

Lauri Kurki

# KATSAUS VIBE-KOODAUKSEEN

Informaatioteknologian ja viestinnän tiedekunta  
Kandidaattitutkielma  
Joulukuu 2025

# TIIVISTELMÄ

Lauri Kurki: Katsaus vibe-koodaukseen  
Kandidaattitutkielma  
Tampereen yliopisto  
Tietojenkäsittelytieteiden tutkinto-ohjelma  
Joulukuu 2025

---

Vibe-koodaus on vuoden 2025 keväällä kehitetty, alun perin sosiaalisesta mediasta lähtöisin oleva ilmiö. Vibe-koodauksella ei ole vakiintuneita määritelmiä, mutta joitakin esitettyjä määritelmiä ovat muun muassa prosessi, jossa suuri osa tai koko kehitystyö delegoidaan tekoälylle, sekä kehittäjän aikomusten ja tarkoitusten välittäminen sekä niiden kehittäminen keskustelemalla tekoälyn kanssa. Vibe-koodaus voi hyödyttää alan ammattilaisia ja opiskelijoita muun muassa nopeuttamalla oppimista, prototyyppointia ja vapauttamalla kognitiivisia resursseja. Organisaatiot taas voivat hyötyä vibe-koodaamisesta resurssien uudelleen jakamisen helpottumista auttavana tekijänä sekä innovaatio- ja reagointinopeuden parantajana. Uhkakuvia yksilöiden kannalta ovat ohjelmointitaitojen rapautuminen sekä koodin laadun ongelmat. Organisaatioiden näkökulmasta eritoten tietoturvaongelmat sekä vastuunjakoon sekä eettisiin seikkoihin liittyvät ongelmat nousevat pinnalle.

Avainsanat: vibe-koodaus, generatiivinen tekoäly, ohjelmistokehitys

Tämän julkaisun alkuperäisyys on tarkastettu Turnitin OriginalityCheck –ohjelmalla.

# TEKOÄLYN KÄYTTÖ OPINNÄYTTEESSÄ

Opinnäytteessäni on käytetty tekoälysovelluksia:

- Ei
- Kyllä

Ilmoitukseni mukaan olen käyttänyt opinnäytteessäni tutkielmaprosessin aikana seuraavia tekoälysovelluksia: -

Tekoälysovellusten nimet ja versiot: -

Käyttötarkoitus: -

Osiot, joissa tekoälyä on käytetty: -

Olen tietoinen siitä, että olen täysin vastuussa koko opinnäytteeni sisällöstä, mukaan lukien osat, joissa on hyödynnetty tekoälyä, ja hyväksyn vastuun mahdollisista eettisten ohjeiden rikkomuksista.

# SISÄLLYSLUETTELO

<b>1</b>	<b>Johdanto .....</b>	<b>1</b>
<b>2</b>	<b>Vibe-koodaus.....</b>	<b>2</b>
2.1	Vibe-koodaus käsitteenä	3
2.2	Motivaatiotekijöitä Vibe-koodaukseen	4
<b>3</b>	<b>Vibe-koodauksen mahdollisuuksia .....</b>	<b>6</b>
3.1	Mahdollisuudet yksilön ja alan ammattilaisten kannalta	6
3.2	Mahdollisuudet organisaation ja laajemman ekosysteemin kannalta	8
<b>4</b>	<b>Vibe-koodauksen riskit .....</b>	<b>9</b>
4.1	Riskit ohjelmoijan ja koodin laadun kannalta	9
4.2	Riskit organisaatioiden ja vastuun kannalta	11
<b>5</b>	<b>Yhteenveto.....</b>	<b>12</b>
<b>6</b>	<b>Lähdeluettelo.....</b>	<b>15</b>

## 1 Johdanto

Tutkielman tavoitteena on selvittää ja käydä läpi mitä vibe-koodaus on, miten se on kehittynyt ja miten se on vaikuttanut tai sen ennustetaan vaikuttavan esimerkiksi ohjelmistokehitykseen ammatillisessa tai koulutuksellisessa mielessä. Tekoäly on ollut pinnalla jo useamman vuoden niin sosiaalisessa mediassa kuin koulutuksen että ammatillisten organisaatioiden saralla. Moni opiskelija on saattanut hyödyntää erilaisia kielimalleja opiskelussaan ja huomannut niiden olevan toisaalta erittäin hyödyllisiä, mutta usein myös vievän aikaa ja keskittymistä pois oikeasta aivotyöstä ja oppimisesta.

Vibe-koodaaminen on uudehko termi, joka on noussut pinnalle alun perin vuoden 2025 helmikuussa, kun alan tutkija käytti termiä ensimmäisen kerran twiitissään (Geng et al., 2025). Siitä lähtien termi on ollut esillä eritoten sosiaalisessa mediassa ja alan yleisenä puheenaiheena. Moni alan ammattilainen näkee tekoälyavusteisen koodauksen sekä vibe-koodauksen valtavana mahdollisuutena, mutta alalta löytyy paljon myös sen epäilijöitä. Miten käy ohjelmistotalan työllistävyydelle, kun juniortason ohjelmointityön voi hoitaa tekoälyllä? Toisaalta onko tekoäly edes niin hyvä ohjelmoimaan itsenäisesti? Onko vaara sille, että organisaatioiden lähdekoodin laatu kärsii, jos se on yhä enenevässä määrin tekoälyn avulla tuotettua?

Esimerkiksi juuri alalla aloittavat opiskelijat ovat aiheen aallonharjalla. Markkinoille tulleet erinäiset keskusteluun pohjautuvat AI-ohjelmointiapurit voivat antaa apua ja voivat auttaa oppimaan keskeisiä konsepteja, mutta niiden toimivuus ja hyödyllisyys koetaan ristiriitaiseksi myös opiskelijoiden keskuudessa (Zhu & Van Brummelen, 2021).

Tutkielma pyrki vastaamaan seuraaviin tutkimuskysymyksiin: (1) Mitä Vibe-koodaus on ja mitä ominaispiireitä siihen liittyy, (2) Mitä potentiaalisia hyötyjä vibe-koodaamisella voi olla yksilöiden ja organisaatioiden kannalta ja (3) Mitä potentiaalisia haittoja vibe-koodaamisella voi olla yksilöiden ja organisaatioiden kannalta.

Tutkielma toteutettiin kirjallisuuskatsauksena. Aineistonhankinnassa on pyritty hyödyntämään mahdollisimman tuoreita ja monipuolisia lähteitä ja muodostamaan niiden pohjalta tarpeeksi kattava yleiskuva aiheesta ja sen taustoista.

Tutkielman aineistonhankinnassa on käytetty seuraavia hakutermejä:

- ”vibe-coding”
- ”vibe-programming”
- ”generative ai implications”

- ”vibe-coding in education”

Aineistonhankinnassa hyödynnettiin akateemisia julkaisualustoja sekä tietokantoja. Aineistonhankinnassa hyödynnettyjä tietokantoja ovat olleet:

- ACM Digital Library
- Computer Science Database (ProQuest)
- ScienceDirect (Elsevier)

Aihealueen uutuuden vuoksi vertaisarvioituja artikkeleita löydettiin hyvin rajallinen määrä, joten hakua on laajennettu kattamaan myös vielä tieteellisissä julkaisuissa julkaisemattomia artikkeleita. Tämän takia myös hakutyökalujen käyttöä laajennettiin myös seuraaviin:

- Google Scholar
- arXiv

Aiheen tuoreuden vuoksi myös aineistonhaussa on priorisoitu vuoden 2022 jälkeen julkaistuja artikkeleita. Lähteitä on pyritty löytämään kattavasti jokaiselta tutkielmassa tarkastellulta aihealueelta, vaikkakin osa lähteistä kattaakin suuria kokonaisuuksia. Ensimmäisten ja kattavampien lähteiden vastapainoksi on pyritty keräämään taustoittavia lähteitä ja eritoten näiden tapauksessa on pyritty priorisoimaan vertaisarvioituja julkaisuja.

Tutkielman luvussa kaksi esitellään ensin vibe-koodaus ja siihen liittyviä määritelmiä, sekä käydään lävitse eri motivaatiotekijöitä vibe-koodaamiseen hyödyntämiseen. Tämän jälkeen siirrytään tarkastelemaan vibe-koodaamisen etuja ja mahdollisuuksia yksilön ja organisaatioiden kannalta luvussa 3, jonka jälkeen käsitellään uhkakuvia, joita vibe-koodaus voi tuoda mukanaan yksilöille ja organisaatioille luvussa 4. Lopuksi tehdään yhteenveto käsitelystä aiheesta ja keskustellaan tuloksista.

## **2 Vibe-koodaus**

Vibe-koodaus on uusi, internetistä lähtöisin oleva termi. Tarkemmin ottaen se on saanut alkunsa twiitistä, jonka tekoälytutkija Anderj Karpathy kirjoitti helmikuussa 2025. Termin uutuuden, sekä ympäristön ja käyttökontekstin myötä sille on esitetty moninaisia eri määritelmiä, mutta vakiintunutta määritelmää tai termistöä ei ole vielä ehtinyt syntyään.

## 2.1 Vibe-koodaus käsitteenä

Yksi tapa käsitellä ja havainnollistaa vibe-koodausta ja sen metodologiaa, on määritellä se tarkoituksen välittämisen (engl. *intent mediation*) kautta (Meske et al., 2025). Perinteisesti ohjelmoijan kontekstissa on tämä tarkoitus tai idea toiminnallisuudesta tai ohjelmistosta täytynyt välittää selkeän formaalin syntaksin, usein ohjelmointikielen välityksellä (Meske et al., 2025). Generatiivisen tekoälyn nousun ja vibe-koodaus-ilmion syntymisen myötä on kehittäjille syntynyt uusia keinoja välittää aikomustaan ja suunnitelmiaan ohjelmiston kehittämisestä (engl. *intent mediation*). Perinteisesti ohjelmoijan on pitänyt sisäistää laajoja ohjelmistokehityksen konsepteja ja syntakseja, mutta generatiivinen tekoäly on tuonut kehittäjille mukanaan mahdollisuuden haluttuun lopputulokseen pääsemiseksi käyttää luonnollista kieltä ja vapaampaa keskustelua (Meske et al., 2025). Määritelmän mukaan vibe-koodauksen ytimessä onkin keskustelevuus ja kanssakäyminen tekoälyn kanssa. Toimintamallien muutos ja kehitys kohti luonnollista kieltä voidaan nähdä myös luonnollisena kehityskulkuna alalla. Nykypäivän ohjelmistokehitysalustat ja työkalut ovat jo ennen generatiivista tekoälyä tuoneet vaadittua abstraktiotasoa aina vain ylemmäs. Tietojenkäsittelyn alkuaikoina on vaadittu tietämys ja osaaminen ollut hyvinkin matalatasoista, ja ohjelmointikielten kuten C tai COBOL myötä, on abstraktiotaso liikkunut ylöspäin (Meske et al., 2025).

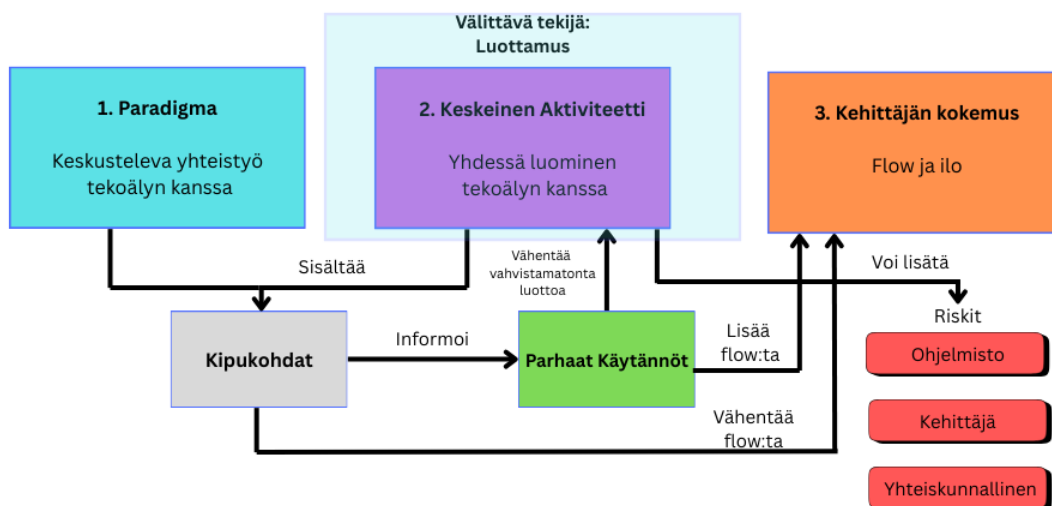
Joidenkin määritelmien mukaan vibe-koodaus voidaan määritellä myös vain prosessin kautta. Vibe-koodausprosessissa työ delegoidaan suurilta osin, tai jopa kokonaan generatiiviselle tekoälylle (Geng et al., 2025). Tekoälyä pyydetään kehotteiden avulla kirjoittamaan tai korjaamaan koodia, eikä sitä välttämättä tarkisteta ollenkaan, vaan luotetaan sen tuotoksiin. Vibe-koodauksesta voidaan myös tehdä rajanveto agenttimalleilla suoritettavaan koodaukseen (engl. *Agentic Coding*) (Sapkota et al., 2025). Termin takana oleva keskeinen ajattelumalli on tekoälyn itsenäinen tai puoli-itsenäinen työskentely, jossa tekoäly suorittaa monimutkaisempaa korkeamman tason suunnittelu- ja toteutustyötä. Tällaisessa toiminnassa ihmiskehittäjän rooli on pikemminkin arkkitehdin tai projektinjohdajan kaltainen. Vibe-koodauksessa ihmiskehittäjän rooli on kuitenkin käytännönläheisempi. Kehittäjä voi pyytää tekoälyä kirjoittamaan yksittäisen funktion, tai uudelleenkirjoittamaan luokkamäärittelyn (Sapkota et al., 2025). Vibe-koodauksessa toiminta on siis yhteistyökeskeisempää.

Vibe-koodaus myös eroaa aikaisemmista generatiivista tekoälyä hyödyntävistä ohjelmoinnin käytännöistä muun muassa filosofian ja kyselyiden tekemisen osalta (Sarkar & Drosos, 2025). Vibe-koodauksen yhteydessä kirjoitettavat kyselyt tekoälylle ovat usein huomattavasti pidempiä ja monimutkaisempia kuin ennen, ja ne voivat olla sisällöltään

myös huomattavan monipuolisia (Sarkar & Drosos, 2025). Kyselyt voivat vaihdella aina esteettisistä toiveista, teknisiin spesifikaatioihin ja laajoihin ideologiatason tavoitteisiin (Sarkar & Drosos, 2025). Kyselyt voivat myös sisältää varsinaista koodia, tai ulkoisia dokumenttilinkkejä, joita käytetään kontekstin lisäämiseen (Sarkar & Drosos, 2025). Aikaisemmat työkalut olivat myös riippuvaisia täsmällisistä tietojen syöttökentistä, kun taas vibe-koodauksessa voidaan hyödyntää sujuvasti niin ääni-, teksti-, ja muita tiedon välittämisen keinoja. (Sarkar & Drosos, 2025)

## 2.2 Motivaatiotekijöitä Vibe-koodaukseen

Aiheen uutuuden vuoksi selviä rajanvetoja vibe-koodauksen suosioon ja siihen vaikuttaviin tekijöihin ei ole vielä tehty. Ohjelmoijien näkökulmasta toistuvia motivaattoreita vibe-koodaukseen voivat kuitenkin olla kognitiivisen kuorman vähentäminen, uusien teknologioiden ja koodityökalujen käytön helpottaminen ja yleinen flow-tilan ja ilon saavuttaminen ohjelmoijissa (Pimenova et al., 2025). Vibe-koodaukseen liitetään kolme keskeistä tekijää. Keskusteleva vuorovaikutus tekoälyn kanssa, luomistyö yhdessä tekoälyn kanssa sekä kehittäjän kokemus flow-tila sekä ilo. Kipukohdat voivat vaikuttaa negatiivisesti flow-tilaan sekä auttaa parhaiden käytäntöjen luomisessa. Parhaat käytännöt taas voivat parantaa flow:ta ja auttaa vähentämään vahvistamatonta luottoa tekoälyn kanssa suoritettavaan yhdessä luomiseen. Yhdessä luominen taas saattaa lisätä riskejä (Pimenova et al., 2025). Vibe-koodauskokemusta havainnollistetaan kuvassa 1.



Kuva 1. Vibe-koodauskokemus (Pimenova et al., 2025).

Myös erilainen kokeilutoiminta voi nopeutua generatiivisen tekoälyn avulla. Esimerkiksi harrasteprojektit ja nopea prototyyppiointi ovat suosittuja syitä vibe-koodaukselle (Pime-

nova et al., 2025). Vibe-koodausta voidaan esimerkiksi hyödyntää niin kutsutussa hiekkalaatikkomaisessa prototypoinnissa, jolloin oppiminen ja intuition rakentaminen helpottuu huomattavasti (Fawzy et al., 2025). Nopeampi kokeilutoiminta ja itseopiskelun hyödyntäminen voivat helpottaa suuresti, sillä vibe-koodaus voi myös toimia selkeänä oppimisen apurina. Vibe-koodausta voidaan usein hyödyntää ikään kuin tuutorin roolissa, jolloin kyselemisen ja avun pyynnön kynnyks voi olla aiempaa alhaisempaa (Fawzy et al., 2025).

Toinen selkeä motivaattori vibe-koodauksen hyödyntämiseen on aika. Ohjelmistotuotanto ja ohjelmointi alana ja ammattina ovat kamppailleet työn kuormittavuuden ja aikapaineiden kanssa jo pitkään. Tarvemäärittelyn heikkoudet ja aliarviointi, muuttuvat vaatimukset sekä pitkät kehitysjaksot aiheuttavat helposti lisääntyvää aikapainetta kehitykseen (Kuutila et al., 2020). Lisäksi organisaatiokulttuuriset tekijät, sekä työilmapiiri voivat vaikuttaa työn kuormittavuuteen. Vaatimukset sitoutuneisuudesta organisaatiolle, työn priorisointi muiden tekijöiden edelle sekä omien saavutusten näkyvyyden merkitys urakehitykseen voivat lisätä aikapainetta ohjelmistokehityksessä (Kuutila et al., 2020). Tutkittaessa motivaattoreita vibe-koodauksen hyödyntämiseen jopa 62 % vastaajista kertoi henkilökohtaiseksi päämotivaattorikseen ajan (Fawzy et al., 2025). Vibe-koodaus voi nousta esiin toimivana keinona lievittää ajan ja aikamäärien luomia paineita, sillä sen avulla voidaan nopeuttaa kehitystä huomattavasti. Vibe-koodaamalla toimivaa koodia voidaan saada parhaimmillaan aikaiseksi viikkojen sijasta jopa tunneissa (Fawzy et al., 2025).

Ohjelmoinnin opetus sekä teknologia-ala yleisesti kamppailevat vieläkin epäoikeudenmukaisuuden ja saavutettavuuden ongelmien kanssa, erityisesti aivan vasta-alkajien keskuudessa (Rea, 2022). Tähän generatiivinen tekoäly ja vibe-koodaus voi tarjota ratkaisuja. Yksi keskeinen motivaattori vibe-koodauksen hyödyntämiseen onkin juuri saavutettavuus ja kokemus voimaantumisesta (Fawzy et al., 2025). Vibe-koodaus voikin tarjota aiempaa matalamman kynnyksen ohjelmoinnin opiskelusta kiinnostuneille ja myös ylläpitää sisäistä motivaatiota. Vibe-koodaus voi myös toimia selkeänä oppimisen apurina. Vibe-koodausta voidaan usein hyödyntää ikään kuin tuutorin roolissa, jolloin kyselemisen ja avun pyynnön kynnyks on alempi. Tässä keskeisenä tekijänä on vibe-koodaukseen usein yhdistetty hiekkalaatikkomainen prototyyppi, jolloin oppiminen ja intuition rakentaminen helpottuu huomattavasti (Fawzy et al., 2025). Mielenkiintoista onkin, että niin aloittelevat kuin kokeneetkin ohjelmoijat kokivat vibe-koodauksen selkeäksi motivaattoriksi juuri oppimisen tehostamisen (Fawzy et al., 2025).

### **3 Vibe-koodauksen mahdollisuuksia**

Vibe-koodaus tuo mukanaan uuden vaihtoehdon perinteiselle ohjelmistokehitykselle ja koodaukselle. On selvää, että se tulee vaikuttamaan alaan niin ammattilaisten, opiskelijoiden kuin organisaatioidenkin välityksellä. Vibe-koodauksen ytimessä olevan tiedon jallostamisen, luomisen ja keskustelun takia sen voidaan kuvailla muuttavan ohjelmistojen kehittämistä teknisestä suorittamisesta kohti strategista organisointia (Meske et al., 2025).

#### **3.1 Mahdollisuudet yksilön ja alan ammattilaisten kannalta**

Uusi, puhtaasta teknisestä suorittamisesta poikkeava lähestymistapa avaa lukuisia mahdollisuuksia uudelleenlaiselle ajattelulle ja uudenlaisten toimintatapojen hyödyntämiselle. Sen voidaan myös nähdä tukevan aiempaa selkeämmin korkeamman tason ajattelua ja edistävän refleктоivaa sekä mukautuvaa ongelmanratkaisua (Meske et al., 2025).

Yksi vibe-koodauksen vahvuuksista liittyy toiminnan ja muutosten tekemisen nopeuteen. Generatiivinen tekoäly kykenee johtamaan haluttuja tarkoituksia luonnollisella kielellä kirjoitettujen kyselyiden takaa, jonka seurauksena prototypointi ja nopea testaaminen on aiempaa mahdollisempaa (Meske et al., 2025). Tämän seurauksena ideointi ja vaihtoehtojen lähestymiskulmien kokeilu ovat mahdollisia selkeästi aiempaa nopeammin (Meske et al., 2025). Oppimisen kannalta generatiivisen tekoälyn käyttö hyödyttää eritoten oppijoita, joilla on alhainen itseluottamuksen ja osaamisen taso (Prather et al., 2024). He myös saattavat hyödyntää generatiivista tekoälyä paljon useammin ja laajemmin kuin kokeneemmat ohjelmoijat (Prather et al., 2024). Toisaalta vibe-koodaus voi myös vähentää tarvetta syvälle tekniselle osaamiselle. Tekoälyn kyky täydentää ihmistietämystä voi mahdollistaa käyttäjien edistymisen jopa vajavaisella teknisellä tietopohjalla (Meske et al., 2025). Tämä voi edistää produktiivisuutta ilman täydellistä toteutukseen liittyvää osaamista (Meske et al., 2025).

Vibe-koodaus voi myös mahdollistaa henkilöstöressurssien vapauttamisen teknisestä toteutuksesta strategiisiin ja luoviin tehtäviin (Meske et al., 2025). Kun kognitiivista kuormaa siirretään ihmisiltä ihmisen ja generatiivisen tekoälyn välisiin yhteistyöratkaisuihin, voivat kehittäjät siirtää toimintaansa kohti artikulaatiota, kritiikkiä ja testausta (Meske et al., 2025). Toisaalta vibe-koodaus voi muokata työympäristöjä myös luomalla uusia tehtäviä ja osaamisalueita. Ohjelmoinnin pääpainon siirtyessä kohti artikulaatiota, kyselyiden kirjoittamista ja strategista ohjaamista, matalan tason osaamisen priorisointi voi vaihtua kohti dialogi- ja arviointitaitoja (Meske et al., 2025). Tämä voi avata ovia aiempaa

laajemmalle ja erilaisia taidoilla varustetuille ihmisille, jotka voivat ottaa osaa ohjelmistokehitykseen ja muokata sen kulkua (Meske et al., 2025).

Vibe-koodaaminen voi myös mahdollistaa kehityssykliden nopeutumista (Meske et al., 2025). Tyypillinen ohjelmistokehitysprojekti saattaa käydä läpi sarjan varsin lineaarisia kehitysvaiheita, aina tarvemäärittelystä testaukseen ja julkaisuun (Meske et al., 2025). Vibe-koodaus ja siihen liittyvä jatkuva keskinäinen mukautuminen (engl. *co-adaptation*) voi tuoda uusia mahdollisuuksia tehdä kehityksestä joustavampaa ja sulavampaa (Meske et al., 2025). Prototyypin nopea luominen sekä testaus ja iterointi mahdollistavat eri vaihtoehtojen kokeilun ja arvioinnin ennen lopullista toteutusta paljon entistä nopeammin (Meske et al., 2025). Vibe-koodaus voi myös vähentää aikaa, joka kuluu muuttaessa abstrakteja vaatimusmäärittelyitä suoritettavaan koodiin (Meske et al., 2025). Automatisoimalla tämä prosessi voidaan minimoida manuaalista työtä ja henkilöstön kognitiivista kuormaa, joka voi johtaa nopeutuvaan kehitykseen ja virheiden vähenemiseen (Meske et al., 2025). Tämän lisäksi kehoitteiden kirjoittamiseen, kyselyiden tulkitsemiseen ja niiden uudelleenkirjoittamiseen pohjautuva työskentely voi eliminoida tarpeen luoda täysin valmiita tarvemäärittelyitä, jolloin myös tämä työvaihe muuttuu entistä joustavammaksi (Meske et al., 2025).

Yksilön kannalta keskeisiä etuja ovat myös saavutettavuuteen ja inklusioon liittyvät seikat. Syrjäytettyihin ja vähemmistöryhmiin kuuluvat ohjelmoijat kohtaavat usein identiteettiin pohjautuvia esteitä perinteisissä ylennys ja urakehitysjärjestelmissä (Rea, 2022). Joissakin tapauksissa tämä voi näyttäytyä senioritason kehittäjien ennakkoluuloina, esimerkiksi niin, että vähemmistöön kuuluvan ohjelmistokehittäjän ajatellaan olevan automaattisesti vähemmän tekniseen tehtävään työllistytävä (Rea, 2022). Vibe-koodaus voi tarjota ratkaisuja ennakkoluuloihin ja eri ihmisryhmien kokemaan eriarvoisuuteen alalla. Vibe-koodaukseen vahvasti liittyvä siirtymä vahvan teknisen ja substanssiosaamisen avulla tehdystä ohjelmistokehityksestä kohti luonnollisella kielellä tapahtuvaa työskentelyä avaa ovia ja mahdollisuuksia myös muillekin kuin alan erikoisosaajille (Meske et al., 2025). Vibe-koodaus myös mahdollistaa ajatusten, tavoitteiden ja visioiden ilmaisemisen luonnollisella kielellä, joka voi johtaa parantuvaan inklusioon ja innovaatiomyönteiseen ilmapiiriin (Meske et al., 2025).

Myös alan ammattilaiset voivat hyötyä vibe-koodaamisen mukanaan tuomista mahdollisuuksista. Luonnollisella kielellä kommunikointi voi auttaa alan asiantuntijoita muuttamaan tietämyksensä ja osaamisensa suoraan toimivaksi ohjelmaksi ja samalla vähentämällä tarvetta formaaleihin ohjelmointitaitoihin (Meske et al., 2025). Vähenevä riippuvuus syvästä ohjelmointitaidosta voi myös auttaa ihmisiä yhä laajemmilta osaamisen aloilta osallistumaan ohjelmiston kehitykseen, samalla demokratisoiden luovaa teknolo-

gista työtä ja laajentamalla pääsyä sen pariin (Meske et al., 2025). Tekoälyn kyky täydentää ihmistietämystä voi myös mahdollistaa käyttäjien edistymisen jopa vajavaisella teknisellä tietopohjalla (Meske et al., 2025). Tämä voi edistää produktiivisuutta ja ilman täydellistä toteutukseen liittyvää osaamista (Meske et al., 2025).

### **3.2 Mahdollisuudet organisaation ja laajemman ekosysteemin kannalta**

Vaikka vibe-koodaus tuo mukanaan uusia mahdollisuuksia ja työskentelytapoja eritoten yksilön näkökulmasta, voivat myös yritykset ja muut organisaatiot hyödyntää sitä. Vibe-koodaus voi muokata toimintaa sekä organisaatio- että ekosysteemitasolla, eritoten liittyen organisaatioiden ketteryteen sekä haasteiden kohtaamiseen (Meske et al., 2025). Vibe-koodaus voi esimerkiksi voimistaa pienten tiimien toimintaa tilanteissa, jotka perinteisillä toimintatavoilla vaatisivat paljon suurempaa panostusta IT-tekniikan saralla (Meske et al., 2025).

Myös organisaation kyvykkyyksien uudelleensijoittelu helpottuu ja organisaatio voi keskittyä palkkaamaan teknisten specialistien sijasta toimiala-asiantuntijoita sekä luovan alan erikoisosaajia (Meske et al., 2025). Näin voidaan optimoida organisaatiossa olevaa kyvykkyyttä ja luoda parempia olosuhteita eri alojen väliselle yhteistyölle (Meske et al., 2025). Tähän liittyen vibe-koodaus voi myös vapauttaa it-osastoja ja kehittäjiä vaativampiin tehtäviin, ja mahdollistaa yksinkertaisen automaatiotyökalujen kehittämisen ilman IT-osastojen kuormittamista (Fawzy et al., 2025). Vibe-koodaus voi siis avata tietojärjestelmien kehittämisen ovela yhä laajemmalle osalle organisaation työntekijöistä, ja täten auttaa IT-alan ammattilaisia keskittymään vaativampiin tehtäviin.

Vibe-koodaus voi myös alentaa taloudellisia esteitä ja avata uusia mahdollisuuksia pyrkiä suorittamaan projekteja, jotka vanhoilla toimintatavoilla ovat olleet joko teknisten tai resurssiongelmien vuoksi mahdottomia toteuttaa (Meske et al., 2025). Tämä voi näkyä eritoten uusien projektien toteuttamisen ja innovaatiokyvyn paranemisena, sillä organisaatioiden näkökulmasta vibe-koodauksen tuomat edut ovat erityisen houkuttelevia kehityksen alkuvaiheissa. Vibe-koodaus toimii tehokkaimmin kehityksen alkuvaiheissa, joissa vaaditaan luovuutta, nopeaa palautetta ja joustavuutta ohjauksessa ja valvonnassa (Sapkota et al., 2025). Se toimii myös hyvin oppimista nopeuttavana työkaluna, jonka avulla organisaation työntekijät voivat nopeasti prototypoimalla kokeilla uusia teknologioita ja kysyä palautetta (Sapkota et al., 2025).

Vibe-koodauksen iteratiivinen ja prototypointiin kannustava luonne voi myös parantaa innovaationopeuden lisäksi kykyä reagoida markkinoiden muuttuvaan kysyntään (Meske et al., 2025). Kun yhdistetään vibe-koodaukseen liittyvät kokeileva ja prototypoiva kehittäminen, nopeampi kehityksen alkuvaihe sekä kehityksen iteratiivinen luonne, voidaan markkinoiden tarpeita ja vaatimuksia seurata helpommin ja saavuttaa kilpailuetua (Meske

et al., 2025). Vibe-koodaus voi myös vapauttaa IT-osastoja ja kehittäjiä vaativampiin tehtäviin, ja mahdollistaa yksinkertaisen automaatiotyökalujen kehittämisen ilman IT-osastojen osallistamista. Vibe-koodaus avaa tietojärjestelmien kehittämisen ovea siis yhä laajemmalle osalle organisaation työntekijöistä. (Fawzy et al., 2025)

## **4 Vibe-koodauksen riskit**

Vibe-koodauksen uutuuden ja standardien puuttumisen vuoksi keskustelu riskeistä on ollut usein epäselvää ja tunnepohjaista. Riskit ja niiden toteutuminen voivat myös riippua yksittäisestä kehittäjästä sekä toimintamallista. Vibe-koodauksesta voidaan kuitenkin jo nyt nimetä tiettyjä riskitekijöitä, jotka on hyvä ottaa huomioon.

### **4.1 Riskit ohjelmoijan ja koodin laadun kannalta**

Vibe-koodaus tuo kehittäjille mukanaan uuden tavan olla vuorovaikutuksessa koodin kanssa. Vibe-koodauksessa kehittäjä irrottaa itsensä koodin suorasta muokkauksesta ja hyödyntää generatiivista tekoälyä välikätenä, jonka avulla hän muokkaa koodia. Tekoälyn generoima koodi, kommentit sekä virheet vaikuttavat kaikki aktiivisesti kehittäjän aikomuksiin ja toimintaan, jotka kehittyvät iteratiivisesti. Prosessin seurauksena voidaan kuitenkin potentiaalisesti huomata koodin manuaalisesta muokkaamisesta sekä sen tuomien haasteiden kanssa kamppailusta aktivoituvan syvän ymmärryksen ja taitojen kehittymisen häviämistä. (Sarkar & Drosos, 2025)

Yksilöiden ja yksittäisten alan ammattilaisten kannalta riskit liittyvät eritoten ohjelmointitaitojen ylläpitoon ja niiden rapautumiseen. Vibe-koodauksen yhteistyömäinen ja keskusteleva luonne voi johtaa ohjelmoijien roolin muuttumiseen. Vibe-koodatessa ohjelmoijan asema voi siirtyä nopeasti toteuttajasta artikuloijaksi, jolloin keskeisten ohjelmistokehitystaidot kuten debuggaus, arkkitehtuurin suunnittelutaidot sekä algoritmien ajattelu voivat helposti kärsiä (Meske et al., 2025). Esimerkiksi tutkittaessa opiskelijoiden mielikuvia generatiivisesta tekoälystä (Rachh & Kavatagi, 2024), osoittautui eritoten tekoälyn generoiman koodin oikeellisuuden tarkastaminen haasteelliseksi (Rachh & Kavatagi, 2024).

Ohjelmointitaitojen rapautumiseen liittyviä riskitekijöitä ovat myös esimerkiksi yli-itsevarmuus, kuviteltu osaaminen sekä tietotaidon ja osaamisen välittämisen heikkeneminen (Meske et al., 2025). Generatiivisen tekoälyn kanssa käytävän keskustelevan kehotteiden kirjoittamisen ja vibe-koodaamisen avulla saadaan nopeasti aikaan toimivaa koodia ilman syvää ymmärrystä, jolloin riski kriittisten haavoittuvuuksien kuten tietoturvahkien ylen-

katsonnalle kasvaa huomattavasti (Meske et al., 2025). Eritoten aloittelevilla ohjelmoijilla generatiivisen tekoälyn käyttö saattaa johtaa väärinkäsityksiin ja metakognitiivisiin ongelmiin (Prather et al., 2024). Noviisitason ohjelmoijat voivat myös helposti perustella tekoälyn käyttöönsä tavalla, joka on ristiriidassa heidän todellisten tekojen ja osaamisensa kanssa (Prather et al., 2024). Kokeneemmat ohjelmoijat eivät kohtaa vastaavia ongelmia samoissa määrin, joten huolet vibe-koodaamisen ja generatiivisen tekoälyn vaikutuksesta oppimiseen ovat erityisen suuria uran ja oppimisen alkuvaiheessa (Prather et al., 2024).

Toinen uhkakuvana liittyy itse lähdekoodin ominaisuuksiin. Perinteisesti ohjelmistokehitysympäristöissä on usein asetettu tiukat rajat, säädökset ja standardit, joiden pohjalta kehitystä tehdään (Meske et al., 2025). Generatiivisen tekoälyn lisääminen kehitysprosessiin voi kuitenkin rapauttaa näiden tuomaa ja ylläpitämää laatua. Vibe-koodauksessa yleinen toistuvien kyselyjen tekeminen voi tuottaa ennalta-arvaamattomia korjauksia lähdekoodiin ja tehdä paikallisesta testaamisesta tehottomampaa (Meske et al., 2025). Erityisesti kehittäjien näkökulmasta on perinteisesti ajateltu eritoten ymmärrettävyyden ja huollettavuuden olevan ensiarvoisen tärkeää. Tämä voidaan ymmärtää siten, että koodin tulee olla hyvin dokumentoitua, luettavaa ja sillä tulee olla hyvä rakenne (Börstler et al., 2023). Myös läpinäkyvyys on keskeinen kysymys vibe-koodauksen vaikutuksia arvioiessa. Tekoälyn ottaessa enemmän roolia kaikilla kehityksen osa-alueilla, myös testaus ja laadunvarmistus käyvät vähemmän läpinäkyvimiksi ja luotettaviksi (Meske et al., 2025).

Laadukasta koodia ei voida myöskään helposti ja yksiselitteisesti määritellä, vaan siihen voidaan vaikuttaa lukuisilla eri tasoilla. Esimerkiksi mikrotason laadun priorisointi ei välttämättä vielä takaa laadukasta koodia systeemitasolla (Börstler et al., 2023). Generatiivisen tekoälyn kanssa keskustelemalla tuotetussa koodissa voi myös usein olla puutteita juuri rakenteen, mallien johdonmukaisuuden ja oikeanlaisen dokumentaation saralla, joka voi helposti vaikeuttaa tulevaisuudessa tehtävää muokkausta. (Meske et al., 2025)

Vibe-koodaukseen liittyy myös ongelmia esimerkiksi visuaalisten ideoiden kommunikoinnissa. Vibe-koodauksessa pääasiassa käytetty tekstin välityksellä kommunikointi voi osoittautua hankalaksi, esimerkiksi tilanteissa, joissa halutaan kommunikoida diagrammien muokkausehdotuksia (Sarkar & Drosos, 2025). Myös päätökset siitä, hyödynnetäänkö tietystä tilanteesta generatiivista tekoälyä ja vibe-koodausta, vai ihmisen suorittama muokkausta, voi luoda lisää kognitiivista kuormaa kehittäjälle (Sarkar & Drosos, 2025).

## 4.2 Riskit organisaatioiden ja vastuun kannalta

Tiedon tuottamisen jakaminen generatiivisen tekoälyn kanssa tuo mukanaan lukuisia uusia ongelmia ja riskejä eritoten vastuunjaon, tiedon jakamisen ja ongelmien ratkaisun saralla. Kun vastuu on jaettu useiden eri toimijoiden, mukaan lukien tekoälyjärjestelmien kesken, voi yhden osapuolen olla entistä haastavampaa selittää miksi tiettyihin lopputuloksiin on päädytty (Meske et al., 2025). Kun toisaalta kehittäjät hyötyvät lisääntyneestä tehokkuudesta vibe-koodausta hyödynnettäessä, ovat he myös raportoineet riskejä liittyen yliriippuvuuteen tekoälyratkaisuista. (Ge et al., 2025) Jos osa tekoälyn generoimaa koodia aiheuttaa haavoittuvuuksia, syy voi yhtä hyvin olla kehittäjän, tarkastajan tai AI-mallin palveluntarjoajan harteilla, jonka vuoksi erilaisten vastuullisuuskehysten merkitys kohoaa (Ge et al., 2025).

Niin kutsutut musta laatikko -tekoälymallit eivät mahdollista pääsyä ja tarkkailua niiden sisällä toimivaan logiikkaan, algoritmeihin, opetusdataan tai oppimiskyvykkyyksiin. Ainoa keino päätellä tai selvittää mallin toimintaa on pyrkiä korreloimaan siihen syötettyä ja siitä ulos saatua dataa. Tämän takia musta laatikko -tekoälymalleja voidaan pitää vähemmän luotettavina. (Shridhar, 2024) Organisaatioiden kannalta black box -koodikannat sekä vaikeudet noudattaa eettisiä ja tietoturvaan liittyviä säädöksiä tuovat mukanaan suuria vaikeuksia tarkastettavuuden ja todennettavuuden saralla, etenkin tarkasti säädellyillä aloilla kuten terveydenhuollossa (Meske et al., 2025). AI-järjestelmät myös sitouttavat organisaatiota käyttämään tiettyjä palveluita ja teknologisia ratkaisuja, jotka voivat lisätä organisaation taloudellista ja infrastruktuurista riippuvuutta ja heikentää sen autonomiaa (Meske et al., 2025).

Vibe-koodaukseen ja tekoälyyn liittyvä iteratiivinen ja keskusteleva vuorovaikutus tekee selkeän rajan vetämisen ihmisen ja tekoälyn tuotosten välille erittäin vaikeaksi (Meske et al., 2025). Tämän lisäksi kehittäjät voivat helposti eriytyä järjestelmän sisäisestä logiikasta, jolloin sen ymmärrys, virheiden korjaus muodostuu paljon vaikeammaksi (Meske et al., 2025). Vibe-koodaamisen tyypillinen työnkulku voi myös helposti hämärtää alkuperäisten vaatimusten ja toteutettujen ratkaisujen välistä yhteyttä, jolloin koodin ylläpito tulevaisuudessa vaikeutuu, kun uudet ylläpitäjät eivät voi selvittää koodin tuottajan alkuperäistä tarkoituspäätä ja taustoja keskeisten päätösten takana. (Meske et al., 2025)

Eettiset kysymykset sekä tietoturva ovat myös selviä riskitekijöitä. Generatiivisen tekoälyn rajoitettu kyky selittää päätöksiään ja sen taustalla toimiva todennäköisyyksiin pohjautuva päätöksenteko kohottaa riskiä tietoturvarikkomuksiin ja huomaamatta jääneisiin eettisiin rikkomuksiin (Meske et al., 2025). Tekoälyagentit saattavat myös tietyissä tilan-

teissa rajoittaa toimintaansa liialti pohjaten organisaatiotason ohjeisiin ja rajoitteisiin, jolloin työskentely sekä vibe-koodamiselle ominainen helppo ja keskusteleva työtapo voi olla vaikeaa (Pimenova et al., 2025).

Vibe-koodaukseen keskeisesti liitetty riski kehitystyön suorittajan virheellisestä itsevarmuudesta voi myös tuottaa ongelmia organisaatioille. Vibe-koodausmenetelmällä tuotetut ohjelmat voivat ulkoisesti vaikuttaa täysin päteviltä, mutta piilottaa vakavia ongelmia. Jos kehitystiimit tottuvat julkaisemaan vibe-koodaamalla tuotettuja hauraita tai viallisia ohjelmia, on riskinä se että laadunvarmistuksen rima alenee koko organisaation tasolla. Ajan kuluessa tämä voi johtaa tilanteeseen, jossa testaamaton koodi on organisaatiokulttuurin puolesta täysin hyväksyttävää. Tämän seuraksena riski kalliisiin epäonnistumisiin, katkoksiin ja tietoturvarikkomuksiin nousee. (Curtis et al., 2022)

Vibe-koodausta hyödyntävät organisaatiot voivat myös törmätä linjauksiin liittyviin ongelmiin. Vibe-koodauksessa hyödynnettävät agentit ja järjestelmät pohjaavat päätöksensä todennäköisyyksiin, poiketen täten determinististä kääntäjästä (Ge et al., 2025). Erityisesti tavoiteorientoituneet mallit optimoivat toimintaansa usein tekstillisen menestyksen kannalta sen sijaan, että ne priorisoisivat normatiivisia tuloksia (Ge et al., 2025). Tämän takia vibe-koodausagentit vaativatkin jatkuvaa valvontaa, ihmisen tekemää tarkastusta ja ymmärrettäviä palkitsemismalleja, jotta voidaan varmistaa että kehittäjän alkuperäinen tarkoitus välittyy myös agentin suorittamiseen (Ge et al., 2025). On myös mielenkiintoista pohtia kuinka laajemmat tekoälyn käyttöön ja sen kasvuun liittyvät riskit peilautuvat vibe-koodaukseen. Tekoälyyn liittyvät lait ja säädökset eivät ole pysyneet tekoälyn hyödyntämisen huiman kasvuvauhdin mukana, ja tekoälyn maailmanlaajuisen luonteen vuoksi säädöksiä tekeminen ja käyttöönotto on mutkikasta. (Curtis et al., 2022)

## 5 Yhteenveto

Tutkielmassa käsiteltiin vibe-koodausta ja sen aikaansaamia vaikutuksia, hyötyjä ja haittoja erityisesti yksilöiden ja organisaatioiden kannalta. Vibe-koodauksen käsite on kehittynyt vasta vuoden 2025 aikana, joten mielikuvat sen ominaisuuksista sekä käyttökohteista vaihtelevat, eikä sillä vielä ole täysin vakiintuneita määritelmiä ja käyttökohteita. On kuitenkin selvää että vibe-koodaukselle ominaisia asioita ovat keskusteleva vuorovaikutus (Meske et al., 2025), sekä tekoälyn vastuun kasvattamista, joissain tapauksissa jopa ikään kuin ohjelmistokehityksen päävastuuseen (Geng et al., 2025). Vibe-koodauksesta voidaan myös tehdä rajanveto muihin tekoälyä hyödyntäviin toimintamalleihin (Sarkar & Drosos, 2025). On myös mielenkiintoista vertailla vibe-koodausta

ja siihen liittyvää työskentelyä perinteisempään tekoälyavusteiseen ohjelmistokehitykseen sekä esimerkiksi agenttimalleja hyödyntäviin työskentelymalleihin. Rajanveto voi usein olla vaikeaa, mutta keskeistä lienee juuri erottelu vibe-koodaukseen keskustelevaan yhteistyöhön, sekä perinteisempiin tekoälytyöskentelyn malleihin kuten vaikkapa koodieditoreiden automaattisten korjausehdotusten ja sisäänrakennettujen apureiden kesken.

Vibe-koodaus tuo mukanaan valtavia mahdollisuuksia ja toisaalta myös pitkän listan realistisia uhkakuvia. Tutkielmassa ilmenneet keskeisimmät riskit ja mahdollisuudet on lisätty taulukoissa 1 ja 2. Löydöksiä ei ole järjestelty systemaattisesti, vaan järjestys on sattumavarainen.

Taulukko 1. Keskeisimmät vibe-koodaukseen liitetyt riskit

	Organisaatiolle	Yksilölle
1.	Eettiset ja vastuunjaolliset ongelmat	Ohjelmointitaitojen rapautuminen
2.	Black Box -tekoälymallien valvontaan ja säätelyyn liittyvät ongelmat	Tavoitteiden kommunikoinnin ja täten toteutuksen epäonnistumisen riski
3.	Koodin laadun ja tietoturvan heikentyminen	Koodin laadun heikentyminen
4.	Organisaation osaamistason heikentyminen	Metakognitiiviset ongelmat

Taulukko 2. Keskeisimmät vibe-koodaukseen liitetyt mahdollisuudet

	Organisaatiolle	Yksilölle
1.	Innovaationopeuden parantuminen	Prototypoinnin ja testaamisen nopeutuminen
2.	Kehityksen alkuvaiheen nopeutuminen ja kustannuksien aleneminen	Oppimisen helpottaminen ja nopeutuminen

3.	Kyvykkyyksien uudelleen sijoittamisen helpottuminen	Manuaalisen työn ja kognitiivisen kuorman väheneminen
4.	Reaktionopeuden parantuminen	Saavutettavuuden ja inklusion parantuminen

Tutkielmassa ilmenneitä vibe-koodauksen hyötyjä ovat erityisesti kognitiiviseen vapautumiseen liittyvät edut sekä prototypoinnin ja uusien teknologioiden opetteluun helpottuminen. Generatiivisen tekoälyn kanssa keskustelemalla ja vibe-koodaamalla myös kynnys kysyä ja kyseenalaistaa voi olla alempi, kuin ihmiskehittäjien kanssa. (Fawzy et al., 2025). Parhaimmassa tapauksessa tämä voi olla erittäin myönteinen tekijä eritoten alaa opiskelevien ja tulevien ammattilaisten oppimisen kannalta. Vibe-koodaamisen riskit taas liittyvät usein ohjelmointitaitojen rapautumisen ja tuotetun koodin ominaisuuksiin kuten tietoturvariskeihin (Meske et al., 2025). Ohjelmointitaitojen mahdollinen rapautuminen sekä tekoälyn kirjoittaman koodin riskit ovat myös erinomainen ja mielenkiintoinen jatkotutkimuksen kohde, joka tulee todennäköisimmin näkymään myös tulevissa tutkimuksissa.

Organisaatioiden näkökulmasta vibe-koodaus voi tuoda hyötyjä muun muassa kehityksen nopeutumiseen, resurssien uudelleenjakamista helpottavana tekijänä (Meske et al., 2025). Lisäksi vibe-koodauksen mahdollistama nopeampi kehitystahti voi osaltaan parantaa yrityksen prototypointikykyä, sekä tätä kautta auttaa organisaatiota vastaamaan nopeammin ja tehokkaammin markkinamuutosten asettamiin haasteisiin (Sapkota et al., 2025). Riskejä organisaatioiden kannalta ovat erityisesti vastuukysymykset sekä eettiset seikat. (Meske et al., 2025). On mielenkiintoista nähdä, miten näihin haasteisiin vastataan tulevaisuudessa, jossa tekoäly ja sen myötä myös mahdollisesti vibe-koodaus kasvaa koko ajan isompaan rooliin organisaatioissa.

Aihe on varsin uusi, eikä vertaisarvioitua tutkimusta siitä ole juurikaan vielä julkaistu. Tämän vuoksi kirjallisuuskatsauksessa hyödynnettyjen lähteiden tutkimusmenetelmät olivat usein haastattelu- ja kirjallisuuskatsauspainotteisia. Tämän vuoksi riski tulosten vääristymiselle ja virheellisille johtopäätöksille on olemassa. Aihe tulee varmasti keräämään lisää mielenkiintoa ja uusia tutkimuksia tulevina vuosina, kun ilmiö kasvaa.

## 6 Lähdeluettelo

Börstler, J., Bennin, K., Hooshangi, S., Jeurung, J., Keuning, H., Kleiner, C., MacKellar, B., Duran, R., Störrle, H., Toll, D., van Assema, J. (2023) ‘Developers talking about code quality’, *Empirical Software Engineering*, 28(6). doi:10.1007/s10664-023-10381-0.

Curtis, C., Gillespie, N. and Lockey, S. (2022) ‘AI-deploying organizations are key to addressing “perfect storm” of AI risks’, *AI and Ethics*, 3(1), pp. 145–153. doi:10.1007/s43681-022-00163-7.

Deng, Y., Liao, L., Lei, W., Yang, G., Lam, W., Chua, T. (2025) ‘Proactive conversational AI: A comprehensive survey of ad-vancements and opportunities’, *ACM Transactions on Information Systems*, 43(3), pp. 1–45. doi:10.1145/3715097.

Fawzy, A., Tahir, A., & Blincoe, K. (2025). ‘Vibe coding in practice: Motivations, challenges, and a future outlook — A grey literature review.’ arXiv. <https://doi.org/10.48550/arXiv.2510.00328>

Kuutila, M., Mäntylä, M., Farooq, U., & Claes, M. (2020). ‘Time pressure in software engineering: A systematic review. *Information and Software Technology*, 121, 106257. <https://doi.org/10.1016/j.infsof.2020.106257>

Meske, C., Hermanns, T., Weiden, E., Loser, K.-U., & Berger, T. (2025). ‘Vibe coding as a reconfiguration of intent mediation in software development: Definition, implications, and research agenda’ (arXiv:2507.21928). arXiv. <https://doi.org/10.48550/arXiv.2507.21928>

Pimenova, V., Fakhoury, S., Bird, C., Storey, M., Endres, M. (2025) ‘Good vibrations? A qualitative study of co-creation, communication, Flow, and trust in vibe coding’, arXiv.org. Available at: <https://arxiv.org/abs/2509.12491> (Accessed: 06 October 2025).

Prather, J., Reeves, B., Leinonen, J., MacNeil, S., Randrianasolo, A., Becker, B., Kimmel, B., Wright, J., Briggs, B. (2024) ‘The widening gap: The benefits and harms of Generative AI for novice programmers’, *Proceedings of the 2024 ACM Conference on International Computing Education Research - Volume 1*, pp. 469–486. doi:10.1145/3632620.3671116.

Rachh, R. and Kavatagi, S. (2024) ‘Study on students’ perceptions of Generative AI in Learning Programming Courses’, *2024 3rd International Conference for Advancement in Technology (ICONAT)*, pp. 1–5. doi:10.1109/iconat61936.2024.10774681.

Rea, A. (2022) ‘Coding Equity: Social Justice and Computer Programming Literacy Education’, in *IEEE Transactions on Professional Communication*, vol. 65, no. 1, pp. 87-103, doi: 10.1109/TPC.2022.3143965., *Information Society and the Workplace: Spaces, Boundaries and Agency*. (ss. 29-46). Routledge

Sapkota, R., Roumeliotis, K. I., & Karkee, M. (2025). ‘Vibe coding vs. agentic coding: Fundamentals and practical implications of agentic AI.’ arXiv. <https://doi.org/10.48550/arXiv.2505.19443>

Sarkar, A., & Drosos, I. (2025). ‘Vibe coding: Programming through conversation with artificial intelligence.’ In *Proceedings of the 36th Annual Conference of the Psychology of Programming Interest Group (PPIG 2025)*. arXiv.

<https://doi.org/10.48550/arXiv.2506.23253>

Shridhar, 'Exploring the Dark Side of AI: The Challenges of Explain Ability, Algorithmic Bias, and Ensuring Fairness', *2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG)*, Indore, India, 2024, pp. 1-6, doi: 10.1109/ICTBIG64922.2024.10911065.

Zhu, J. and Van Brummelen, J. (2021) 'Teaching students about conversational AI using Convo, a conversational programming agent', *2021 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 1-5. doi:10.1109/vl/hcc51201.2021.9576290.