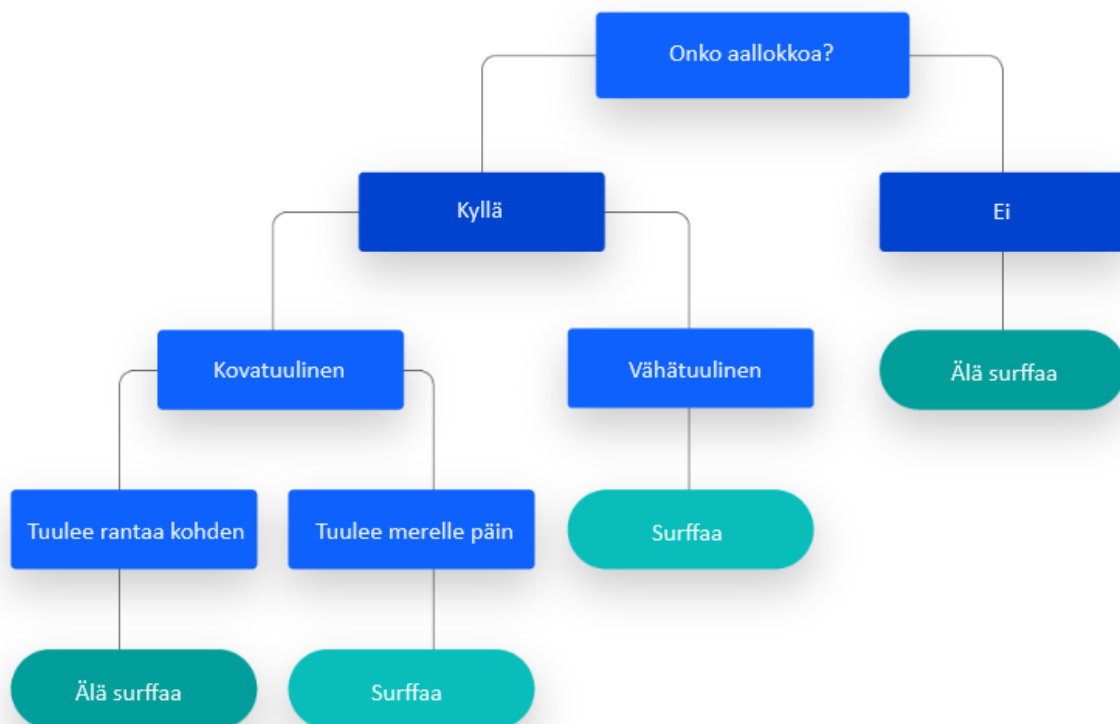


Kuvio 5. Tekoälyn peruskomponentit (How Does Artificial Intelligence Work? n.d, muokattu)

5.2 Koneoppimisen hyödyntäminen

Koneoppiminen on modernin tekoälyn perustoimintoja, vaikka tekoälyt kykenevät toimimaan myös ilman koneoppimista, mutta hyvin rajallisesti. Koneoppiminen nimensä mukaan tarkoittaa koneen oppimista, mikä tapahtuu syöttämällä sille koulutusaineistoa. Hyvät tekoälyt ovat kuitenkin erittäin kalliita opettaa, sillä niiden opettamiseen tarvitaan merkittäviä määriä dataa, etenkin generatiivisen tekoälyn tapauksessa. Koulutusaineiston laadukkuudella on myös merkittävä vastuu siitä, minkälaisia tuloksia tekoäly tuottaa. Heikkolaatuisen datan avulla voidaan saada aikaiseksi merkittäviä virheitä ja pahimmissa tapauksessa jopa syrjintään johtavia tapauksia, kuten lähihistoria on jo osoittanut. Tästä syystä koneoppimista käyttävien tekoälyjen testaaminen ja kalibrointi on hyvin tärkeässä roolissa. Tekoälyjen tarkkuutta voidaan yksinkertaisesti testata antamalla syöte ja validoimalla saatu lopputulos.

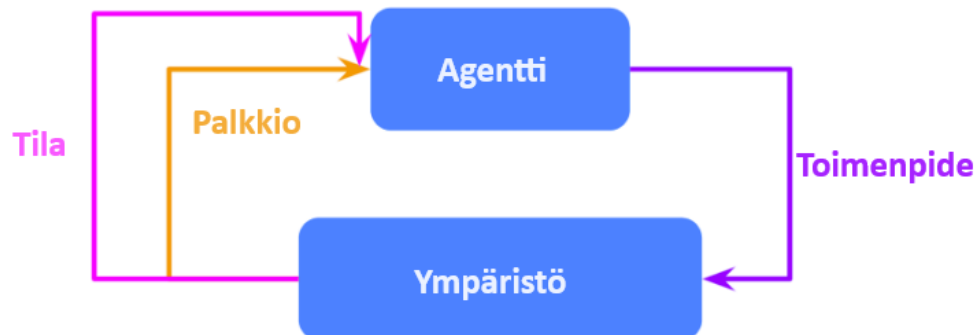
Yksinkertaisimpia esimerkkejä koneoppimisen käytöstä ovat niin sanotut päätöspuut. Päätöspuut muodostetaan ohjatulla oppimisella ja ne rakentuvat nollakohdasta, solmukohdista sekä mahdollisista lopputuloksista. Etuina päätöspuulle on sen tulkittavuus ja vähäinen datan tarve käyttöönottoa varten. Heikkoutena tälle kuitenkin on sen viriheherkkyys pienien poikkeuksien edessä sekä skaalautuminen. (What is decision tree? n.d.) Kuviossa 6 on esitelty hyvin yksinkertaistettu päätöspuun rakenne arkielämän päätöksenteosta.



Kuvio 6. Esimerkki koneoppimisen päätöspuusta (What is a decision tree? n.d, muokattu)

Ohjatun oppimisen haasteita on ratkaistu useammalla erilaisella tavalla, joista yksi tapa on vahvistusoppiminen, joka on yksi koneoppimisen osajoukoista. Vahvistusoppiminen ei aivan ole ohjattua, mutta ei myöskään ohjaamatonta oppimista vaan hieman sekä että. Tavoitteena on kouluttaa tekoäly kevyen perusdatan sekä palautteiden perusteella. Jokaisesta toiminnasta annetaan tekoälylle joko positiivinen tai negatiivinen palaute, minkä avulla tekoäly pystyy kehittämään syötteiden analysointia sekä prosessoituja tuloksia. (Keserer 2023.) Vahvistusoppimisen edut nousevat esiin etenkin, kun päätöksiä tarvitsee tehdä peräkkäin, eikä yhden päätöksen tekeminen enää riitä.

Vahvistusoppimista on syytä välttää, jos käytössä on vanha malli, jota voidaan hyödyntää tai väärillä päätöksillä voidaan aiheuttaa merkittäviä kustannuksia. (Agrawal & Wu 2021.) Kuviossa 7 on esitelty yksinkertaistettu vahvistusoppimisen prosessi.



Kuvio 7. Vahvistusoppimisen prosessi (Keserer 2023, muokattu)

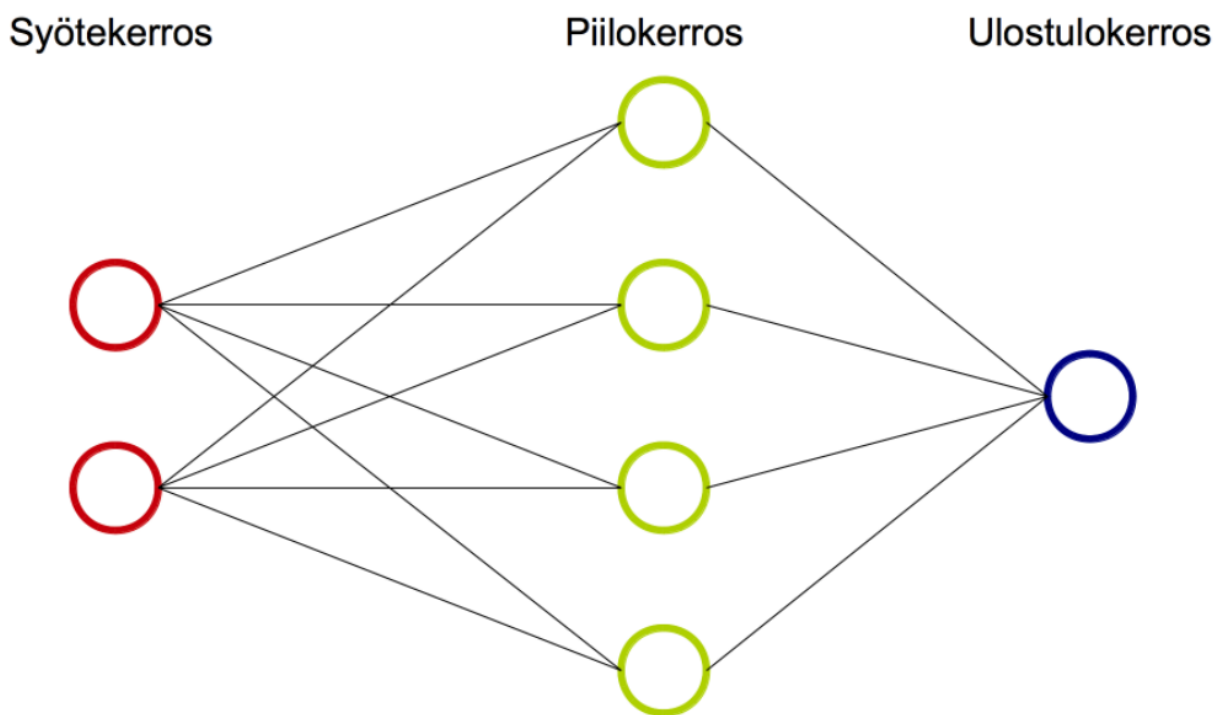
Toisena yleisesti käytössä olevana koneoppimisen vaihtoehtona on syväoppiminen, jonka käyttö on kehittyneiden keinotekoisien neuroverkkojen myötä yleistynyt merkittävästi. Nykypäivänä syväoppimisen mallit käyttävät lähes poikkeuksetta keinotekoisia neuroverkkoja, jotka perustuvat useisiin eri tasoihin. Perustoimintaperiaatteena toimii päätöspuun tapainen malli, jossa syötteen prosessointi siirtää tiedon noodilta toiselle, mutta merkittävänä erona toimii mahdollisuus siirtyä rajattomasti eri noodien välillä toisin kuin päätöspuussa. Syväverkkojen käyttäminen vaatii valtavasti laskentatehoa, sillä neuroverkkojen monimutkaisuus ja opetettavan datan määrä on merkittävä (Keserer 2023). Generatiiviset tekoälyt hyödyntävät syväoppimista, sillä toimiakseen ne vaativat suuren määrän dataa, minkä käsittelemiseen juuri syväoppiminen pystyy. Syväoppimista käytetään etenkin optimoituihin tehtäviin, kuten kasvojen tai äänentunnistuksen järjestelmiin tai itseohjautuvien autojen ohjaamiseen. (Keserer 2023.) Koska keinotekoiset neuroverkot ovat olennainen osa syväoppimisen toimintaa, tullaan seuraavassa luvussa tarkastelemaan neuroverkkojen toimintaa tarkemmin.

5.3 Keinotekoisien neuroverkkojen toiminta

Keinotekoisella neuroverkolla (jatkossa neuroverkko) pyritään imitoimaan ihmisten aivojen kaltaista käyttäytymistä. Neuroverkoista voidaan tehdä todella monimutkaisia, minkä takia tekoäly

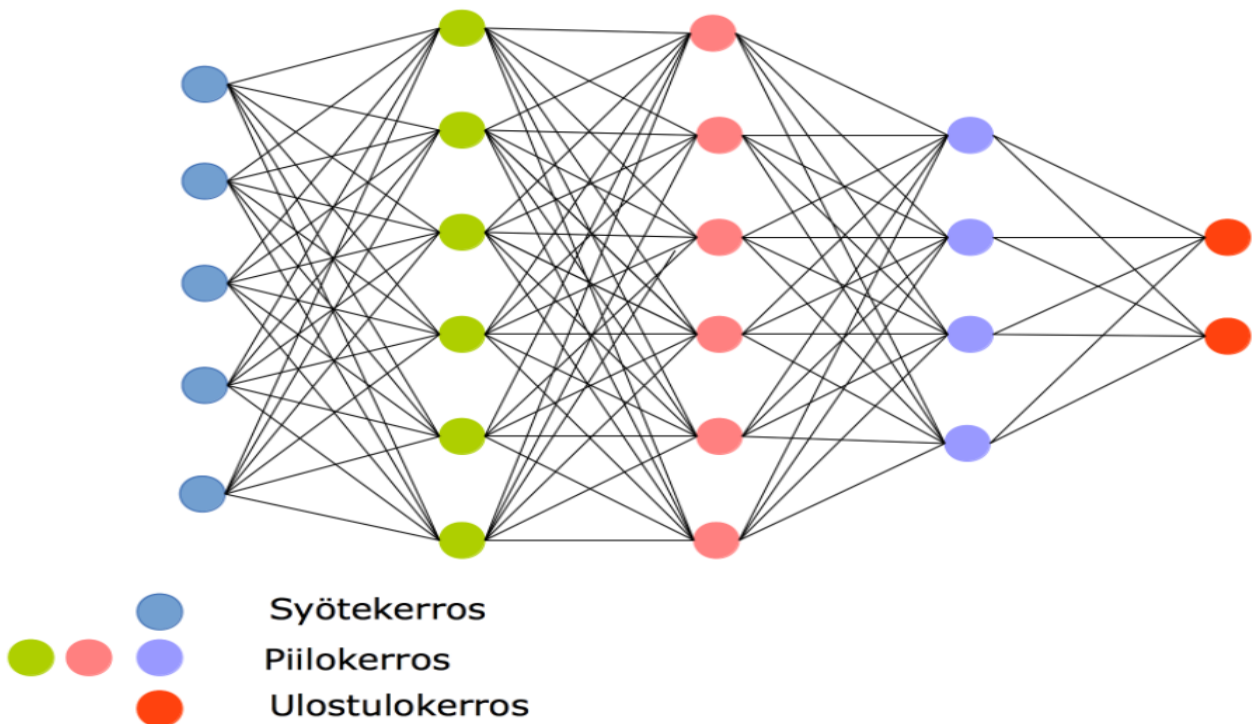
kykenee suorittamaan yhä haastavampia tehtäviä. Jokainen neuroverkko koostuu syöte- ja ulostulokerroksista, sekä vähintään yhdestä piilokerroksesta. Nämä kerrokset muodostuvat neuroneista. (Tuominen ja Neittaanmäki 2019, 24.)

Yksinkertaisimmillaan neuroverkko voi muodostua vain muutamista neuroneista (ks. kuvio 8), joiden perusteella dataa käsitellään ja tuotetaan. Neuroverkot voivat kuitenkin olla hyvin monimutkaisia (ks. kuvio 9) ja sisältää jopa tuhansia piilokerroksia datan käsittelemisen. Tällaisten neuroverkkojen rakentaminen on haastavaa ja optimoiminen aikaa vievää.



Kuvio 8. Yksinkertainen neuroverkko (Tuominen ja Neittaanmäki 2019, 24)

Syvä neuroverkko



Kuvio 9. Syvä neuroverkko useammalla piilokerroksella (Tuominen ja Neittaanmäki 2019, 25)

Neuroverkot koostuvat kerrosten lisäksi neuronien välisistä yhteyksistä, jotka sisältävät painoarvoja ja bias-arvoja. Näiden lisäksi jokainen neuroni sisältää summa- ja aktivaatiofunktion. Neuroverkkojen toimintaa voidaan säätää nimenomaa painoarvojen ja bias-arvojen avulla, joita hyödynnetään, kun arvoja lasketaan summafunktiossa, jonka arvo siirtyy neuronin aktivointifunktioon. (What Is the Necessity of Bias in Neural Networks? n.d.) Aktivointifunktioita on ennalta olemassa eri käyttötapauksia varten.

Kuten kuviosta 9 voidaan havaita, kasvaa potentiaalisten vaihtoehtojen määrä moninkertaisesti jo muutaman neuronin tai piilokerroksen lisäyksellä. Myös syötteiden määrä vaikuttaa lopputuloksen tarkkuuteen. Niin sanottu ylisovittaminen on yleinen ongelma neuroverkoille, jolloin malli alkaa kiinnittämään liikaa huomiota poikkeuksiin ja epäkohtiin, mikä aiheuttaa ongelmia, kun mallin pitäisi soveltaa oppimaansa uusien tilanteiden kanssa. Alisovittamisessa malli on liian yksinkertainen, jolloin datan käsittely jää vajavaiseksi eikä malli opi tarvittavia asioita riittävällä tasolla. Tällöin mallin hyödyntäminen on heikkoa.

5.4 Kielimallien käyttäminen

Luonnollisen kielen käsittely on yksi suurimmista syistä tämänhetkiseen laajaan kiinnostukseen tekoälyn suhteen. Luonnollisen kielen käsittelyn ja siitä periytyneiden suurten kielimallien avulla ihmisen ja tekoälyn välinen kommunikointi on arkipäiväistynyt, kun molemmat osapuolet ymmärtävät toistaan samalla kielellä. Tästä kertoo myös markkinatutkimus, jonka perusteella luonnollisen kielen käsittely on melkein 30 miljardin arvoista liiketoimintaa vuonna 2024 ja liiketoiminnan nähdään jopa kasvavan yli 150 miljardiin vuoteen 2032 mennessä (Market Research Report 2024). Esimerkkejä tämän hetken luonnollisen kielen käsittelyä hyödyntävistä toteutuksista ovat erilaiset analyysi- ja käännöstyökalut (ks. kuvio 10).

Erialaisten kielimallien avulla tekoäly pystyy siis tulkitsemaan ja tuottamaan tekstiä. Myös puheenymmärrys on kehittynyt jo hyvin pitkälle nykypäivän tekoälysovelluksissa. Kielimallien avulla onnistuu myös dokumenttien analysointi sekä erilaiset tiedon poimimisen automaatiot. Tunnetuimmat kielimallit tällä hetkellä ovat OpenAI:n tuottama GPT, Claude sekä Googlen kehittämä BERT. Suuria kielimalleja on olemassa jo useita ja usein taustalla on isompi yritys, jonka on mahdollista rahoittaa massiivisten datamäärien käyttö tekoälyn kielimallien koulutukseen.



Kuvio 10. Luonnollisen kielen käyttötapauksia (Natural Language Processing (NLP) and its Role in Advancing Conversational Artificial Intelligence 2023, muokattu)

Kielimallit ovat siis hyvin laajoja syväoppimisen malleja, jotka on koulutettu todella laajoilla aineistoilla. Näiden aineistojen avulla ne pystyvät käymään sujuvia keskusteluja. (What are Large Language Models (LLM) n.d.) Koulutusaineistot saattavat sisältää jopa miljardeja sanoja, joiden avulla ne oppivat tunnistamaan kielen rakenteet, sanastot ja merkitykset eri tilanteissa. Toiminta perustuu ennustamiseen, joka pohjautuu koulutettuun aineistoon. Tarpeeksi kattavalla aineistolla ennustettavuus paranee ja virheentodennäköisyydet pienenevät. (Numminen 2023.)

Koulutusaineistot koostuvat useassa tapauksessa netissä julkisesti jaossa olevasta materiaalista, kuten e-kirjoista, artikkeleista tai verkkosivuilla olevista teksteistä. Syviä neuroverkkoja hyödyntämällä kielimallit oppivat tehokkaasti aineistoja. Neuroverkot on säädetty jopa miljardeilla eri parametreilla vastaamaan kielen ominaispiirteitä. Koulutusvaiheessa malli tekee virheitä, mutta parametreja säätämällä pystytään parantamaan ennustettavuutta tuleville kerroille. Malleille voidaan myös syöttää yksityiskohtaisempaa dataa esimerkiksi tietyistä toimialasta, jonka avulla se kykenee tehokkaammin vastaamaan juuri kyseistä toimialaa koskeviin kysymyksiin. Aineiston omaksumisen jälkeen kielimalli pystyy ymmärtämään tekstiä paremmin sekä tuottamaan sisältöä koulutetun aineiston pohjalta. (Numminen 2023.)

Kielimalleilla on myös oleellinen osa puheenymmärryksessä. Automaattinen puheentunnistusjärjestelmä prosessoi puhuttua audiota ja tunnistaa sieltä sanoja analysoimalla signaaleja ja tunnistamalla puheen eri komponentteja, kuten sanoja ja lauseita. Tämän jälkeen data jaetaan pieniin osiin, jonka jälkeen niitä verrataan akustisiin malleihin, joista pyritään tunnistamaan todennäköisesti puhutut sanat. Tämän jälkeen puheesta tekstiksi muutettua dataa analysoidaan kielimallien avulla. (Inside Speech to Note: How AI Understands Your Voice and Converts it to Text 2023.)

Yhtenä kielimallien suurimpana haasteena on edelleen aidosti ymmärtää tekstiä. Tekstin ymmärrys ja tuottaminen perustuu puhtaasti tilastollisiin todennäköisyyksiin, joiden pohjalta käyttäjäkin saa vastauksensa. Koneilta puuttuu myös niin sanottu maalaisjärki, minkä takia osa syötteistä ei välity kielimallille samalla tavalla kuin ihmiselle. Sarkasmi on toinen hyvä esimerkki kielimallien puutteellisesta ymmärryksestä. Myöskään tekstin tuottaminen loogisesti ei aina tarkoita, että tekstin sisältö olisi oikeassa ja loogista, vaikka se käyttäjälle näin välittyisikin. (Numminen 2023.)

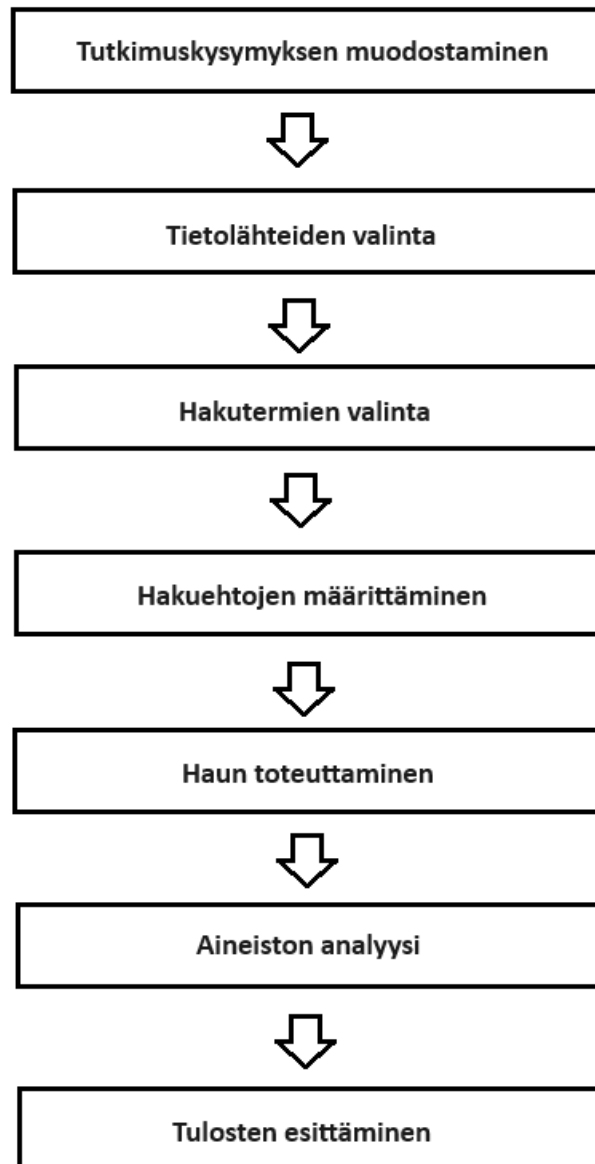
6 Kirjallisuuskatsauksen toteutus

6.1 Integratiivinen kirjallisuuskatsaus tutkimusmenetelmänä

Integratiivinen kirjallisuuskatsaus on yksi monista eri kirjallisuuskatsauksen päätyypeistä. Muut päätyypit ovat narratiivinen ja systemaattinen katsaus sekä meta-analyysi. Integratiivinen kirjallisuuskatsaus sijoittuu osittain narratiivisen ja systemaattisen katsauksen puoliväliin hyödyntäen osia molemmista tyypeistä. Narratiivisen katsauksen ollessa hyvin kuvaileva ja jossa tiedonhaku pystytään toteuttamaan melko ketterästi, pyrkii systemaattinen katsaus määrittämään tarkemmin hakuprosessin. Haasteena systemaattisessa katsauksessa on tietynlainen jäykkyys, kun narratiivinen taas antaa hieman enemmän vapauksia tekijälleen ilmaisuiden suhteen. Integratiivisessa katsauksessa toteutetaan systemaattisen katsauksen mukaista järjestelmällistä tiedonhakuja, mutta toteutetaan tulosten käsittelyä hieman vapaammin narratiivisen katsauksen mukaisesti. (Vilka 2023, 16–17.)

Integratiivisen kirjallisuuskatsauksen edellytyksenä on kirjoittajaltaan jo hyvä tuntemus aiheesta, sillä vaikka haut suoritetaankin ennakkoon suunnitellusti ja lähteitä tarkastellaan kriittisesti, niin tarkoituksena ei ole kuitenkaan syventyä lähteisiin yhtä syvästi kuin systemaattisessa katsauksessa. Myös määrällisten ja laadullisten tutkimusten hyödyntäminen samassa integratiivisessa katsauksessa on sallittua ja empiiristen tutkimusten sijaan voidaan käyttää myös teoreettisia tutkimuksia. Näiden avulla voidaan muodostaa parempi kokonaiskuva aiheesta. Integratiivinen katsaus sopii myös paremmin tutkimuksiin, joista ei välttämättä löydy julkaisuja ja vertaisarvioituja raportteja tai muuta kirjallisuutta. (Vilka 2023, 21–22.) Vaikka tekoälyä on tutkittu hyvin pitkään ja aiheesta löytyy varsin kattavasti kirjallisuutta, on kuitenkin syytä pitää mielessä, että tekoälyn kehitys on viime aikoina ollut hyvin vauhdikasta, jolloin liian vanhoihin tutkimuksiin ja raportteihin ei tule luottaa, jos halutaan löytää luotettavat vastaukset opinnäytetyön tutkimuskysymyksiin.

Kirjallisuuskatsaus seuraa Finkin (2005) määrittelemää prosessia soveltuvin osin (ks. kuvio 11). Tutkimuskysymykset ovat aikaisemmin esiteltyinä luvussa 3.2. Näihin kysymyksiin pyritään nyt hakemaan vastauksia prosessin mukaisesti. Seuraavat tämän luvun aliluvut käyvät lävitse aineiston hankintaan ja sen analysointiin liittyviä vaiheita.



Kuvio 11. Kirjallisuuskatsauksen vaiheet (Fink 2005, 54)

6.2 Aineiston hankinta

Tutkimuskysymykset muodostivat pohjan eri tietolähteiden valinnalle. Tietokannoista haluttiin keskittyä niihin kantoihin, jotka sisältävät mahdollisimman kattavasti teknillistä aineistoa ja erityisesti tietotekniikkaan liittyen. Alustava listaus potentiaalisista tietokannoista tehtiin Jamkin tiedonhaun ohjeissa olevan tietokantalistauksen avulla. Tietokantojen valintaehtona yllä mainitun ehdon lisäksi olivat:

- Kansainvälisyys
- Riittävät suodatinvaihtoehdot

- Pilottihaun tuloksien määrä ja laatu

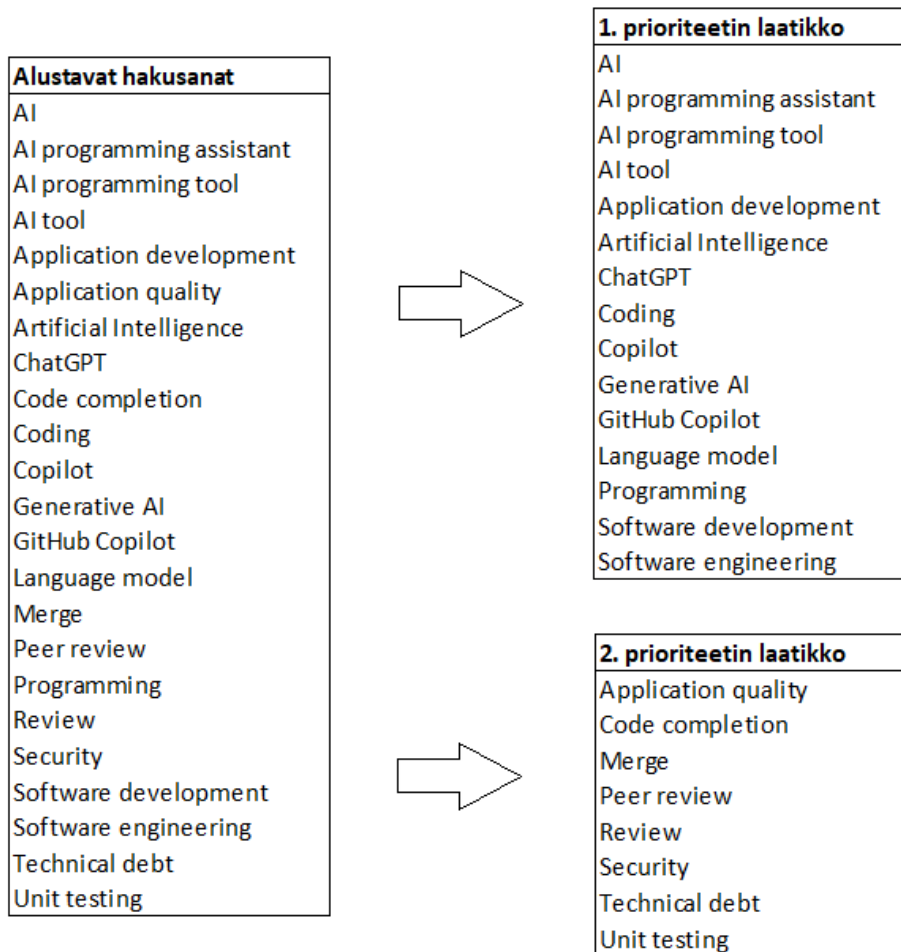
Yllä asetettujen ehtojen takia tutkimuksessa päädyttiin käyttämään neljää eri tietokantaa. Tietokannat olivat ScienceDirect, IEEE Xplore, ProQuest sekä Google Scholar. Kolme ensimmäistä tietokantahakua toteutettiin samoin hakuehdoin, mutta Google Scholarin tapauksessa haku toteutettiin manuaalisesti ja tiukemmilla suodattimilla.

Prosessin mukaisesti (ks. kuvio 11) seuraavaksi aloitettiin määrittelemään tarkemmat hakusanat. Hakusanojen pohjana käytettiin tutkimuskysymyksiä, joiden perusteella lopulliset hakusanat muodostettiin. Hakusanojen muodostaminen toteutettiin kolmessa eri vaiheessa:

1. Muodostettiin alustava lista mahdollisista hakusanoista tutkimuskysymysten pohjalta.
2. Hakusanoilla pilotoitiin hakuja valittuihin tietokantoihin.
3. Lopullisen hakusanalistauksen muodostaminen pilotoinnin tulosten perusteella.

Ensimmäisen vaiheen alustavassa listauksessa hakusanoja valittiin 23 kappaletta. Hakusanojen ennakointiin jo tässä vaiheessa tuottavan merkittävästi hakutuloksia, minkä takia hakusanat jaettiin kahteen eri prioriteettilaatikkoon. Jos ensimmäisen prioriteetin laatikon hakusanat eivät tuottaisi tarpeeksi hakutuloksia, niin 2. prioriteetin laatikosta lisättäisiin hakusanoja mukaan tietokantoihin tehtäviin hakuihin. Ensimmäisen vaiheen hakusanalistaukset ovat nähtävissä taulukossa 3.

Taulukko 3. Ensimmäisen vaiheen hakusanat



Toisen vaiheen pilotointi osoittautui erittäin tärkeäksi osaksi aineiston hakua, sillä se paljasti hyvin nopeasti haasteet joidenkin hakusanojen kanssa. Esimerkiksi hakusanat "AI" tai "programming" osoittautuivat hyvin generisiksi ja niiden tuottamat hakutulosten määrät olivat valtavat jopa erittäin tiukkojen suodattimien kanssa. Tästä syystä, lopullinen hakusanalista supistettiin vielä pienemmäksi, jotta tietokantoihin tehdyt haut saatiin paremmin kohdistettua tuottamaan hakutuloksia, jotka vastasivat paremmin tutkimuskysymyksiin. Pilottihaussa mukana olivat myös suomenkieliset käännökset hakusanoista, mutta tulokset jäivät hyvin vähäisiksi, minkä takia lopullisessa haussa käytettiin vain englanninkielisiä hakusanoja. Lopullisessa haussa käytettiin yhdeksää eri hakusanaa ja nämä ovat esiteltyinä taulukossa 4.

Taulukko 4. Tietokantahakujen lopulliset hakusanat

Hakusanat
AI programming assistant
AI programming tool
Artificial Intelligence
ChatGPT
Generative AI
GitHub Copilot
Language model
Software development
Software engineering

Hakusanoista muodostettiin hakulauseke boolean-operaattorien avulla. Hakulausekkeessa käytettiin boolean-operaattoreista sekä AND että OR-operaattoreita. Näiden operaattoreiden sekä oikeanlaisen jäsentelyn ja ryhmittelemisen avulla pystyttiin muodostamaan lauseke, minkä avulla jokaiseen kantaan pystyttiin toteuttamaan kaksi eri hakua. Nämä haut tuottivat katsaukseen tarvittavan aineiston. Hakulausekkeet olivat seuraavanlaiset:

1. ("Software development" OR "Software engineering") AND ("Artificial intelligence" OR "Generative AI" OR "Language models" OR ChatGPT OR "AI programming assistant" OR "AI programming tool")
2. GitHub Copilot

GitHub Copilot on alustavan tutkimuksen mukaan yksi käytetyimmistä tekoälytyökaluista kehittäjien keskuudessa, minkä takia sitä on syytä tarkastella tarkemmin. Kyseinen tuote on valmiiksi tekoälypohjainen, minkä takia se ei vaadi kattavampaa hakulauseketta.

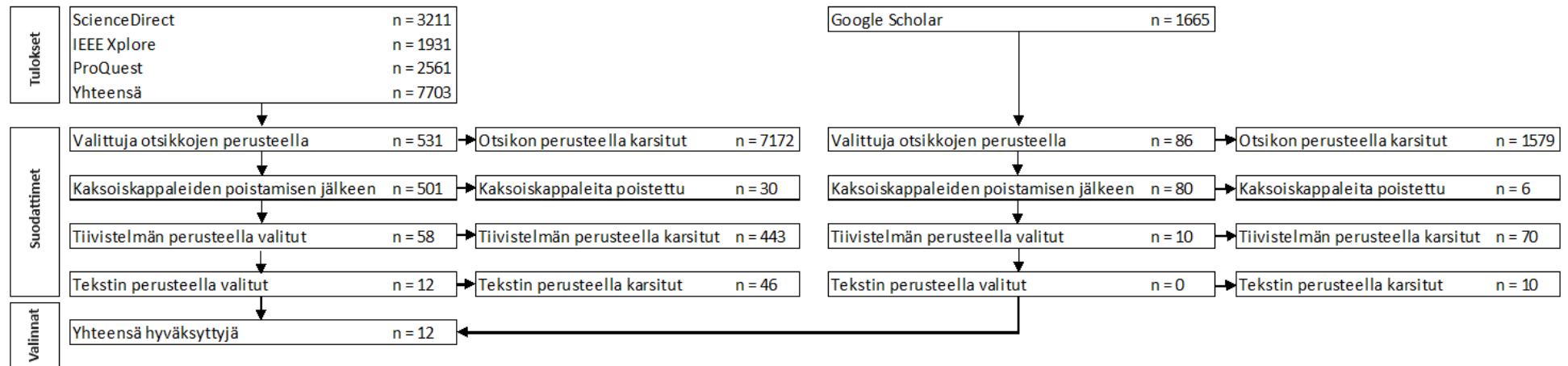
Hakuehdot määriteltiin siten, että aineiston tuli olla julkaistu vuosien 2023 tai 2024 aikana. Julkaisun kielen tuli olla englanti. Lisäksi tekstin tuli olla kokonaan saatavilla. Muita hakuetoja ei vaadittu, sillä hauissa arvostettiin ennen kaikkea julkaisun tuoreutta, koska tekoäly tieteenalana kehittyy jatkuvasti. Poikkeuksen tähän teki Google Scholariin tehty manuaalinen haku, joka rajattiin koskemaan vain vuoden 2024 vertaisarvioituja hakutuloksia.

6.3 Haun tulokset ja valinnat

Annetuilla hakuehdoilla ja tietokannoilla saatiin 9368 hakutulosta eli tuloksia saatiin hyvin kattavasti. Hakutulosten määrän takia jokainen tulos analysoitiin ensimmäiseksi vain otsikon perusteella. Tämän avulla hakutuloksia saatiin suodatettua 617 tutkimukseen eli alle kymmeneen prosenttiin alkuperäisistä hakutuloksista. Seuraavaksi luettiin otsikoiden avulla suodatettujen hakutulosten tiivistelmät lävitse ja poistettiin eri tietokannoista tulleet kaksoiskappaleet. Näiden suodatusten jälkeen jäljellä oli enää vain 68 eri tulosta. Viimeisessä vaiheessa varmistettiin, että koko teksti on saatavilla ja luettiin se. Lukemisella varmistettiin, että aineisto vastaa tutkimuskysymyksiin. Tämän lisäksi aineistoille toteutettiin laadunarviointi.

Hakutuloksien kuvaamista on havainnollistettu Prisma 2020 -diagrammin avulla (Page, McKenzie, Bossuyt, Boutron, Hoffmann, Mulrow, Shamseer, Tetzlaff, Akl, Brennan, Chou, Glanville, Grimshaw, Hróbjartsson, Lalu, Li, Loder, Mayo-Wilson, McDonald, McGuinness, Stewart, Thomas, Tricco, Welch, Whiting & Moher 2021). Diagrammi on esitetty kuviossa 12 ja sen perusteella lukijan on mahdollista seurata haku- ja suodatusvaiheiden etenemistä.

Lopullisessa tietokantahakujen aineistossa on 12 eri tutkimusta tai artikkelia. Google Scholarin tuloksista ei päädytty ottamaan yhtään aineistoa mukaan, sillä ne eivät täyttäneet edellä mainittuja ehtoja tai olivat kaksoiskappaleita. Näiden lisäksi kirjallisuuskatsauksen aineistoon valittiin kolme eri tutkimusta tietokantahakujen ulkopuolelta. Kaksi näistä tutkimuksista ovat alan johtavien yritysten tuottamia (Stack Overflow sekä Gartner). Kolmannen yrityksen (LinearB) tutkimus löydettiin webinaarin kautta. Integratiivisessa kirjallisuuskatsauksessa on mahdollista ottaa tutkimuksia mukaan hieman kattavammin ja tästä syystä on myös hyvä tarkastella yritysten tuottamaa materiaalia ja pyrkiä muodostamaan synteesejä tutkimusten kanssa. Lopullisessa kirjallisuuskatsauksessa on siis mukana yhteensä 15 eri tutkimusta tai artikkelia. Kirjallisuuskatsauksen aineiston lähteet on selkeyden vuoksi sijoitettu liitteelle 1.



Kuvio 12. Aineiston valinta PRISMA 2020 -diagrammin mukaisesti (Page ym. 2021)

6.4 Aineiston laadun arviointi

Jokaisen kirjallisuuskatsauksen tulee toteuttaa laadunarviointi valituille aineistolle. Tällä pyritään estämään yhtä kirjallisuuskatsauksen yleisimmästä ongelmasta, jossa arviointi on jätetty tekemättä. Arvioinnin perusteella pystytään muun muassa määrittämään, mitkä aineistot ovat tutkimuksen osalta olennaisimmat ja luotettavimmat sekä mitkä kannattaa jättää kokonaan pois. Tämän avulla voidaan varmistaa tutkimuksen luotettavuus kokonaisvaltaisesti. (Vilka 2023, 74.) Tämän kirjallisuuskatsauksen laatuanalyysi mukaan valituille aineistoille löytyy liitteestä kaksi.

Integratiivisen kirjallisuuden lähdeaineiston arviointi on hieman haastavampaa, koska mukaan voidaan ottaa huomattavasti laajemmin erilaista lähdemateriaalia kuin esimerkiksi systemaattisessa kirjallisuuskatsauksessa. Analyysiä tehdessä on kuitenkin pyritty varmistamaan, että lähdeaineisto on riittävän laadukasta katsausta varten. Liian epämääräiset tai puutteelliset aineistot pudotettiin pois ennen lopullista valintaa. Liitteestä 2 voi tarkastella jokaisen aineiston analyysin ja valintaperusteet laadun näkökulmasta katsottuna. Lähtökohtaisesti tutkimusta voidaan pitää kirjallisuuskatsauksen kannalta laadukkaana silloin, kun se vastaa hyvin tutkimuskysymyksiin sekä tutkimusasetelma on kuvattu tarpeeksi selvästi. Tutkimuksen tuoreudella ei analyysivaiheessa ollut enää niin suurta painoarvoa, koska tämä oli ollut yksi hakuja rajaavista ehdoista. Analyysia tehdessä ei myöskään otettu kantaa tutkimuksen tuloksiin vaan keskityttiin varmistamaan aineiston laadukkuus. Arviointi suoritettiin numeerisesti perustellen. Numero yksi tarkoittaa välttävää ja numero viisi erinomaista arvosanaa aineistolle. Pienin mukaan katsaukseen hyväksytyt kokonaisarvosana oli kolme. Aineiston laadunarvioinnissa tarkastettiin seuraavat asiat:

- Julkaisun tarkoitus
- Tutkimusmenetelmien kuvaus
- Viitekehys
- Tutkimuksen johtopäätösten selkeys
- Puolueettomuus
- Alkuperäisyys ja lähteiden luotettavuus

6.5 Aineiston analysointi

Aineiston analyysissä oli tarkoitus tiivistää kirjallisuuskatsauksesta saatuja havaintoja siten, että ne vastaavat asetettuihin tutkimuskysymyksiin. Analyysimenetelmäksi valittiin temaattinen analyysi, jossa aineistoa käsitellään koodaamalla se eri alateemoihin. Alateemat on muodostettu tutkimuskysymysten pohjalta. Näihin eri teemoihin lisättiin muistiinpanoja havaintojen pohjalta samalla kun aineistoa tarkasteltiin. Alateemat olivat toteuttaminen, dokumentointi, testaaminen ja tietoturva.

Toteuttamisella tarkoitetaan itse koodin tuottamista tekoälytyökaluja hyödyntäen. Katsauksen aikana pyrittiin löytämään tapoja tuottaa koodia tehokkaammin tekoälyn avulla, sekä selvittää, voiko tekoäly jotenkin toimia kehittäjän tukena ongelmatilanteissa.

Dokumentointi kohdassa on tarkoitus keskittyä kooditason dokumentointiin tai mahdollisesti koodista generoitaviin dokumentteihin, joissa voidaan hyödyntää tekoälyä. Tarkoituksena ei ole huomioida kaikkea projektin aikana toteutuvan dokumentoinnin tuottamista tekoälyn avulla.

Testaaminen ei sisällä järjestelmän kokonaisvaltaista testaamista vaan ainoastaan kooditasolla tehtävät yksikkötestaukset. Kirjallisuuskatsauksen tuotoksien läpikäynnissä on siis tarkoitus selvittää, voidaanko myös yksikkötesteihin luotua koodin generointia tehostaa.

Tietoturva on nykypäivänä syytä ottaa huomioon kaikessa tekemisessä. Katsauksen aikana keskitytään tarkastelemaan ainoastaan sitä, miten tekoäly ottaa huomioon kooditason tietoturvan koodin generointi tehtävissä.

Näiden alateemojen lisäksi muodostettiin alateema nimeltä ”sekalaiset” siltä varalta, että katsauksen aikana löydettäisiin hyödyllistä tietoa, mikä vastaisi tutkimuskysymykseen, mutta ei selkeästi kuuluisi mihinkään edellä mainituista alateemoista. Myös näiden havaintojen dokumentointi auttoi eri synteiesien muodostamisessa tuloksien käsittelyn aikana.

Lopullisissa tuloksissa haettiin synteesejä eri lähteiden väliltä, joista muodostettiin tulokset tutkimukselle. Tulosten analysoinnissa edettiin induktiivisesti eli aineistolähtöisesti. Tutkittavan alueen

tunteminen antaa paremmat edellytykset tuloksien analysointiin ja päätelmien tekemiseen (Vilka 2023, 70).

7 Tulokset

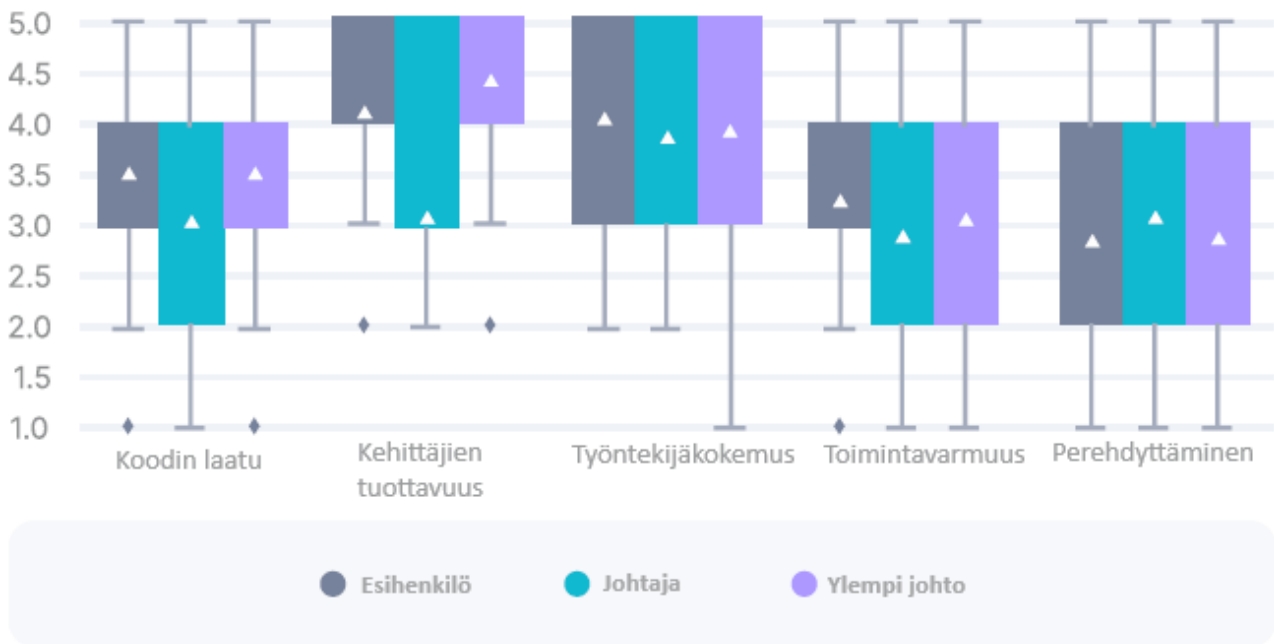
Kirjallisuuskatsauksen aineistosta eli läpikäydyistä tutkimuksista muodostettiin vastaukset tutkimukselle asetettuihin tutkimuskysymyksiin. Tuloksissa käsiteltiin molempia tutkimuskysymyksiä rinnan, sillä aineiston läpikäynnin perusteella hyvin moni tekoälyn ominaisuus saattaa koskea molempia kysymyksiä. Aineistoja tarkasteltiin usealta eri näkökulmalta, mutta jokaisessa pääpaino pysyi ohjelmointiin liittyvissä tehtävissä. Tutkimuskysymykset olivat seuraavat:

1. Voidaanko tekoälyn avulla tehostaa ohjelmistokehitystä?
2. Voidaanko tekoälyn avulla parantaa ohjelmistokehityksen laadukkuutta?

7.1 Tekoälyn kypsyys ohjelmoinnissa

Tekoälyn hyödyntäminen ohjelmointiin liittyvissä tehtävissä on kehittynyt merkittävästi viime vuosina ja siitä onkin OpenAI:n julkaiseman ChatGPT:n myötä tullut yksi historian odotetuimmista teknologioista ohjelmistokehityksessä. Jo pelkästään ChatGPT keräsi yli 100 miljoonaa kävijää ensimmäisen kahden kuukauden aikana ja tämän myötä ChatGPT:stä tuli kaikkien aikojen nopeitten kasvanut tekoälytyökalu (Kuhail, Mathew, Khalil, Berengueres & Shah 2024).

Batchun, Walshin, Scheibmeirin, Nairin, Bhatin ja Egiazarovin (2024) tekemän analyysin mukaan tekoälytyökalujen avulla ohjelmoijat voivat tuottaa koodia nopeammin, mikä tehostaa kannattavuutta. Samalla työkalut parantavat työntekijäkokemusta, kun yksinkertaiset tehtävät voidaan toteuttaa tekoälyn avulla. Kuviosta 13 käy ilmi, miten tekoälyn arvellaan parantavan toimintaa eri alueilla. Kuvissa numero 1 tarkoittaa heikko parannusta ja numero 5 selkeää parannusta.

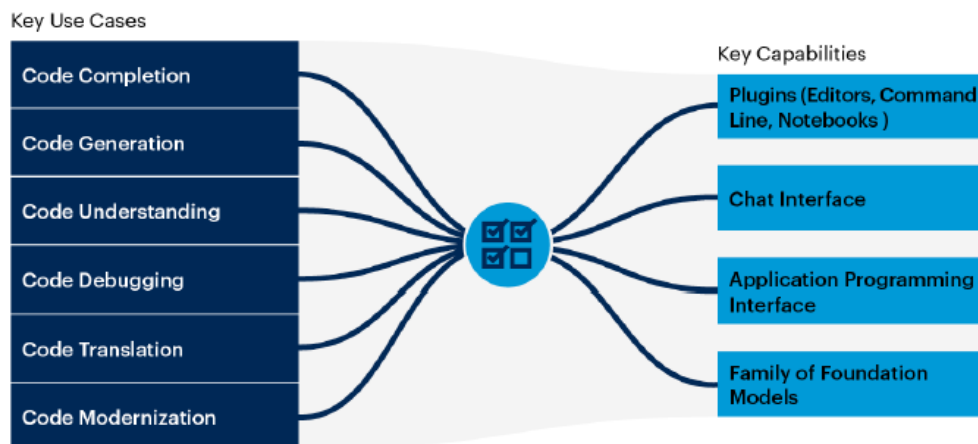


Kuvio 13. Tekoälyn avulla parantaminen eri osa-alueilla (Measuring Impact: The GenAI Code Report 2024, muokattu)

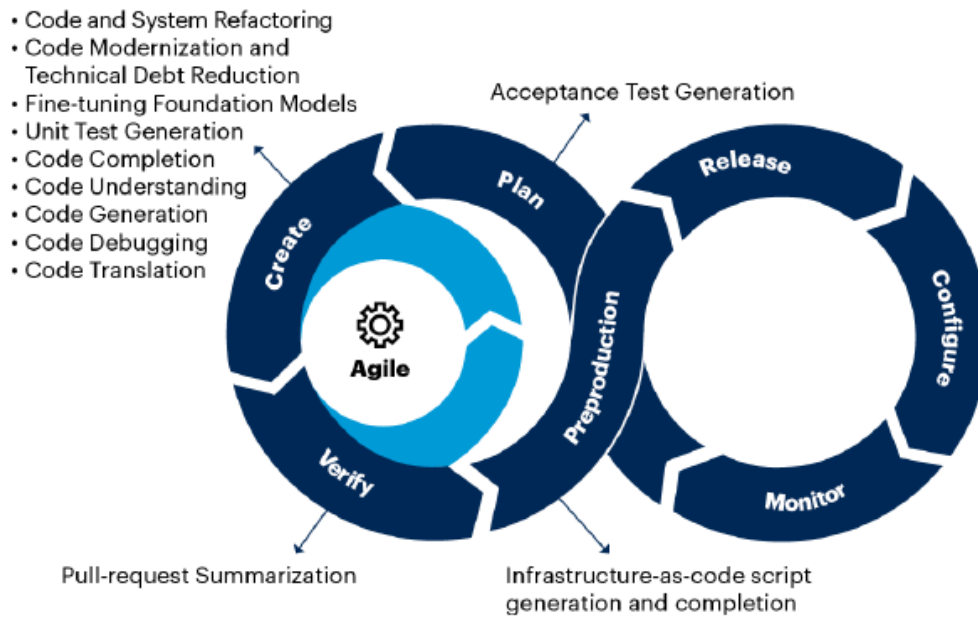
Batchun ja muiden (2024) mukaan johtajilla on edessään todella suuria haasteita, koska kilpailukykyisenä pysyäksään heidän tulee valita sopivat tekoälytyökalut tukemaan omaa liiketoimintaansa. Ohjelmointia tukevien tekoälytyökalujen määrä on vuonna 2023 noussut yli 40 kappaleeseen, kun vuonna 2022 niitä oli markkinoilla vasta muutamia. Oikeita työkaluja valittaessa on pystyttävä ottamaan huomioon muun muassa miten riskit ja laadunhallinta otetaan huomioon. Oman haasteensa valintaan tuo myös laidasta toiseen olevat raportit tekoälytyökalujen hyödyistä. Tämän takia johtajilla on ollut vaikeuksia tehdä investointipäätöksiä, sillä hyötyjä ei vielä kunnolla ymmärretä. Vuoden 2024 tutkimuksessa kävi ilmi, että 17 % yrityksistä on ottanut tekoälytyökaluja käyttöönsä, 23 % on vielä ottamassa, 20 % pilotoimassa ja 13 % suunnittelemassa käyttöönottoa. Kaiken kaikkiaan siis noin 73 % yrityksistä on ottamassa tekoälyä käyttöönsä. Melko vastaava luku käy ilmi myös Measuring Impact: The GenAI Code Report (2024) raportista, missä vastaava luku on 87 %. 2023 Developer Survey (2023) kyselytutkimuksen mukaan 70 % vastanneista käyttää tai tulee käyttämään tekoälyä. Saman tutkimuksen mukaan tekoälyn käyttö on yleistä etenkin alle viisi vuotta ohjelmointia tehneiden parissa ja harvinaisempaa yli 21 vuotta ohjelmoineiden keskuudessa. Myös tekoälyn hyödyntäminen oppimisessa on huomattavasti yleisempää kokemattomilla kehittäjillä.

Tällä hetkellä ohjelmistokehittäjät ovat yrityksiä edellä työkalujen käytössä. Joidenkin ennusteiden mukaan jopa yli 85 % kehittäjistä tulee käyttämään tekoälytyökaluja vuoteen 2028 mennessä. Tämä saattaa kuitenkin aiheuttaa useita eri ongelmia. Käytössä on laaja skaala erilaisia työkaluja sekä eri lisenssiversioita. Heikosti määritellyt ohjeistukset tuovat mukanaan tietoturvariskejä. Pahimmassa tapauksessa yrityksen koodit saattavat joutua tekoälyjen käyttämien aineistojen tietokantaan ja tätä kautta muiden yritysten käyttöön. Tästä syystä yrityksen on hyvä muodostaa jo alussa monipuolinen työryhmä, joka koostuu useista eri osaajista ja rooleista, jotta esimerkiksi tietoturva, arkkitehtuuri ja lakiasiat tulevat otettua huomioon, kun tekoäly otetaan käyttöön. (Batchun ym. 2024.)

Batchun ja muiden (2024) sekä Kuhailin ja muiden (2024) mukaan tekoälytyökalujen avulla kehittäjät voivat muun muassa tehokkaammin tuottaa koodia ja analysoida virheitä sekä tuottaa dokumentointia. Suurin osa ohjelmoinnissa käytettävien tekoälytyökalujen hyödyistä saadaan toteutusvaiheessa. Käyttöä ei kuitenkaan ole rajattu tähän. Esimerkiksi pilvipalveluihin pystytettävät palvelimet voidaan pystyttää tekoälyn tekemien skriptien avulla. Lisää käyttötapauksia on esitelty kuvioissa 14 ja 15. Raportissa korostetaan myös toiminnan mittaamista, sillä hyödyt saattavat olla osittain yrityskohtaisia ja jokaisen yrityksen on ymmärrettävä investointiensä hyödyt.

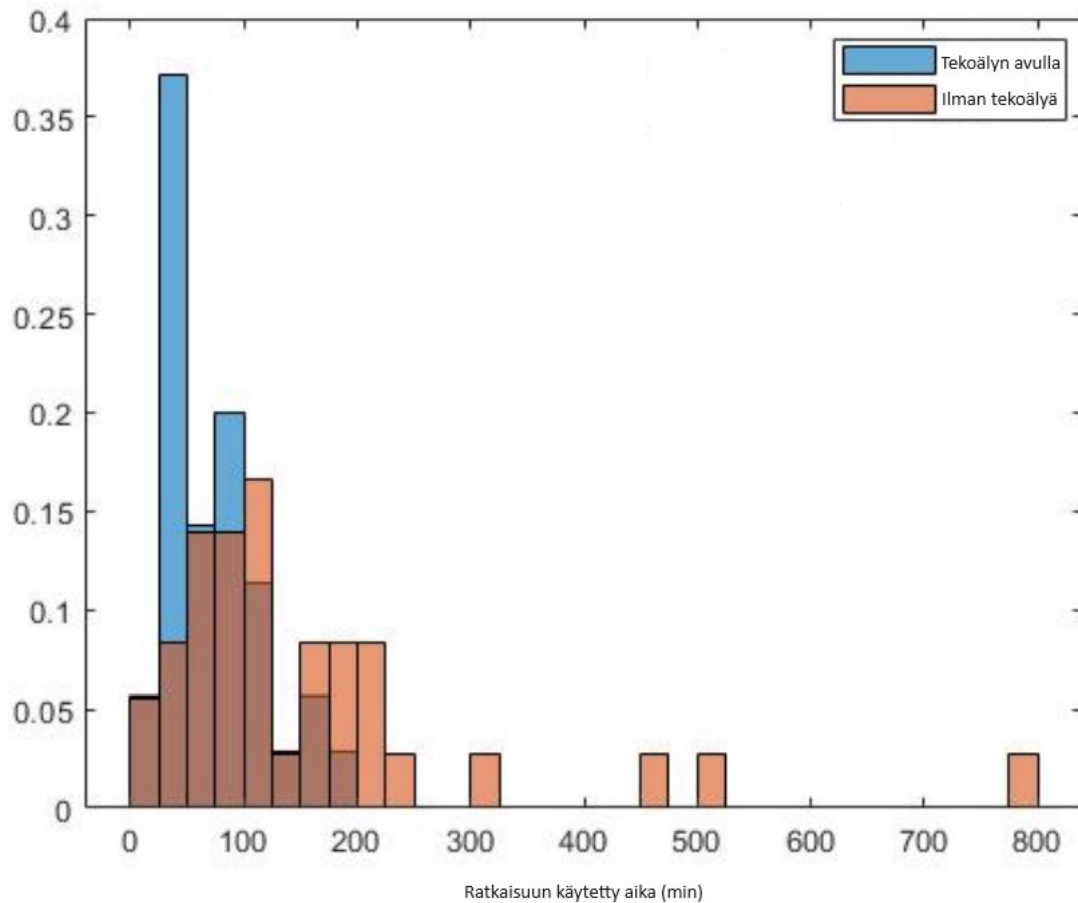


Kuvio 14. Tekoälytyökalujen tärkeimpiä käyttötapauksia (Batchun ym. 2024, muokattu)



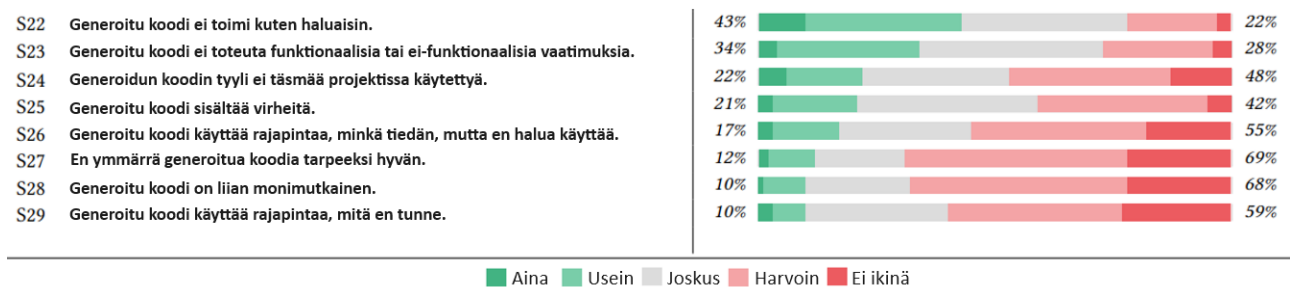
Kuvio 15. Tekoälytyökalujen käyttötapauksia muun muassa toteutusvaiheessa (Batcun ym. 2024, muokattu)

Batcun ja muiden (2024) mukaan jopa yli kaksikolmasosaa kyselyyn vastanneista uskoo generatiivisen tekoälyn lisäävän tuottavuutta yli 50 %. Myös Kuhailin ja muiden (2024) teettämä tutkimus tuki väitettä yli 50 % tuottavuuden kasvusta. Peng, Kalliamvakoul, Cihon ja Demirer (2023) teettivät empiirisen tutkimuksen, jossa kaksi eri kohderyhmää työstimät samoja tehtäviä. Toisella ryhmällä oli käytettävissään tekoälytyökalu ja toisella ei. Tekoälyä hyödyntäneet kehittäjät suoriutuivat tehtävistä 55,8 % nopeammin. Keskimäärin tekoälyn avulla tehtäviä suorittaneet kehittäjät käyttivät tehtäviin 71,17 minuuttia ja ryhmä ilman tekoälyä käytti tehtäviin 160,89 minuuttia. Kuviossa 16 on nähtävissä eri pilottiryhmien väliset erot käytettyyn aikaan nähden.



Kuvio 16. Eri pilottiryhmien erot tehtävien ratkaisuisissa (Peng ym. 2023, muokattu)

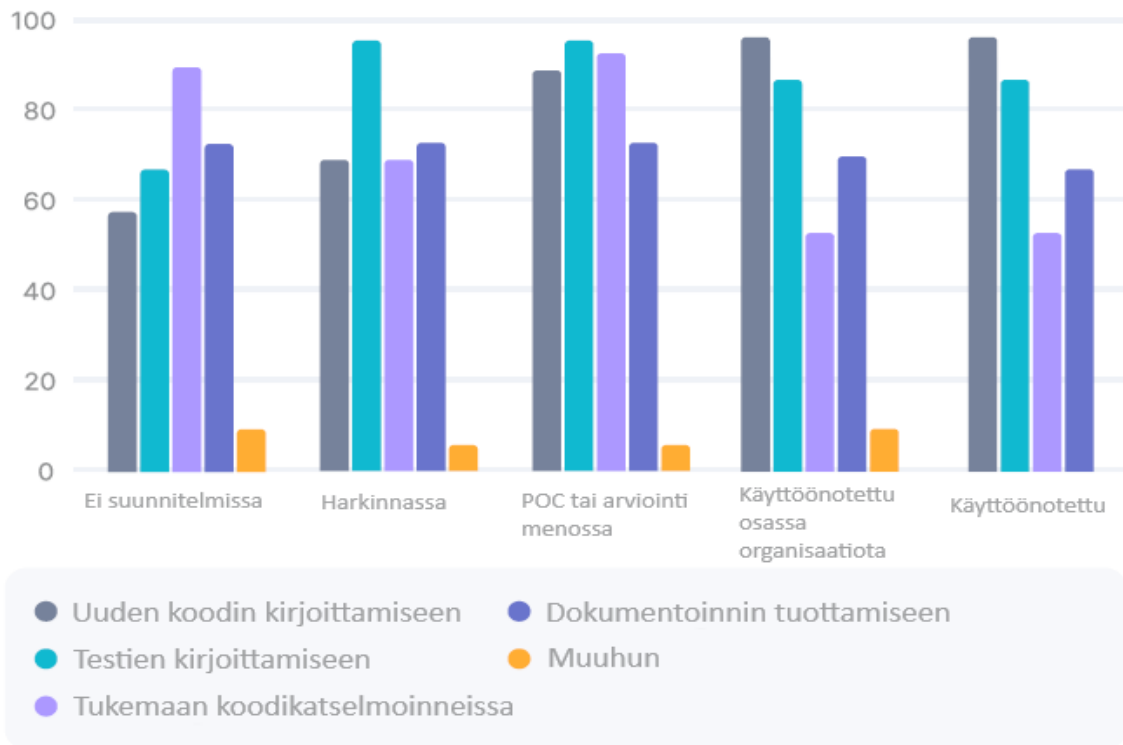
Liang, Yang ja Myersin (2023) tuottaman tutkimuksen mukaan tekoälytyökaluissa on kuitenkin edelleen omat haasteensa. Yhtenä isoimpana ongelmana nähdään tekoälyn kyvykyys tuottaa halettua koodia. Generoitu koodi ei myöskään aina täyttänyt vaatimuksia ja vastannut muun projektin ohjelmointityylejä (ks. kuvio 17).



Kuvio 17. Tekotyökalun haasteet (Liang ym. 2023, muokattu)

Jokaisen yrityksen on syytä toteuttaa konseptitodistus (POC) ja arvioida sen avulla tekoälyn hyötyjä yrityksen tarpeeseen nähden. On myös suositeltavaa käyttää vain yrityskäyttöön tarkoitettuja tekoälyjä turvallisuuden takia ja unohtaa yksityiskäyttöön tarkoitettut. Tässä vaiheessa on myös hyvä alkaa toteuttamaan suunnitelmaa kehittäjien kouluttamisesta. (Batcun ym. 2024.)

Tällä hetkellä yritykset näkevät koodin kirjoittamisen olevan tärkein ominaisuus, jota generatiiviselta tekoälyltä odotetaan. Tämän jälkeen selkeästi toisella sijalla tulee testien kirjoittaminen tekoälyn avulla. Dokumenttien generointi ja katselmoinnit koetaan tärkeänä, mutta monella yrityksellä ne ovat vasta tulossa, kuten kuviosta 18 käy ilmi.



Kuvio 18. Tekoälyn tilanne yrityksissä käyttökohteiden perusteella jaettuna (Measuring Impact: The GenAI Code Report 2024, muokattu)

7.2 Koulutuksen tehostaminen

Uusien ohjelmistokehittäjien kouluttaminen on aina kallista ja aikaa vievää niin oppilaitoksille kuin yrityksille. Myös uusien ohjelmointikielien skaala haastaa opiskelijaa, vaikka yhden ohjelmointikie-

len oppiminen tukeekin usein myös muiden kielien oppimista. Tekoälyn uskotaan helpottavan uusien kehittäjien oppimista ja mahdollistavan kehittäjien keskittymistä korkeamman tason ongelmiin (Bull & Kharuffa 2023). Koulutukselle nähdään kuitenkin tarvetta, sillä tekoälytyökalujen käyttämiseen kehittäjä tarvitsee hyvät pohjatiedot (Kuhail ym. 2024). Kuhail ja muut (2024) jatkavat, että opetuksessa olisi hyvä kuitenkin ottaa mukaan tekoälyn käyttäminen, jotta valmiudet työelämään paranevat.

Bullin ja Kharuffan (2023) mukaan automaatiot tuovat mukanaan myös huolen siitä, että tekoälyllä voi olla myös negatiivinen vaikutus oppimiseen etenkin, kun tarkastellaan kriittistä ajattelua ja ongelmanratkaisukykyä. Myös Moon, Yangin, Chain ja Kimin (2023) mukaan tekoälyn tuottamat vastaukset voivat aiheuttaa ongelmia pitkässä juoksussa, sillä käyttäjä ei välttämättä ymmärrä vastausta kunnolla. Tämä voi aiheuttaa ongelmia tiedon hyödyntämisessä eri tapauksissa, eikä se myöskään auta kehittäjää vahvistamaan ongelmanratkaisukykyään tai luovaa ajatteluaan. Bull ja Kharuffa (2023) kuitenkin mainitsevat, että vaikka ongelmassa voi olla perää, katsotaan tekoälyn hyötyjen olevan kuitenkin potentiaalista ongelmaa suuremmat, kun kehittäjien ei tarvitse opetella ulkoa niin paljon asioita, kuten funktioiden nimiä. Tekoälyn tapauksessa on tärkeää ymmärtää ero perinteiseen hakukoneeseen, koska tekoäly muodostaa syötteen perusteella vastauksen, joka perustuu osittain tekoälyn koulutuksen pohjalta saamaan ennustukseen. Käyttäjällä on kuitenkin vastuussa siitä, että ymmärtää, mitä koodi tekee ja että se on sopiva haluttuun käyttötapaukseen.

Vaikka tekoäly on teknologiana vielä ohjelmoinnissa suhteellisen tuore, ennustavat Bull ja Kharuffan (2023), että ohjelmistokehittäjien arki tulee muuttumaan. Tulevaisuudessa koodin kirjoittamisen sijaan kehittäjä työskentelee generatiivisen tekoälyn kanssa ja suunnittelee toteutuksia, kun taas tekoäly pitää huolen esimerkiksi tunnetuista algoritmeista. Tämän lisäksi jo olemassa oleva työelämätaito eli koodin lukeminen nousee vielä tärkeämpään rooliin tekoälyn kanssa. Myös Kuhailin ja muiden (2024) mukaan on tärkeää, että kehittäjät ymmärtävät tekoälyn tuottamaa koodia varmistaakseen laadukkuuden.

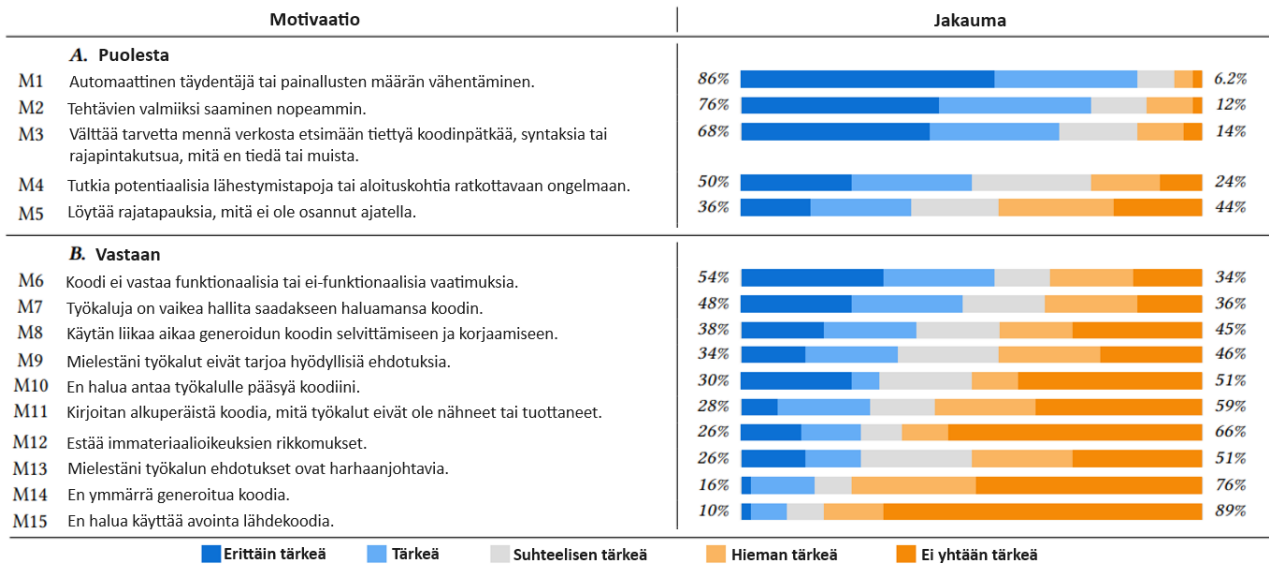
Bullin ja Kharuffan (2023) teettämässä tutkimuksessa mukana olleiden osallistujien mukaan ChatGPT ei tekoälytyökaluna vaadi syvällistä perehdytystä, vaan lyhyt esittely toiminnallisuudesta on riittävä. On myös tärkeää, että kehittäjällä on perustaidot ohjelmistokehityksestä. Osallistujien

mukaan tekoälystä on todennäköisesti hyötyä kaiken tasoille ohjelmistokehittäjille. Kokemattomamat kehittäjät voivat hyödyntää tekoälyä yksinkertaisten koodien luomiseen ja koodin selittämiseen, kun taas kokeneemmat kehittäjät voivat hyödyntää tekoälyä järjestelmän suunnittelemiseen.

Kuhail ja muut (2024) korostavat tekoälyn hyödyttävän uusia kehittäjiä siten, että ne voivat selittää mitä koodissa tapahtuu, miten koodin syntaksin kuuluisi olla, parhaita ohjelmoinnin käytänteitä sekä miten eri tietorakenteet ja algoritmit toimivat. Tekoälyn avulla voidaan siis saada nopeasti vastauksia ja tukea kehittäjille, jotka eivät vielä täysin ymmärrä helppoja tai keskivaikeita ohjelmointiin liittyviä asioita. Myös Moon ja muut (2023) mainitsevat tekoälyn toimivan hyvänä tuutorina kehittäjälle, sillä sen vastauksia voidaan personoida käyttäjän mukaan, eikä tekoäly tuomitse kysyjää. Personoidun tekoälyn avulla voidaan tehostaa kehittäjän oppimista, sillä tekoälyn vastaukset voidaan asettaa vastaamaan käyttäjän osaamistasoa, jolloin asian ymmärtäminen helpottuu. Vastauksien antaminen tasolla, jota käyttäjä ei kunnolla ymmärrä, heikentää oppimista ja voi aiheuttaa väärinymmärryksiä.

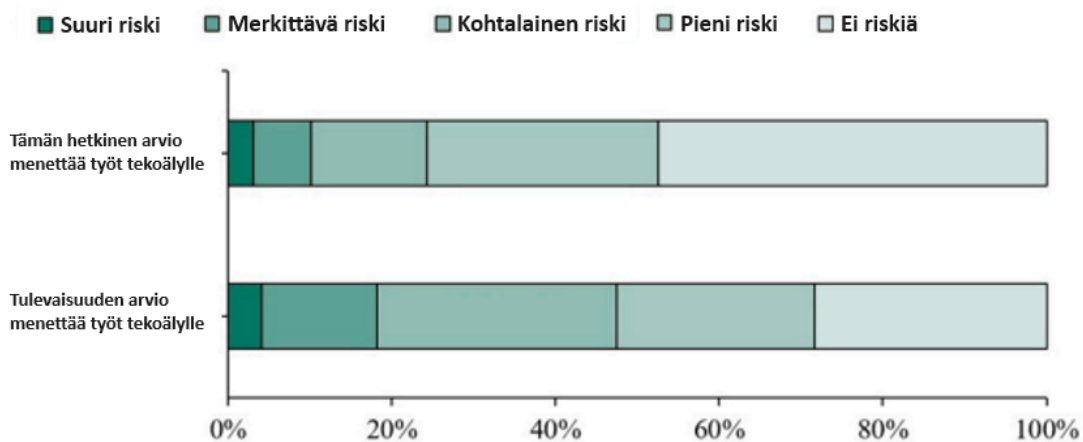
7.3 Tekoäly ohjelmoijan tukena

Kuhalin ja muiden (2024) tutkimuksen perusteella voidaan todeta, että kehittäjät käyttävät mielellään tekoälytyökaluja, sillä niiden avulla voidaan vähentää tehtävien suorittamiseen käytettyä aikaa ja näin ollen nostaa tehokkuutta. Tekoälytyökalujen käyttäminen vapautti aikaa enemmän luovien ja sosiaalisten tehtävien tekemiseen. Tekoälytyökalujen käytön katsottiin myös vähentävän pelkoa oman työnsä menettämisestä tekoälylle. Liang ja muut (2023) kuitenkin huomauttavat, että osa kehittäjistä ei halua käyttää tekoälyä tukena, sillä sen ei koeta tuottavan koodia, joka vastaisi oikeisiin vaatimuksiin. Lisäksi oikean syötteen antaminen tekoälylle koettiin haastavaksi. Liang ja muiden (2023) tekemän tutkimuksen mukaan kehittäjät hyväksyvät hieman alle 30 % tekoälyn tuottamasta koodista. Kuviossa 19 on esiteltyä tekoälypohjaisten työkalujen käyttöön liittyviä syitä, mutta myös perustelut sille, miksi tekoälyä ei haluta ainakaan vielä hyödyntää.



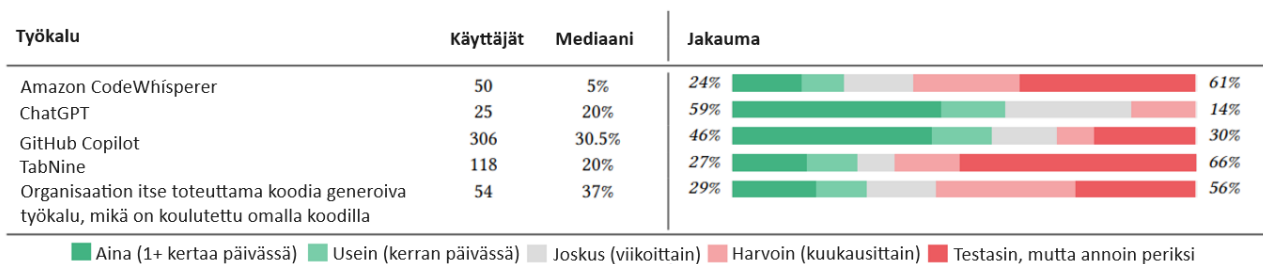
Kuvio 19. Perusteluita työkalun käytön puolesta ja vastaan (Liang ym. 2023, muokattu)

Kyselytutkimuksen perusteella kehittäjät eivät tällä hetkellä koe tekoälystä suurta uhkaa omista työtehtävistään syrjäyttämiseen (ks. kuvio 20), mutta epävarmuus kasvaa hieman, kun kysytään tulevaisuudesta. Vastauksien perusteella kehittäjiä tarvitaan ymmärtämään kokonaisuuksia sekä ratkaisemaan haastavampia ongelmia. Ne, jotka näkivät tekoälyn uhkana, kokivat työnkuvan muuttuvan siten, että tulevaisuudessa tarvitaan enemmän arkkitehtejä ja suunnittelijoita, ei ohjelmoijia. (Kuhail ym. 2024.) Myöskään Purwokon, Abdullahin, Wijayan, Gunawan ja Saputran (2023) eivät kokeneet tekoälyn pystyvän syrjäyttämään kehittäjiä tulevaisuudessa, vaan tehostavan kehitystä ja antavan kehittäjien keskittyä luovuutta vaativiin tehtäviin.



Kuvio 20. Miten vastaajat kokevat tekoälyn uhkana ammatilleen (Kuhail ym. 2024, muokattu)

Kuhailin ja muiden (2024) teettämässä kyselyssä lähes 60 % kehittäjistä käytti työssään ChatGPT:tä. Noin 26 % käytti GitHub Copilottia ja loput jotain muuta tekoälytuotetta. Yleisimpinä ohjelmointikielinä olivat Python ja JavaScript sekä Java. LinearB:n tuottaman raportin perusteella, GitHub Copilot on selkeä markkinajohtaja, sillä jopa 85 % käyttäjistä kertoi käyttävänsä kyseistä tuotetta (Measuring Impact: The GenAI Code Report 2024). 2023 Developer Surveyn (2023) mukaan noin 55 % käyttäjistä on käyttänyt tai käyttää GitHub Copilottia. Liang ja muiden (2023) mukaan 46 % kehittäjistä kertoi käyttävänsä pääasiassa GitHub Copilottia ja vastaavasti 59 % ChatGPT:tä. Kuviossa 21 on esiteltyä yhden tutkimuksen tuloksia eri tekoälytyökalujen käyttöasteesta.



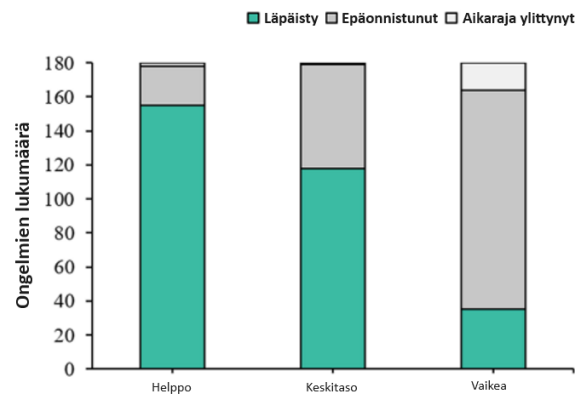
Kuvio 21. Tekoälytyökalujen käyttöaste (Liang ym 2023, muokattu)

Uuden koodin tuottaminen

Kuhailin ja muiden (2024) mukaan tekoäly pystyy antamaan suosituksia ohjelmistokehittäjälle ja ratkomaan ongelmia. Tekoälyn tehokkuus on kuitenkin selkeästi suhteessa ongelman haastavuuteen. Testeissä on havaittu, että tekoäly kykenee jo generoimaan koodia englanninkielisen syöteen avulla, sillä tasolla, että se päihitti yli 80 % aloittavista kehittäjistä pilotoinnissa. Toisessa testissä tekoäly kykeni olemaan melkein 60 % kerroista ihmistä nopeampi. Ratkaisujen muistinkäyttö oli myös lähes 70 % tapauksista ihmistä vähäisempi. Nopeuden lisäksi tekoäly kykeni siis luomaan tehokkaampia ja vähemmällä muistilla toimivia koodinpätkiä kuin opiskelijat. Purwokon ja muiden (2023) tutkimuksen mukaan etenkin web-pohjaisten sovellusten kehittäminen tehostuu tekoälyn avulla.

Tekoälyn haasteena ovat kuitenkin vaikeammat tehtävät ja ongelmat, joten näihin tarvitaan edelleen ratkaisemaan ihminen. Kuviosta 22 voidaan havaita, miten tekoälyn kyvykyys ratkaista ongelmia heikkenee ongelman vaikeutuessa. Tekoälyn tuottamassa koodissa on myös ihmisen tuottamaan koodiin verrattuna enemmän sekä syntaksi-, että semanttisia virheitä. Moni kehittäjä

hyödyntääkin tekoälyä mieluummin esimerkiksi yksikkötestien ja koodipohjien luomiseen ja ratkaisee itse vaikeammat tehtävät. (Kuhail ym. 2024.)



Kuvio 22. Tekoälyn ongelmanratkaisukyky ongelman vaikeutuessa (Kuhail ym. 2024, muokattu)

Tekoälypohjaisella työkalulla on myös uuden koodin tuottamisessa rajoituksia. Esimerkiksi Chat-GPT:llä on rajoituksia useamman eri asian yhtäaikaisessa käsittelyssä. Tämän lisäksi käytössä on rajoitus tekstin pituudelle, mikä saattaa katkaista tekoälyn generoiman tekstin kesken kaiken. Tämä saattaa johtaa toimimattomaan tai väärin toimivaan ohjelmaan. Ratkaisuna on jakaa tehtävänantoa vielä pienempiin osiin ja antaa tekoälyn generoida tekstiä pienemmissä osissa. (Kuhail ym. 2024.)

Ohjelmistojen korjaaminen

Ohjelmistojen ongelmien korjauksissa yleisesti yksi eniten aikaa vievistä kohdista on vian paikantaminen. Bayrin ja Demirelin (2023) sekä DePalman, Miminoshvilin, Henselderin, Mossin ja AIOmarin (2024) mukaan tekoäly pystyy auttamaan ongelman juurisyyn paikantamisessa. Se kykenee analysoimaan muun muassa virhelokeja ja käyttäjän antamaa palautetta potentiaalisesti virheellisistä kohdista. Tämän avulla tekoäly pystyy myös ehdottamaan mahdollisia korjauksia vialle ja nopeuttamaan kriittisten ongelmien korjaamista.

Eri tietolähteiden määrä voi olla hyvinkin laaja ja tekoälyn kyky käsitellä nopeasti suuria määriä dataa mahdollistaa kokonaisvaltaisia analyysejä ongelmasta. Tekoäly kykenee myös jatkuvasti oppi-

maan uutta ja kehittymään, mikä edesauttaa sen suoriutumista ongelmien ratkaisussa. Oletuksena on, että tekoälyn hyödyntäminen virheiden korjauksessa tulee olemaan hyvin olennainen osa ohjelmistokehittäjien arkea.

Refaktorointi

Refaktoroinnin tarve on aina olemassa ohjelmistokehityksessä. Tähän ajaa muun muassa jatkuvasti kehittyvä ala, joka tuo mukanaan uusia tapoja tuottaa tehokasta koodia. Refaktorointi tähtää koodin laadun ja tehokkuuden parantamiseen tekemällä koodiin muutoksia tai kirjoittamalla jopa koodin osia uusiksi muuttamatta kuitenkaan ohjelmiston toiminnallisuutta.

DePalman ja muiden (2024) ja Purwokon ja muiden (2023) mukaan ChatGPT:n avulla voidaan koodin refaktorointi viedä seuraavalle tasolle. Vaikka ChatGPT häviää useammalle olemassa olevalle työkalulle joissain asioissa, kuten koodin semanttisen rakenteen ymmärtämisessä tai koodin yhteenvedossa, on sillä kuitenkin kokonaisvaltaisesti paljon potentiaalia mullistaa ohjelmistokehitystä. ChatGPT:llä on kykyjä, joita kaikilta muilta vastaavilta tekoälyiltä ei vielä löydy, ja sen kyky muistaa pidempiä keskusteluja ja luontainen tapa kommunikoida ovat poikkeuksellisia.

ChatGPT:n kyvykkyydestä refaktoroida koodia kertoo data-analyysi, jossa sille syötettiin 320 eri syötettä ja pyydettiin refaktorimaan koodia. ChatGPT epäonnistui vain yhdessä tapauksessa eli onnistumisprosentti oli huikea 99,7 %. Tämän lisäksi tekoäly kykenee luomaan dokumentaatiota refaktorointiin liittyen. Testissä vain kymmenessä eri tapauksessa 320:stä havaittiin virheitä. (DePalman ym. 2024.)

DePalman ja muiden (2024) tuloksista käy ilmi, että tekoäly ei kuitenkaan refaktoroinnilla saavuttanut aina merkittäviä parannuksia koodin tehokkuuteen. Useimmissa tapauksissa tekoäly tarjosi kyllä parannetun version koodista, mutta muutosten laajuus ja laadukkuus vaihteli laidasta toiseen. Joissain tapauksissa tekoäly saattoi vain muuttaa muuttujien nimiä parempaan suuntaan. Jossain tapauksissa tekoäly kykeni tehostamaan koodia esimerkiksi poistamalla tarpeettomia funktioita. Parhaiten tekoäly soveltui tehtäviin, joissa sen piti tehdä muutoksia Java-ohjelmointikielelle sovittujen käytänteiden mukaisesti, kuten uudelleennimetä luokkia tai muuttujia. Vastaavaan lopputulokseen päätyivät myös Purwokon ja muut (2023) tutkimuksessaan. Tekoälyä hyödyntävän

kehittäjän vastuulle jää aina varmistaa, että koodi pysyy luettavana, sillä usein myös muut ohjelmistokehittäjät lukevat samaa koodia ennemmin tai myöhemmin.

Tekoälyn on haastava ymmärtää sille syötettyyn koodinpätkään liittyvää kokonaisuutta ja sen sisältöä, minkä takia se tuottaa toisinaan virheitä tai ehdottaa parannuksia, jotka eivät ole tarpeellisia. Kehittäjällä on lähtökohtaisesti aina parempi ymmärrys kokonaisuudesta, johon myös tekoälylle syötetty osuus kuuluu. Vastauksien muodostamiseen liittyy myös olennaisesti se, minkälaisella datalla ja kuinka laajasti tekoäly on koulutettu. Tämän lisäksi joitain tietoja on voitu rajoittaa tekoälyltä tai ongelma voi vain yksinkertaisesti olla liian haastava tekoälyn ratkaistavaksi tässä vaiheessa kehitystä. (DePalman ym. 2024; Kuhal ym. 2024.)

DePalman ja muiden (2024) mukaan tekoälyä pystyi hyödyntämään parhaiten silloin, kun syötetty kuvaus ei ollut liian pitkä. Lyhyiden syötteiden todettiin olevan tehokkaimpia etenkin koodin ongelmien korjaamiseen. Myös syötteen kanssa tuli olla tarkkana, sillä pienetkin muutokset saattoivat muuttaa tekoälyn tapaa käsitellä syötettä. Tämän lisäksi tutkijat havaitsivat, että vaikka tekoälylle syöttää saman kysymyksen 100 kertaa, saa takaisin aina eri vastauksen. Vastaus on samantyylinen, mutta ei samanlainen.

Tekninen velka

Tekninen velka voidaan määritellä siten, paljonko ohjelmiston pitää maksaa pidemmällä tähtäimellä, koska kehittäjä tahallisesti tai tahattomasti teki valintoja, joilla saavutettiin lyhyen aikavälin tavoitteet. Tekninen velka on aina negatiivinen asia, mutta se on myös aina olemassa ja koko ajan entistä haastavampana. Teknisen velan korjaaminen alkaa tunnistamalla tekninen velka, priorisoimalla, seuraamalla ja korjaamalla sitä. (Binta, Kaushal & Pandi 2023.) Tästä syystä on myös hyvä tarkastella tekoälyn hyödyntämistä teknisen velan vähentämisessä.

Bintan ja muiden (2023) mukaan tekoälyn luonnollisen kielen avulla voidaan analysoida eri lähteitä, kuten kommentteja ja huomata mahdollisia ongelmia, jotka aiheuttavat teknistä velkaa. Myös dokumentaatiosta voidaan havaita puutteita ja velkaa tekoälyn analyysien avulla. Yhtenä tehokkaana tapana tunnistaa ongelmia on yhdistää tekoälyn ja staattisten analysointivälineiden voima.

Staattinen analysaattori voi koodin pohjalta tunnistaa teknistä velkaa ja tekoäly käsitellä dokumentaation avulla, miksi valinta on tehty ja päätellä teknisen velan vakavuuden tällä perusteella.

Toisena vaihtoehtona on käyttää hyväkseen tekoälyn syväoppimista, jonka avulla voidaan tarkastella suuria määriä koodia ja ymmärtää eri kuvioita, mitkä saattavat johtaa tekniseen velkaan. Tutkimuksessa, jossa hyödynnettiin tekoälyn syväoppimista, tunnistettiin teknistä velkaa koodista. Tämän lisäksi tekoäly osasi korjata havainnot. Jopa yli 55 % teknisestä velasta saatiin poistettua tekoälyä hyödyntäen. Toisessa tutkimuksessa tarkasteltiin tekoälyn avulla myös lähdekoodia. Teknisen velan tunnistamisen lisäksi tekoäly arvioi, paljonko tekninen velka pahenee vuosien saatossa ja miten heikentävästi se vaikuttaa ohjelmiston toimintaan. (Binta ym. 2023.)

Dokumentointi

Dokumentointi on olennainen osa ohjelmiston ylläpitoa ja jatkokehitystä. Ilman kunnollista dokumentaatiota vain ohjelman kehittänyt kehittäjä voi tehokkaasti tehdä koodiin muutoksia. Myös tässä piilee vaara, että tekijä joutuu tekemään muutoksia monen vuoden jälkeen, jolloin ohjelman tekninen puoli ei ole enää niin hyvin muistissa.

Aikaisemmin dokumentaatiota on tehty joko manuaalisesti tai osittain automaation avulla. Tämä luo kuitenkin oman haasteensa aina, kun muutostarpeita dokumentaatioon esiintyy. Tekoälyn avulla, joka hyödyntää koneoppimista ja luonnollisen kielen käsittelyä, voidaan dokumentoinnin automatisointia ja tämän myötä luotettavuutta parantaa. Tekoäly voi automaattisesti havaita päivitystarpeet ja päivittää dokumentaation. (Dhyani, Nautiyal, Negi, Dhyani & Chaudhary 2024.)

7.4 Testauksen kehittäminen

Ohjelmistojen testaaminen on välttämätöntä, jotta mahdolliset virheet ja haavoittuvuudet voidaan paikantaa ja korjata. Tyypillisesti testausta on toteutettu manuaalisesti tai staattisten testaus työkalujen avulla aina yksikkötestauksesta laajempiin kokonaisuuksiin. Tekoälyn avulla useita näitä prosesseja voidaan tehostaa ja automatisoida, mikä auttaa tehostamaan testausta ja parantaa ohjelmistojen laatua. Tekoälyn avulla voidaan parantaa testauksen tuottavuutta, vähentää kustannuksia sekä säästää aikaa. (Bayri & Demirel 2023.) Myös Mehmoodin, Janjuan ja Ahmedin

(2023) mukaan tekoälyn hyödyntämisen luonnollisen kielen avulla testejä voidaan luoda huomattavasti aikaisempaa nopeammin ja kattavammin.

Bayrin ja Demirelin (2023) mukaan tekoälyn avulla voidaan tehostaa yksikkötestauksen tehokkuutta ja tarkkuutta, sillä tekoäly kykenee jo soveltamaan useita eri tekniikoita tällä alueella. Yhtenä suurena etuna on vaatimuksen analysointi ja oikeanlaisen testidatan luominen testausta varten. Tutkimuksessa oli tekoälyn avulla luotu 207 eri luokkaa Java-ohjelmointikielillä. Melkein 70 % näihin luokkiin tekoälyn avulla generoiduista testitapauksista pystyttiin suorittamaan ilman virheitä. Myös Mehmoodin ja muiden (2023) testeissä onnistumisprosentti oli 75 %, mikä on hyvin linjassa Bayerin ja Demilerin (2024) tuloksiin. Joissain tapauksissa tekoäly havaitsi testien avulla virheitä, joita käsin tehdyt testit eivät löytäneet. Mehood ja muut (2023) kuitenkin mainitsivat, että myös tekoälyllä on testien luomisen suhteen rajoituksia. Tekoäly ei esimerkiksi pysty hallitsemaan haastavia rakenteita ja datojen välisiä riippuvuuksia.

Myös testitapausten kattavuus on merkittävä osa testaamista. Bayr ja Damiel (2023) vertasivat aikaisempaa työkalua, joka loi koodin pohjalta testitapauksia ChatGPT:seen. Työkalu, joka ei hyödynnä tekoälyä, pääsi parhaimmillaan 77,4 % kattavuuteen. ChatGPT pääsi parhaillaan vain 55,4 % kattavuuteen, eli tekoälyllä on vielä matkaa tässä tapauksessa, mutta ottaen huomioon kyseisen työkalun uutuuden, on kyseessä varsin kiitettävä suoritus.

Automaattisten testien luettavuuden on myös oltava riittävällä tasolla, jotta ohjelmoijat ymmärtävät, mitä koodissa tapahtuu ja mitä testi testaa. Tämän lisäksi heillä on oltava tarvittaessa kyky muokata ja päivittää testejä. Bayrin ja Damielin (2023) tutkimuksen mukaan ChatGPT:n automaattisesti luomien testien luettavuus oli hyväksyttävällä tasolla ja joskus jopa paremmalla tasolla kuin käsin kirjoitetuilla testeillä.

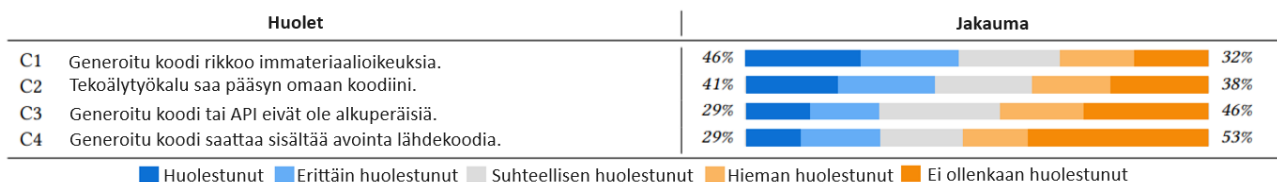
Bayerin ja Demirelin (2023) mukaan tekoälyllä voidaan luoda kokonaisia testauspaketteja vaatimusten perusteella. Tämä vähentää merkittävästi kustannuksia ja nopeuttaa ohjelmistokehitystä, kun manuaalinen vaihe on ulkoistettu tekoälylle. Generatiivisen tekoälyn mukaan tuleminen uusien suurien kielimallien kanssa on mahdollistanut käyttäjäläheisen ja ymmärrettävän tekoälyn hyödyntämisen. Esimerkiksi tekoälyn kanssa voi vielä tarvittaessa tarkentaa vaatimuksia tai haluttuja tuloksia käyttämällä normaalia kieltä.

Tekoälyn hyödyntäminen testauksessa vähentää pullonkaulojen muodostumista, mikä vähentää läpimenoaikaa. Kun tekoäly hoitaa osan laadunvarmistuksesta, jää kehittäjille aikaa keskittyä vaativampiin tehtäviin, kuten arkkitehtuurin ja käyttäjäkokemuksen suunnittelemiseen. Etenkin suuria kielimalleja hyödyntävillä tekoälyillä on valtava mahdollisuus uudistaa ohjelmistotestaus ja etenkin tapaa, jolla testausta on toteutettu. (Bayer & Demirel 2023.)

7.5 Turvallisuuden ja luotettavuuden huomioiminen

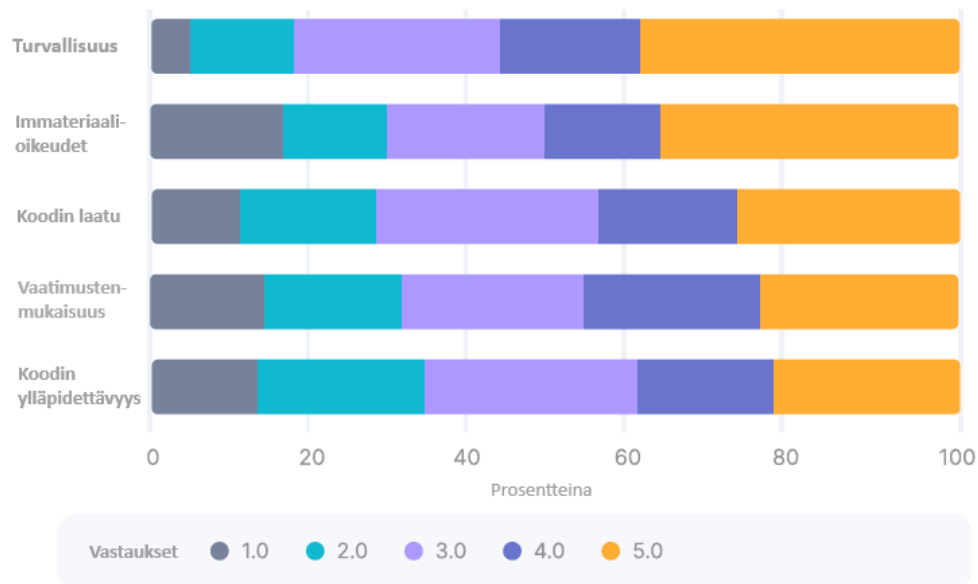
Tekoälyjen käyttämät suuret kielimallit vaativat suuria määriä dataa toimiakseen tehokkaasti ja luotettavasti. Tämän myötä on olemassa riski, että tekoäly paljastaa tahtomattaan arkaluonteista tai yksityisyyteen liittyvää dataa. Tämän takia on suotavaa, että datan anonymisointiin sekä tiedonpääsyn hallintaan on panostettu. (Bayri & Demirel 2023.)

Shin, Sakibin, Shahriarin, Loin, Chin ja Qian (2023) tutkimuksen perusteella tekoälytyökalut voivat luoda ehdotuksia koodimuutoksista ja estää turvattomien koodien tuottamisen. Valitettavasti tekoälyt saattavat myös olla itse syyllisiä haavoittuvan koodin tuottamiseen. Niiden tuottamat tulokset eivät aina ole käyttäjälle turvallisia. Vastaavia havaintoja nousi esille myös Liang ja muiden (2023) tutkimuksesta, jossa kehittäjät olivat huolissaan, että tekoäly saattoi tuottaa virheellistä koodia. Onneksi tekoälytyökalujen kehittäjät ovat panostaneet tähän ja esimerkiksi yksi käytetyimmistä tekoälytyökaluista GitHub Copilot sisältää toiminnon, jonka pitäisi estää haavoittuvien syötteiden tuottaminen. Suurimmaksi huoleksi kehittäjät kuitenkin nostivat tekoälyn tuottamaan koodiin liittyvät tekijänoikeudelliset asiat (ks. kuvio 23). Tällä hetkellä myös kehittäjän vastuulla on tarkkaan määritellä, minkälaista koodia hän tekoälyltä odottaa. Esimerkiksi yksinkertainen kirjautumiseen vaadittava koodinpätkä saattaa sisältää tietoturva-aukkoja, jollei kehittäjä varta vasten pyydä tekoälyä huolehtimaan, ettei niitä synny. (Shin ym. 2023.)



Kuvio 23. Kehittäjien huolia tekoälyn tuottamasta koodista (Liang ym. 2023, muokattu)

Kyselyiden perusteella yrityksiä huolestuttaa tekoälyn käytössä eniten turvallisuuteen ja immateriaalioikeuksiin liittyvät asiat. Tämän jälkeen tulevat koodin laatu, jolla saattaa olla vaikutuksia liiketoimintaan ja käyttäjien turvallisuuteen. Neljäntenä yritykset ovat huolissaan siitä, että ratkaisut ovat vaatimustenmukaisia. Kuviossa 24 on esiteltyä jakauma vastaajien kesken. Numero 1 tarkoittaa, että käyttäjää huolestuttaa vähän ja numero 5 tarkoittaa merkittävää huolta asiasta. (Measuring Impact: The GenAI Code Report 2024.)



Kuvio 24. Huolestuttavimmat alueet tekoälyn käytössä (Measuring Impact: The GenAI Code Report 2024, muokattu)

8 Pohdinta

8.1 Tutkimuksen luotettavuus

Hyvään tieteelliseen tutkimuksen toteuttamiseen kuuluu muun muassa yleinen huolellisuus ja tarkkuus tutkimustyössä, sekä tiedon esittämisessä että tulosten arvioinnissa (Hyvä tieteellinen käytäntö 2023). Tutkimuksen toistettavuudella voidaan myös varmistaa sen luotettavuus. Kirjallisuuskatsauksessa tärkeimmiksi asioiksi toistettavuuden kannalta nousevat katsauksessa käytettyjen lähteiden merkitseminen sekä aineiston keruun kuvaaminen, jotta vastaavat hakutulokset ovat saatavissa mahdollisimman tarkkaan. Tässä tutkimuksessa aineistonkeruun suunnitelma on selkeästi nähtävillä. Kuvauksesta käy ilmi, mistä tietokannoista ja millä hakusanoilla aineistoa kerättiin.

Lisäksi käytetyt tietokannat ja hakusanat ovat esitelty perusteltuina. Tämän lisäksi aineiston käsittelyprosessi on kuvattuna PRISMA-mallin avulla. Kaikki katsauksessa käytetyt lähteet ovat selkeyden vuoksi sijoitettu erikseen liitteeseen 1, jotta lukijan on helpompi arvioida käytettyjä lähteitä näin halutessaan.

Kirjallisuuskatsauksen yleisimpiä ongelmia on katsauksen lähteiden luotettavuus (Vilkka 2023). Valitut aineistot analysoitiin eri kriteerien perusteella, jotka ovat nähtävissä liitteessä 2. Kaikki mukaan valitut aineistot täyttivät tieteellisen tutkimuksen kriteerit muutamia poikkeuksia lukuun ottamatta. Integratiivinen kirjallisuuskatsaus sallii systemaattista kirjallisuuskatsausta laajemman aineiston keruun (Vilkka 2023). Tätä mahdollisuutta hyödyntäen mukaan valittiin aineistoja, jotka ovat yritysten tuottamia. Osasta materiaaleista käy ilmi kaupalliset taustat, mutta tutkittu aineisto tuki hyvin muita tutkimuksia sekä toi mukaan uusia näkökulmia. Luotettavuuden näkökulmasta tutkimus on siis hyvin läpinäkyvä ja toistettava.

8.2 Tutkimuksen eettisyys

Eettisesti tarkasteltuna toimeksiantajan toimeksianto ei ohjaa työtä suuntaan tai toiseen vaan tutkimus toimii päätöksenteon tukena tulevaisuuden investointeja silmällä pitäen. Työssä on vaalittu hyvän tutkimusetiikan mukaisia tapoja muun muassa kunnioittamalla katsaukseen valittujen tutkimusten tuotoksia selkeillä ja asianmukaisilla viittauksilla. Tuloksien käsittelyssä on pyritty tuomaan esille tekoälyn käyttämisen positiivisia ja negatiivisia puolia. Tässä tutkimuksessa esitellyt tulokset ja johtopäätökset perustuvat aikaisempiin tutkimuksiin, mutta siitä huolimatta on hyvä ymmärtää, että tutkimuksessa esitetyt asiat ovat kirjoittajan havaintoja aikaisempien tutkimuksien pohjalta ja saattavat sisältää puutteita. On siis tärkeätä, että tutkimuksen viittaukset ja lähteet ovat kunnossa, jotta kriittinen tarkasteleminen on tarvittaessa mahdollista.

Tekoälyn hyödyntämiseen liittyy myös useita eettisiä ongelmia. Tietojärjestelmien virheiden on jo nähty aiheuttavan vääriä tuomioita ihmisille, kuten esimerkiksi Englannissa, jossa postin käyttämän järjestelmävirheen vuoksi useita ihmisiä tuomittiin väärin perustein, jopa vankeusrangaistukseen (Peachey 2021). Myös tekoälyn hyödyntäminen päätöksenteossa esimerkiksi vakuutusten tai lainan saannin suhteen voi olla ongelmallista. Esimerkiksi ihmisten ihonväri tai asunnon sijainti voi johtaa kielteiseen päätökseen (Browne & Sigalos 2023). Tekoälyä ei välttämättä ole opetettu hyö-

dyntämään näitä tietoja päätöksenteossa, mutta sille annettu opetusaineisto saattaa sisältää näihin liittyviä asioita. Tämän takia tekoälylle koulutettu aineisto on sen vahvuus, mutta samalla myös heikkous, sillä puutteellinen tai virheellinen aineisto voi johtaa hyvinkin epäeettisiin johtopäätöksiin.

Tutkimusten mukaan tietotyöntekijöiden työtehtävät tulevat muuttumaan tai muokkaantumaan tekoälyn hyödyntämisen takia tavalla tai toisella. Tekoälyn tulemistä on verrattu seuraavaksi teolliseksi vallankumoukseksi (Velarde 2020). Tällä kertaa tehtaiden sijasta kyseessä on tietotyöläiset, joiden tehtäviä voidaan tehostaa ja muokata tekoälyn avulla. Esimerkiksi Pohjois-Pohjanmaan hyvinvointialue korvasi 63 tekstinkäsittelijää puheentunnistusohjelmistolla (Tekstinkäsittelijöitä koskevat yhteistoimintaneuvottelut ovat alkaneet 2023). Myös finanssialan yhtiö Klarna korvasi jopa 700 työntekijän tehtävät tekoälyn avulla (Kelly 2024). Etenkin tekoälyn käyttäminen toiminnan tehostamiseksi samalla kun ihmiset jäävät työttömäksi aiheuttaa eettisiä kysymyksiä.

Tekoälyn hyödyntämään koulutusaineistoon liittyy myös useita eettisiä ja immateriaalioikeuteen kuuluvia ongelmia. Useat tekoälyt hyödyntävät internetissä vapaasti olevaa materiaalia oppimiseen, ja joukossa saattaa olla myös aineistoa, mikä rikkoo käyttöoikeuksia. Esimerkiksi uutislehti New York Times haastoi OpenAI:n ja Microsoftin oikeuteen, sillä heidän mukaansa ChatGPT on hyödyntänyt miljoonia lehden maksullisia artikkeleita ilmaiseksi ilman lupaa (Grynbaum & Mac 2023).

8.3 Johtopäätökset

Tutkittavaa materiaalia tekoälyn hyödyntämisessä ohjelmoinnissa löytyy nykypäivänä todella paljon. Generatiivisen tekoälyn hyödyntäminen nimenomaan ohjelmointitehtävissä työkalujen muodossa rajaa aluetta paljon, mutta tietoa on silti huimasti saatavilla. Tämä haastaa kokonaisvaltaisen analyysin muodostamista, sillä lähdeaineiston kriittinen arviointi on työlästä ja haastavaa. Aineistojen vertaaminen keskenään tuo hyvin esiin mahdolliset epäkohdat tutkimusten välillä. Tutkimuksen osalta oli erittäin tärkeää valita mahdollisimman tuoreita tutkimuksia, sillä tekoälyn kehittyminen on tällä hetkellä todella nopeaa ja osa tutkimuksista vanhenee hyvin nopeasti. Alustavassa aineistonkeruu suunnitelmassa haku oli tarkoitus toteuttaa vuosille 2022–2024, mutta aineiston määrä jo vuosille 2023 ja 2024 yllätti, minkä takia tarvetta vuoden 2022 tutkimuksien tarkastelulle ei enää nähty.

Katsauksen tuloksista käy selkeästi ilmi, että tekoälypohjaisten työkalujen hallitseminen tulee olemaan työelämätaito myös ohjelmistokehityksen alueella. Työkalujen olemassaoloaika on ollut erittäin lyhyt, joten siihen nähden niiden tämänhetkinen kypsyyden on jo yllättävän hyvällä tasolla. Tuotavuuden kasvu yli 50 %:lla kuulostaa todella isolta, mutta kaikki katsaukseen valitut tutkimukset osoittivat vastaavia tuloksia. Tutkimuksia oli lisäksi toteutettu hieman eri tavalla, mikä lisää uskottavuutta kasvun lukemaan. Lähtökohtaisesti kuitenkin jokaisen yrityksen tulisi itse luoda mittarit ja pilotoida tekoälytyökaluja, ennen kuin tekee isompia investointipäätöksiä niiden suhteen.

Tekoälyn hyödyistä ja mielenkiinnosta aiheeseen kertoo ennätysmäiset käyttäjäluvut esimerkiksi ChatGPT-työkalun suhteen. Tekoälypohjaisilla työkaluilla uskotaan saavan hyötyjä muun muassa koodin laatuun, kehittäjien tuottavuuteen, työntekijäkokemukseen, toimintavarmuuteen sekä perehtymiseen. Potentiaali on jo hyvin monipuolinen pelkissä ohjelmointiin liittyvissä tehtävissä. Tämä selittää todennäköisesti sen, miksi jopa yli 70 % yrityksistä suunnittelee tekoälyn käyttöönottamista toiminnassaan. Generatiivisen tekoälyn monipuolisuus ja käyttökohteiden ymmärtäminen on todennäköisesti vasta alkanut ja potentiaalisissa käyttökohteissa on raapaistu vasta pintaa. Käynnissä on vaihe, missä yritämme hyödyntää tekoälyä integroimalla sitä nykyisiin sovelluksiin, mutta uusien tekoälyä hyödyntävien sovelluksien tutkiminen on vielä suhteellisen alkutekijöissään.

Generatiivisten tekoälyjen käyttö ei vielä kuitenkaan onnistu ongelmitta. Useista tutkimuksista kävi ilmi, että tekoälyn tuottama koodi ei aina toimi, kuten kehittäjä olisi halunnut. Joissain tapauksissa koodi saattaa toimia, mutta se ei sisällä tarpeellisia toimintoja, toteutettu tyyli ei täsmää muun projektin kanssa tai toteutuksessa on mukana rajapintoja tai kolmannen osapuolen koodia, mitä ei haluta käyttää omassa toteutuksessaan. Toisinaan myös tekoälyn luoma koodi on epäselvää, liian monimutkaista tai se sisältää kehittäjälle tuntemattomia asioita.

Useista tutkimuksista käy myös ilmi hyödyt tekoälyn hyödyntämisestä ohjelmoinnin opiskelun tehostamisessa. Tekoäly kykenee tarvittaessa toimimaan esimerkiksi tuutorina ja selittämään koodin toiminnallisuutta kehittäjälle. Useissa tutkimuksissa tekoälyn hyödyt korostuivat juuri alle 5 vuotta ohjelmointia tehneiden parissa, joille koodin syntaksi tai algoritmit ovat vielä tuntemattomia. Negatiivisena asiana tässä koettiin se, että usein tekoäly tarjoaa liian valmiita vastauksia, jolloin konkreettinen oppiminen jää heikoksi. Loogisen päättelykyvyn sekä ongelmanratkaisutaitojen omaksuminen nouseekin entistä tärkeämpään rooliin tulevaisuudessa tekoälyä käytettäessä.

Todennäköisesti tekoäly tulee korvaamaan tiettyjä taitoja, eikä niiden opiskeleminen ole enää ohjelmoinnin kannalta relevanttia. Esimerkiksi ohjelmointikielät ovat parhaillaan todella korkean tason kieliä, eikä monellekaan ohjelmoijalle ole enää relevanttia opiskella matalan tason kieliä, kuten C-kieltä. Myös staattiset ohjelmistoautomaatiot ovat jo pitkään helpottaneet ohjelmistojen testausta.

Lukuisat ohjelmistokehittäjät ovat jo ottaneet tekoälytyökaluja käyttöönsä. Tästä kertoi katsauksessa tarkasteltu tutkimus, jossa yli 70 % vastanneista kertoi käyttävänsä tai käyttäneensä tekoälytyökaluja. Yritysten olisi tästäkin syystä tärkeää reagoida tekoälyyn liittyvien ohjeiden ja säädösten luomiseen, sillä tekoälyn käytön estäminen yrityksen työasemalta voi olla erittäin haastavaa. Todennäköisesti kehittäjät löytävät aina tavan niin halutessaan. Liika esteleminen ja asian huomioimattomuus voi siis pahimmissa tapauksessa heikentää työntekijäkokemusta, mutta ennen kaikkea vaarantaa yrityksen tietoturvan, kun tekoälyyn liittyvistä pelisäännöistä ei ole sovittu.

Useimmat ohjelmistokehittäjät kokivat tekoälypohjaisten työkalujen nopeuttavan tehtävien valmistumista. Tekoäly kykenee esimerkiksi täydentämään ja generoimaan koodia automaattisesti tai syötteiden perusteella. Tämän lisäksi se pystyy ehdottamaan tiettyjä funktioita, jos kehittäjä ei niitä muista. Tämän lisäksi tekoäly voi selittää kehittäjälle koodia ja tuoda uusia parannusehdotuksia. Suurin osa kehittäjistä ei kokenut tekoälyä uhkana omalle työlleen vaan ennemmin työtä tukevana avustajana. Tutkimusten mukaan etenkin työkalun käyttäminen vähensi pelkoa oman työn menettämisestä tekoälylle nyt ja tulevaisuudessa. Tekoälyn kehitystä kuitenkin hieman pelättiin etenkin tulevaisuuden osalta.

Tekoälyn kypsyttä kuvaa hyvin sille annettujen tehtävien tulokset. Toteutettujen tutkimusten mukaan tekoäly kykenee olemaan jopa 80 % aloittavaa kehittäjää nopeampi. Tehtävien ratkaisuprosentti kuitenkin heikkenee merkittävästi silloin, kun tehtävien vaikeustaso nousee. Jo hyppy helpoista tehtävistä keskitason tehtäviin laskee onnistumista kymmeniä prosentteja. Näiden testien perusteella tekoälyä on suhteellisen turvallista verrata kehittäjään, jolla kokemusta löytyy korkeintaan noin viisi vuotta. Myös tietyt rajoitukset tekoälytyökaluissa haastavat kehittäjiä, kuten esimerkiksi tekstien pituudet. Tähänkin on kuitenkin jo löydetty ratkaisuja kehittäjien keskuudessa, mutta tekoälyn kehittyminenkin varmasti auttaa tulevaisuudessa näissä ongelmissa.

Ongelmien korjaaminen on myös yksi haastavimmista tehtävistä ohjelmistokehityksessä. Tekoölyn avulla ongelmien paikallistamista pystyttiin nopeuttamaan, sillä tekoöly kykenee käsittelemään laajasti eri tietolähteitä ja luomaan näiden pohjalta analyyskejä. Tekoölyn hyötyjä oli tutkittu myös refaktoroinnin osalta. Tulokset olivatkin yllättävän positiiviset, sillä syötettyjen koodinpätkien refaktorointi onnistui lähes poikkeuksetta. Saadut hyödyt vaihtelivat laidasta laitaan, mutta välillä saaduista tuloksista saatiin myös tehokkuutta parantavia asioita.

Tekninen velka on myös aina haastanut yrityksiä ja kehittäjiä. Sovellusten ylläpitäminen on kallista ja usein tuloksiin on päästävää mahdollisimman nopeasti ja halvalla. Tekoölyn avulla voidaan helpottaa teknisen velan tunnistamista ja parhaillaan korjata ongelmia. Yhden kirjallisuuskatsauksessa mukana olleen tutkimuksen mukaan testeissä tekoöly kykeni korjaamaan jopa yli 55 % teknisestä velasta. Myös yhtenä mielenkiintoisena käyttökohteena voidaan todeta tekoölyn kyky analysoida nykyistä koodia ja luoda ennusteita siitä, miten tekninen velka pahenee tulevaisuudessa, jos asialle ei tehdä mitään. Tekoöly kykenee tarvittaessa myös korjaamaan dokumentoinnin teknistä velkaa.

Yhtenä merkittävimmästä tekoölyn hyödyntämiskohteista oli automaatiotestauksen kehittäminen tekoölyn avulla. Testien perusteella tekoöly kykeni parhaimmillaan luomaan yli 70 % testikattavuuden. Myös testien luettavuus oli riittävällä tasolla, mikä mahdollisti sen, että kehittäjät pystyisivät tulevaisuudessa ylläpitämään testejä ja tekemään pieniä muutoksia. Tekoölyn avulla voidaan siis poistaa myös testauksen pullonkauloja ja näin vähentää läpimenoaikoja.

Tekoölypohjaisten työkalujen vertailussa oli selkeitä eroja käyttäjien määrän suhteen. Pääsääntöisesti kolme käytetyintä tekoölytyökalua olivat ChatGPT, GitHub Copilot sekä TabNine. ChatGPT:n ja GitHub Copilotin sijoitus käyttäjämäärissä vaihteli tutkimusten välillä, mutta lähtökohtaisesti GitHub Copilottia voidaan pitää tämän hetken markkinajohtajana tekoölytyökalujen suhteen ohjelmoinnissa.

Kuten aikaisemmin mainittiin, tekoölyn käyttö on jo todennäköisesti hyvin arkipäivää monessa yrityksessä, halusi yritys sitä tai ei. Kehittäjillä on kuitenkin yhä suurempi vastuu tekoölyn tuottamasta koodista, sillä harmittomien virheiden lisäksi se voi sisältää myös tietoturvariskejä. Tutkimusten mukaan tekoöllylle annetun syötteen kanssa tulee olla hyvin tarkkana, sillä tekoöly tuottaa

juuri sitä, mitä kehittäjä pyytää. Generoitu koodi voi myös sisältää kutsuja rajapintoihin tai kolmannen osapuolen koodia, mitä käyttäjä ei halua tai hänen ei kannata käyttää. Myös huoli immateriaalioikeuksista nousee esille monissa kyselyissä.

Tekoälyn kypsyyden ohjelmistokehityksessä voidaan siis tänä päivänä todeta olevan riittävällä tasolla. Kirjallisuuskatsauksen tuloksien perusteella voidaan selkeästi todeta, että tekoälyn käyttö erilaisissa ohjelmistotehtävissä tehostaa ja nopeuttaa koodin tuottamista. Myös laadun parantamiseen tekoälyllä on merkittävä rooli, esimerkiksi nopeuttamalla uusien testien luomista ja parantamalla niiden laatua. Yritysten on tällä hetkellä erittäin suotavaa lähteä tutkimaan tekoälytyökalujen hyödyntämistä omassa ohjelmistokehityksessään, ennen kuin kilpailevat yritykset ottavat kehityksessä liian pitkän etumatkan tekoälyä hyödyntäen. Riskit ovat aina olemassa, mutta riittävällä ohjeistuksella mitigoitavissa, jolloin tekoälyn hyödyt ohjelmoinnissa kasvavat arvoon mittamattomaan.

8.4 Jatkotutkimusehdotukset

Kirjallisuuskatsauksen ajankohta tekoälyn hyödyntämiselle ohjelmointikehityksessä oli erittäin onnistunut. Tietoa aiheesta löytyy jo erittäin kattavasti, joten yhteen kokoava katsaus helpottaa asiasta kiinnostuneiden perehtymistä aiheeseen. Tekoälyn hyödyntämisen murros tulee kuitenkin jatkumaan vielä pitkään, minkä takia julkaistujen tutkimusten relevanttius heikkenee hyvin nopeasti. Vastaavan kirjallisuuskatsauksen tekeminen ei välttämättä tuo lisäarvoa vielä vuoden tai kahden sisällä, mutta mahdollisesti sen jälkeen ala on edennyt jälleen seuraavalle tasolle. Tätä odotellessa voisi olla hyvä tarkastella tekoälyn hyödyntämistä eri osa-alueilla vielä tarkemmin. Monesta alueesta kuten koodin generoimisesta tai testien luomisesta löytyy jo hyviä alustavia tutkimuksia, mutta syvällisempi tutkiminen ja eri työkalujen vertailu voisi tuoda käyttäjille sekä yrityksille lisäarvoa.

On myös valitettavasti mahdollista, että tämän hetken tekoälyinnostuksen aiheuttamana sen haittojen tutkiminen jää liian pieneen rooliin. Tutkimuksissa nousi usein esille satunnaisia huolia eri näkökulmista katsottuna, mutta lähtökohtaisesti niiden rooli oli hyvin pieni ja hyötyjen katsottiin kuittaavan potentiaaliset haitat. Näiden tarkempi tarkastelu voisi kuitenkin olla hyvin ajankoh- taista.

Tekoälyn hyödyntäminen oppimisessa tulee olemaan myös todennäköisesti hyvin ajankohtaista. Tätä aihetta voidaan varmasti tarkastella niin hyötyjen kuin haittojen näkökulmasta, mutta on lähes sanomattakin selvää, että siihen reagoiminen on pakollista niin oppimisen kuin uusien työelämätaitojen osalta.

Katsauksessa keskityttiin pääsääntöisesti tekoälyn hyödyntämiseen ohjelmoinnin parissa. Ohjelmistokehitys on kokonaisuudessaan kuitenkin paljon laajempi alue. Tekoälyn hyödyntämistä eri projektin vaiheissa olisi hyvä tutkia kokonaisvaltaisesti. On hyvin mahdollista, että työkalujen avulla voidaan löytää synergiaetuja ja luoda uusia malleja, miten toimia. Toiminnan kehittyminen rutiininomaisista tehtävistä haastavien ongelmien ja suunnittelun pariin on jo tällä hetkellä murrosvaiheessa, minkä takia ohjelmistokehityksen ammattilaisten, roolista riippumatta, on syytä hyväksyä tekoäly osaksi työkalupakkiaan ja muokata omaa toimintaansa sen mukaisesti.

Lähteet

A Definition of AI: Main Capabilities and Disciplines. 2019. Euroopan komissio. Viitattu 10.3.2024. https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=56341.

Abdullahi, A. 2024. Generative AI vs Predictive AI: What's the Difference? Viitattu 12.3.2024. <https://www.eweek.com/artificial-intelligence/generative-ai-vs-predictive-ai/>.

Agrawal, P., Wu, C. 2021. Is reinforcement learning right for your AI problem? MIT. Viitattu 12.3.2024. <https://professional.mit.edu/news/articles/reinforcement-learning-right-your-ai-problem>.

Algaze, B. 2017. Software is Increasingly Complex. That Can Be Dangerous. ExtremeTech. Viitattu 11.11.2023. <https://www.extremetech.com/extreme/259977-software-increasingly-complex-thats-dangerous>.

Artificial Intelligence (AI) vs. Machine Learning. N.d. Columbia engineering. Viitattu 21.11.2023. <https://ai.engineering.columbia.edu/ai-vs-machine-learning/>.

Biswas, S., Carson, B., Chung, V., Singh, S., Thomas, R. 2020. AI-bank of the future: Can banks meet the AI challenge. McKinsey & Company. Viitattu 3.3.2024. <https://www.mckinsey.com/industries/financial-services/our-insights/ai-bank-of-the-future-can-banks-meet-the-ai-challenge>.

Browne, R., Sigalos, M. 2023. A.I. has a discrimination problem. In banking, the consequences can be severe. Viitattu 27.4.2024. <https://www.cnbc.com/2023/06/23/ai-has-a-discrimination-problem-in-banking-that-can-be-devastating.html>.

Challenges Faced by Software Security Engineers. 2024. MoldStud. Viitattu 4.3.2024. <https://moldstud.com/articles/p-challenges-faced-by-software-security-engineers>.

Fernández, M. 2023. Code by the Numbers: How Many Lines of Code in Popular Programs, Apps and Video Games? Softonic. Viitattu 4.3.2024. <https://en.softonic.com/articles/programs-lines-code>.

Fink, A. 2005. Conducting research literature reviews: From the internet to the paper. Thousand Oaks, Calif : Sage Publications. Toinen painos.

Future of Jobs Report 2023. 2023. World Economic Forum. Viitattu 29.2.2024. https://www3.weforum.org/docs/WEF_Future_of_Jobs_2023.pdf.

Generative AI vs. Predictive AI: Unraveling the Distinctions and Applications. N.d. Neebal. Blogi. Viitattu 12.3.2024. <https://www.neebal.com/blog/generative-ai-vs.-predictive-ai-unraveling-the-distinctions-and-applications>.

Gruetzemacher, R. 2022. The power of Natural Language Processing. Harvard Business Review. Viitattu 21.11.2021. <https://hbr.org/2022/04/the-power-of-natural-language-processing>.

Grynbaum, M., Mac, R. 2023. The Times Sues OpenAI and Microsoft Over A.I. Use of Copyrighted Work. The New York Times. Viitattu 27.4.2024. <https://www.nytimes.com/2023/12/27/business/media/new-york-times-open-ai-microsoft-lawsuit.html>.

Haziqa, S. 2023. Koneoppiminen vs. syväoppiminen – keskeiset erot. Unite.ai. Viitattu 21.11.2021. <https://www.unite.ai/fi/machine-learning-vs-deep-learning-key-differences/>.

How does Artificial Intelligence work? N.d. Innoplexus. Blogi. Viitattu 13.3.2024. <https://www.innoplexus.com/blog/how-artificial-intelligence-works/>.

Hyvä tieteellinen käytäntö. 2023. Tutkimuseettinen neuvottelukunta. Viitattu 18.4.2024. <https://tenk.fi/fi/tiedevilppi/hyva-tieteellinen-kaytanta-htk>.

iOS Apple App Store Statistics and Trends. 2024. 42matters. Viitattu 4.3.2024. <https://42matters.com/ios-apple-app-store-statistics-and-trends>.

Inside Speech to Note: How AI Understands Your Voice and Converts it to Text. 2023. Medium. Viitattu 15.3.2023. <https://speechtonote.medium.com/inside-speech-to-note-how-ai-understands-your-voice-and-converts-it-to-text-7b06be8d99a0>.

Kangasniemi, T. 2020. Työttömyystukijärjestelmä USA:ssa hukkuu hakemuksiin – rakennettu cobolilla ja osajista on huutava pula. Viitattu 12.11.2023. <https://www.tekniikkatalous.fi/uutiset/tyotomyyystukijarjestelma-usassa-hukkuu-hakemuksiin-rakennettu-cobolilla-ja-osajista-on-huutava-pula/3f59e992-268b-4016-a8ea-1d3cbb53e329>.

Keserer, E. 2023. Deep Learning vs Reinforcement Learning: Key Differences and Use Cases. Akkio. Viitattu 12.3.2024. <https://www.akkio.com/post/deep-learning-vs-reinforcement-learning-key-differences-and-use-cases>.

Kelly, J. 2024. Klarna's AI Assistant Is Doing The Job Of 700 Workers, Company Says. Forbes. Viitattu 27.4.2024. <https://www.forbes.com/sites/jackkelly/2024/03/04/klarnas-ai-assistant-is-doing-the-job-of-700-workers-company-says/>.

Kivimäki, A. 2023. Tekoäly ei ajattele puolestasi. Tiede, 43, 14, 52–57.

Manning, C. 2020. Artificial Intelligence Definitions. Stanford University, Human-Centered Artificial Intelligence. Viitattu 21.11.2023. <https://hai.stanford.edu/sites/default/files/2020-09/AI-Definitions-HAI.pdf>.

Market Research Report. 2024. Fortune Business Insights. Viitattu 14.3.2024. <https://www.fortunebusinessinsights.com/industry-reports/natural-language-processing-nlp-market-101933>.

Natural Language Processing (NLP) and its Role in Advancing Conversational Artificial Intelligence. 2023. IIT Kanpur. Viitattu 14.3.2024. <https://ifacet.iitk.ac.in/knowledge-hub/artificial-intelligence/natural-language-processing-nlp-and-its-role-in-advancing-conversational-artificial-intelligence/>.

Numminen, L. 2023. Mitä ovat suuret kielimallit ja miten ne toimivat? Finnishup. Viitattu 14.3.2024. <https://www.finnishup.com/mita-ovat-suuret-kielimallit-ja-miten-ne-toimivat/>.

Page, M., McKenzie, J., Bossuyt, P., Boutron, I., Hoffmann, T., Mulrow, C., Shamseer, L., Tetzlaff, J., Akl, E., Brennan, S., Chou, R., Glanville, J., Grimshaw, J., Hróbjartsson, A., Lalu, M., Li, T., Loder, E., Mayo-Wilson, E., McDonald, S., McGuinness, L., Stewart, L., Thomas, J., Tricco, A., Welch, V., Whiting, P. & Moher, D. 2021. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372, 71. Viitattu 7.4.2024. <https://janet.finna.fi>, ProQuest.

Panchal, D. 2023. Artificial Intelligence in Electronics: Building a Smarter Future. Viitattu 1.3.2024. <https://www.upwithtech.com/post/artificial-intelligence-in-electronics#:~:text=Here%20are%20some%20of%20the,to%20enhance%20productivity%20and%20efficiency.>

Peachey, K. 2021. Convicted Post Office workers have names cleared. Viitattu 27.4.2024. <https://www.bbc.com/news/business-56859357>.

Programming Languages Products. N.d. Viitattu 4.3.2024. <https://enlyft.com/tech/programming-languages>.

Riemer, S., Strauß, M., Rabener, E., Kwong Bickford, J., Hilbers, P., Kalra, N., Kapoor, A., King, J., Palumbo, S., Pardasani, N., Pauly, M., Rulf, K., Widowitz, M. 2023. A Generative AI Roadmap for Financial Institutions. BCG. Viitattu 12.3.2024. <https://www.bcg.com/publications/2023/a-generative-ai-roadmap-for-fis>.

Sharma, A. 2024. Top Google Play Store Statistics 2024 You Must Know. Blogi. Viitattu 3.4.2024. <https://appinventiv.com/blog/google-play-store-statistics/#:~:text=Total%20Number%20of%20Apps%20Released%20per%20Week%20and%20Month&text=To%20be%20precise%2C%203%2C739,increased%20number%20of%20mobile%20apps.>

Stack OverFlow: How We Do Deployment – 2016 Edition. 2016. Stack Overflow. Blogi. Viitattu 4.3.2024. <https://stackoverflow.blog/2016/05/03/stack-overflow-how-we-do-deployment-2016-edition/>.

Tekstinkäsittelijöitä koskevat yhteistoimintaneuvottelut ovat alkaneet. 2023. Pohjois-Pohjanmaan hyvinvointialue. Viitattu 27.4.2024. <https://pohde.fi/ajankohtaista/uutiset/tekstinkasittelijoita-koskevat-yhteistoimintaneuvottelut-ovat-alkaneet/>.

Toivonen, H. 2023. Mitä tekoäly on? : 100 Kysymystä ja vastausta. Ensimmäinen painos. Helsinki : Kustannusosakeyhtiö Teos. Viitattu 2.12.2023. <https://janet.finna.fi>, Ellibslibrary.

Tuominen, H., Neittaanmäki, P. 2019. Tekoälyn perusteita ja sovelluksia. Jyväskylän yliopisto. Viitattu 3.12.2023. <https://tim.jyu.fi/print/kurssit/tie/tiep1000/tekoalyn-sovellukset/kirja>.

Vaughan-Nichols, S. 2019. Cobol turns 60: Why it will outlive us all. Viitattu 12.11.2023. <https://www.zdnet.com/article/cobol-turns-60-why-it-will-outlive-us-all/>.

Velarde, G. 2020. Artificial Intelligence and its impact on the Fourth Industrial Revolution: A Review. Viitattu 27.4.2024. <https://arxiv.org/abs/2011.03044>.

Vilkka, H. 2023. Kirjallisuuskatsaus metodina, opinnäytetyön osana ja tekstilajina. Ensimmäinen painos. Art House Oy. Viitattu 2.4.2024. <https://janet.finna.fi>. Ellibslibrary.

Website Speed Stats: Does Your Site Load in 3 Seconds? 2022. Digital. Viitattu 3.4.2024. <https://digital.com/website-speed-statistics/>.

What are Large Language Models (LLM). N.d. AWS. Viitattu 10.3.2024. <https://aws.amazon.com/what-is/large-language-model/>.

What is decision tree? N.d. IBM. Viitattu 13.3.2024. <https://www.ibm.com/topics/decision-trees>.

What Is the Necessity of Bias in Neural Networks? N.d. Turing. Viitattu 13.3.2024. <https://www.turing.com/kb/necessity-of-bias-in-neural-networks>.

What's New in Artificial Intelligence from the 2023 Gartner Hype Cycle. 2023. Gartner. Viitattu 4.3.2024. <https://www.gartner.com/en/articles/what-s-new-in-artificial-intelligence-from-the-2023-gartner-hype-cycle>.

Liitteet

Liite 1. Kirjallisuuskatsauksessa käytetyt lähteet

2023 Developer Survey. 2023. Stack Overflow. Viitattu 17.4.2024. <https://stackoverflow.com/labs/developer-sentiment-ai-ml/>.

Batchu, A., Walsh, P., Scheibmeir, J., Nair, H., Bhat, M., Egiazarov, T. 2024 Innovation Guide for AI coding Assistants. Gartner. Viitattu 16.4.2024.

Bayri, V., Demirel, E. 2023. AI-Powered Software Testing: The Impact of Large Language Models on Testing Methodologies. IEEE Xplore. Viitattu 14.4.2024. <https://doi.org/10.1109/IISEC59749.2023.10391027>.

Binta, S., Kaushal, S., Pandi, S. 2023. Artificial Intelligence for Technical Debt Management in Software Development. ProQuest. Viitattu 14.4.2024. <https://doi.org/10.48550/arXiv.2306.10194>.

Bull, C., Kharuffa, A. 2023. Generative Artificial Intelligence Assistants in Software Development Education: A Vision for Integrating Generative Artificial Intelligence Into Educational Practice, Not Instinctively Defending Against It. IEEE Xplore. Viitattu 15.4.2024. <https://doi.org/10.1109/MS.2023.3300574>.

DePalma, K., Miminoshvili, I., Henselder, C., Moss, K., AlOmar, E. 2024. Exploring ChatGPT's code refactoring capabilities: An empirical study. ScienceDirect. Viitattu 15.4.2024. <https://doi.org/10.1016/j.eswa.2024.123602>.

Dhyani, P., Nautiyal, S., Negi, A., Dhyani, S., Chaudhary, P. 2024. Automated API Docs Generator using Generative AI. IEEE Xplore. Viitattu 15.4.2024. <https://doi.org/10.1109/SCEEC561402.2024.10482119>.

Kuhail, M., Mathew, S., Khalil, A., Berengueres, J., Shah, S. 2024. "Will I be replaced?" Assessing ChatGPT's effect on software development and programmer perceptions of AI tools. ScienceDirect. Viitattu 15.4.2024. <https://doi.org/10.1016/j.scico.2024.103111>.

Liang, J., Yang, C., Myers, B. 2023. A Large-Scale Survey on the Usability of AI Programming Assistants: Successes and Challenges. ProQuest. Viitattu 15.4.2024. <https://doi.org/10.48550/arXiv.2303.17125>.

Mehmood, S., Janjua, U., Ahmed, A. 2023. From Manual to Automatic: The Evolution of Test Case Generation Methods and the Role of GitHub Copilot. IEEE Xplore. Viitattu 15.4.2024. <https://doi.org/10.1109/FIT60620.2023.00013>.

Measuring Impact: The GenAI Code Report. 2024. LinearB. Viitattu 16.4.2024. <https://linearb.io/resources/measuring-impact-the-genai-code-report>.

Moon, J., Yang, R., Cha, S., Kim, S. 2023. chatGPT vs Mentor : Programming Language Learning Assistance System for Beginners. IEEE Xplore. Viitattu 16.4.2024. <https://doi.org/10.1109/ICSECS58457.2023.10256295>.

Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M. 2023. The Impact of AI on Developer Productivity: Evidence from GitHub Copilot. ProQuest. Viitattu 16.4.2024. <https://doi.org/10.48550/arXiv.2302.06590>.

Purwoko, J., Abdullah, T., Wijaya, B., Gunawan, A., Saputra, K. 2023. Analysis ChatGPT Potential: Transforming Software Development with AI Chat Bots. IEEE Xplore. Viitattu 16.4.2024. <https://doi.org/10.1109/IconNECT56593.2023.10327087>.

Shi, Y., Sakib, N., Shahriar, H., Lo, D., Chi, H., Qian, K. 2023. AI-Assisted Security: A Step towards Reimagining Software Development for a Safer Future. IEEE Xplore. Viitattu 16.4.2024. <https://doi.org/10.1109/COMPSAC57700.2023.00142>.

Liite 2. Valittujen aineistojen laadunarviointi

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
2023 Developer Survey. 2023	Kysely ohjelmistokehittäjille, missä selvitetään tekoälyn hyödyntämistä. (5)	Kyselytutkimus käy ilmi heti alusta, mutta asiaa ei paljoa enempää aukaista tässä vaiheessa. (2)	Viitekehys on olematon ja valittuja kysymyksiä ei perustella. (1)	Johtopäätökset tuotu selkeästi esille useammalla eri tavalla. (5)	Vastaajamäärä tekee kyselystä melko puolueettoman, mutta taustalla on kuitenkin yritys, joka tarjoaa tekoälyyn liittyviä palveluita. (3)	Alkuperäinen kyselytutkimus ilman lähteitä. (3)	3,2 / 5,0
Batchu, A., Walsh, P., Scheibmeir, J., Nair, H., Bhat, M., Egiazarov, T. 2024	Analyysi tekoälytyökalujen hyödyntämisessä ohjelmoinnissa. (5)	Ei juuri avattu, mutta alusta käy ilmi, että kyseessä on analyysi aiheesta. (2)	Viitekehys kuvattu heikosti. (2)	Johtopäätökset on tuotu selkeästi esille loogisessa järjestyksessä. (5)	Yritys tarjoaa aiheeseen liittyviä palveluita, mutta on samalla yksi arvostetuimmista ja seuratuimmista yrityksistä uusien teknologioiden saralla. (4)	Alkuperäinen analyysi ja lähteet kuvattu riittävällä tasolla. (4)	3,7 / 5,0

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteiden luotettavuus	Kokonaisarvosana
Bayri, V., Demirel, E. 2023	Suurten kieli-mallien hyödyntäminen ohjelmistojen testaamisessa. (5)	Selkeästi avattu, että kyseessä oli tutkimus, jossa toteutettiin konseptitodistus (PoC). (4)	Viitekehys on selkeästi tuotu esille. Kehystä on tuettu aikaisemmilla tutkimuksilla. (4)	Tulokset olivat melko ohuesti avattuina, vaikka lopputulos oli selkeä. (3)	Tutkimus antaa kuvan puolueettomuudesta, mutta ei tarkemmin avaa miksi tutkimus on haluttu toteuttaa. (3)	Alkuperäisyys hyvä, mutta lähteiden luotettavuus ja laatu vaihtelevat laidasta laitaan. (2)	3,3 / 5,0
Binta, S., Kaushal, S., Pandi, S. 2023	Tutkia tekoälyn hyödyntämistä teknisen velan vähentämisessä. (5)	Kirjallisuuskatsaus olemassa olevaan tutkimukseen tekoälytyökalujen hyödyntämisessä teknisen velan vähentämiseksi. (5)	Viitekehys on tuotu esille selkeästi ja johdonmukaisesti. (5)	Heikohko. Johtopäätöksissä toistettiin katsoituksen kappaleiden sisältöä. (2)	Yliopiston teettämä tutkimus, mikä keskittyy analysoimaan muiden tutkimuksia. (4)	Tutkimus perustuu muiden lähteisiin, mutta koa hyvin tiedon yhteen. Lähteet ovat laadukkaan oloisia ja hyvin kuvattu. (4)	4,5 / 5

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
Bull, C., Kharuffa, A. 2023	Tutkimuksessa selvitetään, että kannattaako tekoälyä hyödyntää opettamiseen. (5)	Tutkimusmenetelmät tulevat tekstistä ilmi, mutta eivät välittömästi. (3)	Viitekehys tulee ilmi heti alussa hyvin selkeästi. (5)	Johtopäätökset tuodaan hyvin ja perustellusti esille. (5)	Yliopiston opettajien teettämä tutkimus ja puolueellisuudesta ei ole viitteitä tekstissä. (5)	Lähteet vaikuttavat laadukailta ja tulevat selkeästi ilmi. Kyselytutkimus tuo tutkimukseen alkuperäisyyttä. (4)	4,5 / 5
DePalma, K., Mimi-noshvili, I., Henselder, C., Moss, K., Alomar, E. 2024	Tutkimuksessa pyrittiin selvittämään tekoälyn käyttöä koodin refaktoroinnissa. (5)	Tutkimusmenetelmät on kuvattu erittäin hyvin ja tarkkaan. Tutkimuskysymykset nousevat selkeästi esille. (5)	Viitekehys on hyvin selkeä. (5)	Johtopäätökset käydään hyvin kattavasti lävitse ja niitä verrataan myös aikaisempiin tutkimuksiin. (5)	Tutkimuksesta ei nouse esille syitä epäillä puolueettomuutta. (5)	Tutkimus tuo selkeästi uutta tietoa tekoälyn hyödyntämisessä refaktoroinnissa ja lähteet ovat hyvin esiteltyinä ja hyödynnettyinä. (5)	5,0 / 5,0

Lähde	Julkaisun tar- koitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteiden luotettavuus	Kokonaisar- vosana
Dhyani, P., Nautiyal, S., Negi, A., Dhyani, S., Chaudhary, P. 2024	Tutkia API do- kumentoinnin generointia te- koälyn avulla. (4)	Epäselvä ja hyppivä. Kirjallisuuskatsauk- sesta käytännön tes- taukseen. (2)	Viitekehys käy teks- tistä hyvin esille. (5)	Hyvin esitetty, mutta melko pintapuolisesti. (3)	Hieman epämääräi- nen, mutta ei mitään syytä epäillä puolueet- tomuutta. (3)	Tekijöiden käytän- nönkoe tuo mukaan al- kuperäisyyttä ja lähteet ovat hyvin ja selkeästi kuvattuina. (4)	3,5 / 5,0
Kuhail, M., Mathew, S., Khalil, A., Ber- engueres, J., Shah, S. 2024	Tutkia teko- älyn hyödyntä- mistä ohjel- moinnissa kehittäjän vas- tauksien pe- rusteella. (5)	Hyvin ja selkeästi ku- vattu. Kyseessä kyse- lytutkimus. (5)	Viitekehys käy teks- tistä hyvin esille. (5)	Erittäin hyvin esitelty ja mo- nipuolisesti ku- vioiden ja tau- lukoiden avulla. Moni- puolista ana- lyysiä. (5)	Tutkimus vaikuttaa puolueettomalta. Lo- pussa on maininta yli- opiston osittaisesta ra- hoituksesta. (5)	Hyvä ja uusi kulma tut- kia tekoälyä. Lähteet vaikuttavat laadukkailta ja ovat hyvin kuvat- tuina. (5)	5,0 / 5,0

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
Liang, J., Yang, C., Myers, B. 2023	Tutkia tekoälyn käytettävyyttä. (5)	Tutkimusmenetelmät käyvät hyvin ilmi alusta alkaen. (5)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset on esitelty hyvin ja selkeästi taulukoita ja kuvioita apuna käyttäen. (5)	Tutkimus vaikuttaa puolueettomalta. (5)	Ensimmäinen löytämäni tutkimus, missä käsitellään tekoälyn käyttöä myös käytettävyyden näkökulmasta. (5)	5,0 / 5,0
Mehmood, S., Janjua, U., Ahmed, A. 2023	Tutkia tekoälyn hyödyntämistä testitapauksien luomisessa ja suorittamisessa. (5)	Tutkimusmenetelmät on kuvattu hyvin tutkimuksessa. (4)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset on esitetty riittävällä tasolla, mutta pientä parantamista selkeyden suhteen jäi. (3)	Yliopiston teettämä tutkimus, mistä puolueellisuutta ei nouse esille. (4)	Hyvä ja tuore kulma tekoälyn testausosaamisen tutkimiseen. Lähteet ovat myös hyvin esiteltyinä. (4)	4,2 / 5,0

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
Measuring Impact: The GenAI Code Report. 2024	Tutkimus perehtyy tekoälyn hyötyjen mittaamiseen ohjelmistokehityksessä. (5)	Kyselytutkimus, mutta tämä tulee hiekkamäen heikosti ilmi. (2)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset käydään hyvin tiivistetyksi kuvien avulla lävitse. (3)	Pohjalla oman tuotteen edustaminen, mikä vie vahvasti pohjan puolueettomuudelta. (2)	Alkuperäisyys tyydyttävä, mutta lähteitä ei ole mainittu. (2)	3,2 / 5,0
Moon, J., Yang, R., Cha, S., Kim, S. 2023	Tutkimuksessa verrattiin ChatGPT-tekoälyn ja mentorin vastauksia opiskelijalle. (5)	Menetelmät on kuvattu hyvällä tasolla. (4)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset on kuvattu hiekkamäen haastavasti. (3)	Yliopiston teettämä tutkimus, mistä puolueellisuutta ei nouse esille. (4)	Hyvä ja mielenkiitoinen tutkimus voidaanko tukeminen korvata tekoälyn avulla. Lähteissä viittauksia omiin aikaisempiin tutkimuksiin. (3)	4,0 / 5,0

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
Peng, S., Kalliamvakou, E., Cihon, P., Demirer, M. 2023	Tutkia tekoälyn tehokkuutta ohjelmoinnissa. (5)	Hyvin ja selkeästi kuvattut tutkimusmenetelmä. (4)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset ovat kuvattu selkeästi, mutta melko pintapuolisesti. (3)	Viitteitä yrityksen rahoituksesta. (2)	Mahdollisesti ensimmäisiä empiirinen tutkimus, missä verrataan eri ryhmiä tekoälyn kanssa ja ilman. Lähteet vaikuttavat luotettavilta. (5)	4,0 / 5,0
Purwoko, J., Abdulah, T., Wijaya, B., Gunawan, A., Saputra, K. 2023	Tutkia tekoälyn hyödyntämistä eri ohjelmointitehtävissä. (5)	Tutkimusmenetelmät on kuvattu hyvin ja testien kooditkin ovat saatavilla. (4)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset tulevat hyvin esille, mutta kaipaivat hiekkaisempää analyysia. (3)	Yliopiston tekemä tutkimus, mistä ei ilmene puolueellisia piirteitä. (5)	Vastaavia testejä on varmasti tehty aikaisemminkin, mutta hyvä ja selkeä esitys. Lähteet ovat myös luotettavan oloisia. (3)	4,5 / 5,0

Lähde	Julkaisun tarkoitus	Tutkimusmenetelmien kuvaus	Viitekehys	Johtopäätösten selkeys	Puolueettomuus	Alkuperäisyys ja lähteen luotettavuus	Kokonaisarvosana
Shi, Y., Sakib, N., Shahriar, H., Lo, D., Chi, H., Qian, K. 2023	Tutkia tekoälyn hyödyntämistä ohjelmoinnin tietoturvassa. (5)	Selkeät, mutta melko pintapuolisesti kuvattuna. (3)	Viitekehys käy tekstistä hyvin esille. (5)	Johtopäätökset tulevat heikosti esille. (2)	Puolueettomuudesta ei ole viitteitä. (5)	Mielenkiintoinen kulma ja testi tekoälyn hyödyntämiseen. Lähteitä melko vähän, mutta ne ovat luotettavan oloisia. (4)	4,0 / 5,0