

Jemina Kotkajuuri

Pariohjelmoinnin kommunikaation tukeminen opetuksessa

Tietotekniikan kandidaatintutkielma

29. huhtikuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Jemina Kotkajuuri

Yhteystiedot: jemina.l.kotkajuuri@student.jyu.fi

Ohjaaja: Antti-Jussi Lakanen

Työn nimi: Pariohjelmoinnin kommunikaation tukeminen opetuksessa

Title in English: Supporting pair programming communication in education

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 24+0

Tiivistelmä: Pariohjelmointi on tietotekniikan opetuksessa käytetty menetelmä, jonka käytöstä on tutkitusti hyötyä opiskelijoille. Hyvä kommunikaatio on tärkeä tekijä pariohjelmointikokemuksessa. Tämä kandidaatintutkielma keskittyy pariohjelmoinnin kommunikaation tukemiseen opetuksessa. Tavoitteena on löytää aiemmasta tutkimuksesta keinoja, joilla pariohjelmointiparien kommunikaatiota voisi tukea.

Avainsanat: pariohjelmointi, kommunikaatio, tietotekniikka, opetus

Abstract: Pair programming is a method used in computer science education, and its use has been proven to be beneficial for students. Good communication is an important factor in the pair programming experience. This Bachelor's thesis focuses on supporting pair programming communication in education. The aim is to find ways to support the communication of pair programming pairs from previous research.

Keywords: pair programming, communication, computer science, education

Sisällys

1	JOHDANTO	1
2	PARIOHJELMOINTI	2
	2.1 Määritelmä.....	2
	2.2 Hyöty opetuksessa.....	2
3	KOMMUNIKAATION MERKITYS PARIOHJELMOINNISSA	4
	3.1 Pariohjelmoinnin kommunikaatiokokemukset.....	4
	3.2 Hyvän kommunikaation piirteet	5
4	KOMMUNIKAATION TUKEMINEN PARIOHJELMOINNISSA	7
	4.1 Istumisjärjestely ja yhteinen tavoite	7
	4.2 Kommunikaatio-ohjeet	8
	4.3 Parien muodostaminen	10
	4.3.1 Taitotaso.....	10
	4.3.2 Sukupuoli	11
	4.3.3 Persoonallisuus	12
	4.3.4 Johtopäätökset parien muodostamisesta	14
5	YHTEENVETO.....	16
	LÄHTEET	17

1 Johdanto

Ohjelmointikursseihin, etenkin ensimmäisiin johdantokursseihin, on tutkitusti liittynyt korkea hylättyjen suoritusten osuus (Watson ja Li 2014). Tämä on yksi monista syistä, miksi tietotekniikan opettamisen vaihtoehtoihin on tehty tutkimusta. Eräs ohjelmoinnin opettamiseen liittyvä menetelmä on pariohjelmointi: menetelmä, jossa kaksi opiskelijaa työskentelee yhdessä yhden tehtävän parissa (Williams ja Upchurch 2001). Pariohjelmoinnin käytöstä etenkin korkeakoulujen ensimmäisillä ohjelmointikursseilla on tehty huomattavasti tutkimusta, ja saadut tulokset viittaavat siihen, että pariohjelmoinnin käytöllä on positiivisia vaikutuksia oppimiseen ja opetukseen.

Huomattavasta pariohjelmoinnin opetuskäyttöä tukevan tutkimuksen määrästä huolimatta pariohjelmointi ei kuitenkaan ole menetelmä, joka on systemaattisesti käytössä ohjelmoinnin opetuksessa. Tätä voi selittää Williamsin ja Kesslerin (2001) huomio siitä, että pariohjelmointi vie opiskelijat heidän mukavuusalueensa ulkopuolelle, sillä tietotekniikan opiskelijat ovat tyypillisesti tottuneet yksintyöskentelyyn. Williams ja Kessler (2001) toteavat, että pariohjelmoinnilla on kuitenkin mahdollisuus muuttaa ohjelmoinnin opetusta positiivisesti, mutta se vaatii sitä, että opiskelijoille osoitetaan vuorovaikutuksen edut ohjelmoinnissa. Tästä huolimatta suurin osa pariohjelmoinnin tutkimuksesta keskittyy pariohjelmoinnin käytöllä saavutettuihin tuloksiin, ja pariohjelmoinnin kommunikaation ja sen tukemisen tutkimus on jäänyt vähäiseksi.

Tässä tutkielmassa tehdään kirjallisuuskartoitus pariohjelmoinnin kommunikaatioon liittyvään tutkimukseen ja tuodaan esiin pariohjelmoinnin kommunikaatioon liittyviä tutkimustuloksia. Tutkielman päätavoite on löytää keinoja, joilla kommunikaatiota voidaan tukea opetuksessa.

Luvussa 2 käsitellään pariohjelmoinnin määritelmää ja sen opetuskäyttöön liittyvää tutkimusta. Luvussa 3 perehdytään pariohjelmoinnin kommunikaatiokokemuksiin ja hyvän kommunikaation piirteisiin. Luvussa 4 käydään läpi tutkimuskartoituksessa esiin nousseet keinot, joilla voidaan tukea kommunikaatiota pariohjelmoinnissa. Luvussa 5 tehdään tiivistelmä aiemmissa luvuissa käsitellyistä asioista ja esitetään jatkotutkimusehdotukset.

2 Pariohjelmointi

Tässä luvussa määritellään pariohjelmoinnin käsite ja kuvataan lyhyesti pariohjelmoinnin historia. Tässä luvussa käydään myös läpi pariohjelmoinnin opetuskäyttöön liittyviä hyötyjä käsittelevät tutkimustulokset.

2.1 Määritelmä

Pariohjelmoinnilla viitataan tässä tutkielmassa ohjelmointityyliin, jossa kaksi ohjelmoijaa työskentelee yhdessä yhden tietokoneen ääressä yhteisen tehtävän parissa (Williams ja Upchurch 2001; Williams ym. 2002). Pariohjelmoinnissa kummallakin parin jäsenellä on tyypillisesti tietty rooli: toinen henkilö toimii kuskina (engl. *driver*) ja toinen navigaattorina (engl. *navigator*) (Williams ym. 2002; Umapathy ja Ritzhaupt 2017). Kuskin tehtävä on käyttää tietokonetta ja kirjoittaa koodia. Navigaattorin tehtävä on seurata ja kommentoida kuskin toimintaa. Parin jäsenten tulisi kommunikoida keskenään aktiivisesti ja vaihtaa ajoittain rooleja (Williams ym. 2002; Umapathy ja Ritzhaupt 2017). Pariohjelmointia on harjoitettu myös hieman poikkeavilla tyyleillä, kuten pariohjelmoimalla ilman roolien ajoittaista vaihtamista (Rodríguez, Price ja Boyer 2017), kahdella eri tietokoneella (Celepkolu ja Boyer 2018) ja täysin etänä (Kuttal, Gerstner ja Bejarano 2019), mutta kartoituksen perusteella kappaleen alussa esitetty pariohjelmoinnin kuvaus on tyypillinen tapa toimia.

Pariohjelmointi on menetelmä, jota on käytetty pitkään tietotekniikan aloilla. Ensimmäiset työpaikoilla tehdyt pariohjelmointikokeilut voidaan jäljittää vuoteen 1978 (Lima Salge ja Berente 2016). Pariohjelmointi tuli laajemmin tunnetuksi 2000-luvun alussa, kun Kent Beck määritteli sen yhdeksi ketterän kehityksen Extreme Programming -metodologian pääkäytännöiksi (Lima Salge ja Berente 2016; Umapathy ja Ritzhaupt 2017).

2.2 Hyöty opetuksessa

Pariohjelmoinnin käytöstä opetuksessa on saatu positiivisia tuloksia. Pariohjelmoinnin opetuskäytöstä tehty meta-analyysi viittaa siihen, että pariohjelmoinnin käyttö kurssilla voi vai-

kuttaa positiivisesti opiskelijoiden tehtävien arvosanoihin, koetuloksiin ja kurssin läpäisyyn (Umapathy ja Ritzhaupt 2017). Tässä Umapathyn ja Ritzhauptin (2017) toteuttamassa meta-analyysissä käsiteltiin 18 tutkimusta. Siihen otettiin mukaan tutkimukset, joissa verrattiin pariohjelmointia ja yksin tehtyä ohjelmointia määrällistä analyysia hyödyntäen ja joissa keskityttiin pääasiassa opintomenestykseen ja opiskelijoiden asenteisiin. Näyttöä löytyy myös sille, että opiskelijat voivat pariohjelmointia käytettäessään tuottaa laadukkaampaa sekä vähemmän virheitä sisältävää koodia (Williams ja Upchurch 2001; McDowell ym. 2002). Pariohjelmoinnin käyttö ohjelmointikursseilla vaikuttaa olevan positiivisesti yhteydessä opiskelijoiden tyytyväisyyteen kursseilla (Williams ja Upchurch 2001; Williams ym. 2002) ja kannustavan opiskelijoita jatkamaan ohjelmointiopintojen parissa jatkossa (Williams ym. 2003; McDowell ym. 2006). Pariohjelmointi voi hyödyttää etenkin naisia tukemalla heidän ohjelmointiin liittyvää itsevarmuutta, muuttamalla heidän käsityksiään teknologia-alasta ja täten kannustamalla heitä hakeutumaan työskentelemään tietotekniikan pariin (Werner, Hanks ja McDowell 2004).

Myös opettajat voivat hyötyä pariohjelmoinnin käytöstä. Pariohjelmoitessa parin jäsenet voivat ongelmia kohdatessaan tukeutua toistensa tietoon, mikä vähentää opettajilta kysyttävien kysymysten määrää (Williams ja Upchurch 2001). Pariohjelmoinnissa pari palauttaa tyypillisesti yhden yhteisen vastauksen, mikä vähentää tarkastettavien tehtävien määrää. Opettajille jää siis enemmän aikaa keskittyä opetuksen muihin osiin. Pariohjelmoinnin käyttö näyttäisi vähentävän myös huijaustapausten määrää, sillä ryhmäpaine pakottaa opiskelijoita panostamaan työskentelyyn (Williams ja Upchurch 2001).

Pariohjelmoinnin opetuskäyttöön liittyvä laaja-alainen tutkimus osoittaa, että pariohjelmointi on toimiva opetusmenetelmä, ja merkittävässä määrässä tutkimuksia se on toiminut paremmin kuin perinteinen yksin tehty ohjelmointi. Pariohjelmoinnin hyödyn maksimoimiseksi tiettyihin asioihin kannattaa kuitenkin kiinnittää huomiota. Seuraavissa luvuissa siirrymme käsittelemään kommunikaatiota, joka on olennainen tekijä pariohjelmoitkokemuksen laadun kannalta.

3 Kommunikaation merkitys pariohjelmoinnissa

Tässä luvussa kartoitetaan pariohjelmointikokemuksiin liittyviä tutkimuksia, joissa kommunikaatiota on käsitelty. Lisäksi tässä luvussa kuvataan hyvän kommunikaation piirteitä pariohjelmoinnissa.

3.1 Pariohjelmoinnin kommunikaatiokokemukset

Kommunikaatio on erittäin oleellinen osa pariohjelmointia, ja se voi vaikuttaa siihen, kuinka hyvin pariohjelmointi toimii. Tämän voi nähdä sekä työpaikoilla että opetuksessa. Begelin ja Nagappanin (2008) tekemästä ammatissaan pariohjelmoiville suunnatusta kyselytutkimuksesta kävi ilmi, että hyviä kommunikaatiotaitoja pidettiin pariohjelmointiparin kolmanneksi tärkeimpänä ominaisuutena. Tätä tärkeämpiä olivat vain täydentävät taidot ja joustavuus. Täydentävillä taidoilla tarkoitetaan sitä, että tutkittavat toivoivat pariensa olevan taitavia asioissa, joita tutkittavat eivät itse tunteneet yhtä hyvin. Parissa arvostettuja kommunikaatioon liittyviä asioita olivat muun muassa kyky olla hyvä kuuntelija, verbaalisuus ja kysymyksien kysyminen sekä mielipiteiden antaminen (Begel ja Nagappan 2008).

Myös opiskelijat kokevat kommunikaation tärkeäksi pariohjelmointiprosessissa. McChesneyn (2016) tutkimuksessa, jossa opiskelijoita pyydettiin jakamaan heidän näkemyksiään pariohjelmoinnin parhaimmista ja huonoimmista puolista, yksi keskeisiä pariohjelmoinnin parhaimpien puolien teemoja oli keskustelun pakollisuus ratkaisuun pääsemiseksi. Opiskelijat toivovat myös pariltaan innostusta, ja he arvostavat parin tarjoamia resursseja ongelmanratkaisussa (Simon ja Hanks 2008). Hyvän kommunikaation merkitys pariohjelmoinnissa tiivistyy hyvin erään pariohjelmointia käyttäneen opiskelijan kommentissa:

Käytännössä me vain puhuimme. Me tarkastelimme ohjelmaa - -. Sitten yritimme selvittää, mikä alue oli se osa, joka aiheutti meille eniten ongelmia. Sitten me vain kävimme sen puhumalla läpi. Käytännössä me annoimme toisillemme ideoita, kuten mitä halusimme tehdä ja minkä ajattelimme olevan ongelma, ja sen jälkeen yritimme keksiä keinoja ratkaista tuo kyseinen ongelma. (Simon ja Hanks 2008, suomennos minun)

Pariohjelmoinnin kommunikaatio ei ole kuitenkaan poikkeuksetta hyvää, ja jos kommunikaatio ei toimi, voi se vaikuttaa pahimmillaan hyvin negatiivisesti pariohjelmointikokemukseen. Kommunikaatio-ongelmat olivat esillä ammattilaisten haastattelussa, jossa pariohjelmoinnin huonoiksi puoliksi kerrottiin muun muassa huono kommunikaatio, erimielisyydet ja persoonien erot. (Begel ja Nagappan 2008).

Ammattilaisten lisäksi myös opiskelijoiden huonot pariohjelmointikokemukset liittyvät usein kommunikaatioon. Riitojen sekä mielipide-erojen on raportoitu olevan yksi pariohjelmoinnin huonoimpia puolia opiskelijoiden näkökulmasta (Simon ja Hanks 2008; McChesney 2016). McChesneyn (2016) tutkimuksessa, jossa kysyttiin opiskelijoiden mielipidettä pariohjelmoinnin huonoimmista puolista, 25 prosenttia vastauksista koski mielipide-eroja ja riitoja. Riitatilanteiden lisäksi toinen tyypillinen esimerkki tapauksista, joissa opiskelijat ovat kokeneet kommunikointiongelmia, ovat tilanteet, joissa toinen parin jäsenistä dominoi työskentelyä selittämättä auki ajatuksiaan, jolloin toinen parin jäsen jää seuraamaan työskentelyä sivusta ilman aktiivista osaa tehtävän ratkaisuun (Simon ja Hanks 2008). Eräs opiskelija kuvaili tämän tyyppistä tilannetta Simonin ja Hanksin (2008) tutkimuksessa:

Hän otti työn haltuunsa. Hän ei ollut kiinnostunut minun palautteestani. Se ei ollut hyvä kokemus, koska en kokenut oppivani. Minä katsoin hänen ohjelmointia. (Simon ja Hanks 2008, suomennos minun)

3.2 Hyvän kommunikaation piirteet

Rodríguez, Price ja Boyer (2017) perehtyivät tutkimuksessaan pariohjelmointiprosessiin ja siihen, mitä piirteitä liittyy tehokkaaseen yhteistyöhön. He käsitelivät myös hyvän kommunikaation piirteitä pariohjelmoinnissa tutkimalla sitä, miten eri dialogiliikkeet korreloivat ohjelmointitehtävässä onnistumisen kanssa. Tutkimuksessa dialogiliikkeillä tarkoitetaan keskusteluissa käytettäviä kommunikointitapoja. Käsiteltäviä dialogiliikkeitä olivat muun muassa palautteen antaminen ja kysymysten kysyminen. Rodríguezin, Pricen ja Boyerin (2017) tutkimuksen tuloksista kävi ilmi, että dialogiliikkeistä etenkin palaute (engl. *feedback*), metakommentit (engl. *meta-comments*) ja selvitykset (engl. *statements*) olivat yhteydessä parempiin tuloksiin tehtävästä. Tässä yhteydessä metakommenteilla tarkoitetaan reflektiota, kuten

ääneen ajattelua. Selvityksillä tarkoitetaan itse tehtävään liittyvän informaation jakamista tai selityksen antamista. Palautteen kohdalla sekä positiivinen että negatiivinen palaute oli yhteydessä parempiin tuloksiin. Tutkimuksen löydöksistä kävi myös ilmi, että etenkin kuskin roolissa toimivan henkilön dialogiliikkeet olivat yhteydessä pariohjelmointiprosessin tehokkuuteen ja että kuskin dialogiliikkeistä etenkin epävarmuuden ilmaiseminen ja selvitykset olivat yhteydessä parempiin tuloksiin.

Saamiensa tulosten pohjalta Rodríguez, Price ja Boyer (2017) suosittelevat, että pariohjelmointia hyödyntävien opettajien pitäisi pyrkiä kannustamaan opiskelijoita interaktiiviseen ja oleelliseen kommunikointiin. Parin välisessä dialogissa tärkeää on etenkin se, että kuskin roolissa toimivat ajattelevat ääneen ja että navigaattorin roolissa toimivat antavat aktiivisesti palautetta. Rodríguez, Price ja Boyer (2017) myös toteavat, että tilanteissa, joissa parin jäsenet ovat hiljaisia tai joissa parin jäsenten keskustelu koostuu lähinnä epävarmuuden ilmaisemisesta, kannattaa opettajan puuttua työskentelyyn ja tarjota apua. Tämä on kuitenkin mahdollista vain opetustilanteissa, joissa opettaja pystyy seuraamaan läheltä opiskelijoiden työskentelyä.

Tässä luvussa on käyty läpi tutkimuksia, jotka kuvaavat kommunikaation merkitystä pariohjelmoinnissa. Käsitellyistä tutkijoista Rodríguez, Price ja Boyer (2017) suosittelevat tutkimuksessaan kommunikaation kannustamista, mutta he eivät määrittele keinoja sen tekemiseksi. Tämän kandidaatintutkielman seuraavassa luvussa pyritään täydentämään tämä aukko käymällä läpi konkreettisia keinoja, joita hyödyntämällä pariohjelmointia käyttävät opettajat voivat mahdollisesti tukea parien kommunikaatiota.

4 Kommunikaation tukeminen pariohjelmoinnissa

Tässä luvussa käsitellään tutkimuskartoituksessa esiin nousseita keinoja, joilla voi tukea parien kommunikaatiota pariohjelmoinnissa. Aluksi käsitellään pariohjelmointitilanteen istumisjärjestelyyn ja yhteiseen tavoitteeseen liittyviä suosituksia. Tämän jälkeen käsitellään kommunikaation tukemista työelämästä peräisin olevien kommunikaatio-ohjeiden avulla. Lopuksi käsitellään kommunikaation tukemista parien muodostamiseen liittyvän tutkimuksen pohjalta, jolloin käsittelyssä on parien muodostaminen taitotason, sukupuolen ja persoonallisuuden perusteella.

4.1 Istumisjärjestely ja yhteinen tavoite

Pariohjelmoinnin tutkimuksesta löytyy tuloksia, jotka viittaavat siihen, että pariohjelmoinnin istumisjärjestelyllä ja parin jäsenten yhteisellä tavoitteella voi tukea parien kommunikaatiota. Klassinen pariohjelmoinnin asetelma on sellainen, jossa kaksi opiskelijaa työskentelee yhden tietokoneen äärellä saman tehtävän parissa. Celepkolu ja Boyer (2018) tutkivat pariohjelmoinnin asetelmaa, jossa kaksi opiskelijaa pariohjelmoivat kahdella eri tietokoneella ja jossa kumpikin parin jäsenistä palautti lopuksi erikseen oman vastauksensa. Tutkimuksen tuloksista kävi ilmi, että kumpikaan asetelma ei ollut merkittävästi toista parempi oppimistulosten kannalta, mutta klassisessa asetelmassa opiskelijat raportoivat suurempaa kommunikoinnin määrää parin kanssa ja korkeampaa tyytyväisyyttä.

Williams ym. (2008), jotka seitsemän vuoden pariohjelmointiopetuksen kokemuksen pohjalta muodostivat ohjeet pariohjelmoinnin käyttämiseen opetustilanteissa, kiinnittävät myös huomiota istumisjärjestelyyn ja yhteiseen tavoitteeseen. Ohjeiden mukaan kummallakin parin jäsenistä pitäisi olla mahdollisimman helppo pääsy paikaltaan teknisten laitteiden, kuten näytön ja näppäimistön, luokse. Heidän mukaansa parin jäsenten pitäisi pystyä istumaan mukavasti vierekkäin niin, että istumajärjestely tukisi parin välistä kommunikaatiota. Williams ym. (2008) kertovat käyttäneensä pariohjelmointiin tietokoneita, joissa on kaksi näyttöä, kaksi näppäimistöä ja kaksi hiirtä. Yhteisen tavoitteen osalta Williams ym. (2008) kertovat hyödyntäneensä kurssillaan käytäntöä, jossa parin jäsenillä ei ollut yhteistä tavoitetta,

vaan he tekivät kumpikin omaa tehtäväänsä. Parin jäsenten piti kuitenkin hyödyntää pariohjelmointia ja avustaa toisiaan tehtävien tekemisessä. Williams ym. (2008) uskovat, että yhteisellä tavoitteella olisi voitu vähentää kommunikaatioon ja motivaatioon liittyviä ongelmia, minkä seurauksena heidän ohjeissaan neuvotaan antamaan parin jäsenille yhteinen tehtävä suoritettavaksi.

Myös Muller ja Tichy (2001), jotka tekivät pariohjelmointikokeilun yliopistossa, kertovat Williamsin ym. (2008) lailla suurimman osan tutkimuksensa pareista käyttäneen kahta näyttöä pariohjelmoinnin aikana. Heidän mukaansa tyypillisesti toisella näytöllä pidettiin esillä ohjelmointikielen dokumentaatiota ja toisella itse tehtävää. 75 prosenttia kokeessa mukana olleista opiskelijoista kertoi kokevansa, että yksi näyttö olisi ollut liian vähän.

Näiden huomioiden pohjalta voimme todeta, että pariohjelmoinnin asetelmalla on vaikutusta kommunikaatioon. Vaikuttaa siltä, että parien istumisjärjestelyyn kannattaa kiinnittää huomiota ja että lähtökohtaisesti parien tietokoneissa kannattaa olla kaksi näyttöä. Istumisjärjestelyä tärkeämpi tekijä vaikuttaa kuitenkin olevan yhteinen tavoite. Tässä luvussa käsiteltyjen tutkimusten perusteella tavoitteen osalta pariohjelmoinnissa kannattaa käyttää klassista asetelmaa, jossa kumpikin parin jäsen työskentelee yhteistä tavoitetta kohti.

4.2 Kommunikaatio-ohjeet

Kommunikaatio-ohjeiden antaminen pareille voi mahdollisesti tukea heidän kommunikaatiotaan. Pariohjelmoinnin käyttöön ohjelmointikursseilla on kehitetty erilaisia ohjeita, mutta nämä ohjeet eivät ole keskittyneet suoraan parin kommunikointiin (Bevan, Werner ja McDowell 2002; Williams ym. 2008). Zarb ja Hughes (2015) ovat pyrkineet kehittämään konkreettisia kommunikaatio-ohjeita aloitteleville pariohjelmoijille. He analysoivat tutkimuksessaan ammattiohjelmoijien kommunikaatiota pariohjelmoinnissa, ja analyysin pohjalta he pystyivät tunnistamaan pariohjelmoinnin tyypillisiä kommunikaatiomalleja (Zarb, Hughes ja Richards 2012; Zarb ja Hughes 2015). Nämä löydettyt kommunikaatiomallit olivat uudelleenkäynnistäminen (engl. *Restarting Pattern*), suunnittelu (engl. *Planning Pattern*) ja toiminta (engl. *Action Pattern*). Nämä mallit edustavat pariohjelmointiprosessin tyypillisiä vaiheita.

Löydettyjen mallien pohjalta Zarb ja Hughes (2015) kehittivät kirjalliset ohjeet siitä, mi-

ten pariohjelmoinnin tyypillisissä vaiheissa kannattaa toimia. Uudelleenkäynnistämisen ohje koskee tilanteita, joissa pari ei etene tehtävässä eteenpäin. Näissä tilanteissa paria neuvotaan rikkomaan keskittyminen ja keskustelemaan tehtävään liittymättömästä aiheesta. Tämän jälkeen paria neuvotaan arvioimaan uudelleen tehtyä työtä ja miettimään seuraavia askelia tehtävän ratkaisuun. Suunnitteluun liittyvän ohjeen tarkoitus on edistää tehtävän tekoa eteenpäin. Ohje kannustaa kumpaakin parin jäsentä tekemään ehdotuksia ja arvioita koodista. Tämän vaiheen ohjeen mukaan on myös tärkeää, että parin jäsenet kysyvät tarvittaessa selvennyksiä toistensa ideoista. Toiminnan ohje koskee tilanteita, joissa pari aktiivisesti kirjoittaa koodia. Ohje kannustaa kuskia ajattelemaan ääneen, jotta navigaattori pystyy helpommin seuraamaan työskentelyä. Navigaattoria taas kannustetaan tekemään kuskille ehdotuksia aktiivisesti.

Zarb ja Hughes (2015) testasivat ohjeiden toimivuutta koeasetelmassa, jossa joukko opiskelijoita jaettiin kahteen ryhmään: kummassakin ryhmässä annettiin ohjeeksi pariohjelmoida, mutta testiryhmä sai lisäksi kirjalliset kommunikaatio-ohjeet ja lyhyen videon niihin liittyen sekä hetken aikaa perehtyä tähän materiaaliin. Kun kokeen tulokset analysoitiin, huomattiin, että ryhmien välillä ei ollut merkittävää eroa valmiiksi saatujen ohjelmien määrien välillä, mutta testiryhmä raportoi merkittävästi suurempaa kommunikoinnin helppoutta. Zarb ja Hughes (2015) mittasivat myös testin jälkeisen kyselyn avulla sitä, kuinka suureksi parin jäsenet kokivat parin toisen jäsenen panostuksen tehtävään. Testiryhmässä raportoitiin merkittävästi korkeampaa kokemusta oman parin panostuksesta.

Zarbin ja Hughesin (2015) ohjeisiin on annettu mahdollinen jatkokehitysidea. McChesney (2016) toteutti kolme vuotta kestäneen pariohjelmointia käsittelevän toimintatutkimuksen. Hän huomasi opiskelijoiden kokevan erilaisia kommunikaatio-ongelmia, ja hänen tutkimuksessa moni näistä ongelmista liittyi etenkin käsitteiden kommunikointiin. McChesney (2016) ehdottaa, että kommunikaatio-ohjeiden lisäksi olisi hyvä myös pyrkiä tukemaan opiskelijoiden käsitystä tarvittavasta teknisestä sanastosta, sillä tämän sanaston puuttuminen hankaloittaa ajatuksien kommunikoimista.

Ohjelmoinnista kommunikoiminen toisen kanssa voi olla opiskelijoille hyvin vieras kokemus, eivätkä he mahdollisesti heti hahmota, miten pariohjelmoinnin aikana kannattaisi keskustella. Vaikuttaa siltä, että konkreettisten kommunikaatio-ohjeiden antaminen opiskelijoil-

le antaa heille tukea ja auttaa heitä keskustelemaan parinsa kanssa. Vaikka tässä luvussa esitellyn tutkimuksen tulokset ovat lupaavia, aiheesta tarvitaan lisää tutkimusta, jotta parhaat mahdolliset kommunikaatio-ohjeet pystytään kehittämään.

4.3 Parien muodostaminen

Parin jäsenten ominaisuuksilla voi olla vaikutusta pariohjelmoinnin toimivuuteen. Tämä tarkoittaa sitä, että jos parin muodostaa siten, että kumpikin jäsen on jonkin ominaisuuden suhteen samanlaisia tai erilaisia, voi pari olla yhteensopivampi kuin silloin, jos pari olisi muodostettu sattumanvaraisesti. Tyypillisesti opetuksessa käytettyä pariohjelmointia varten kehitetyissä ohjeissa mainitaan huomion kiinnittäminen parien muodostamiseen (Bevan, Werner ja McDowell 2002; Williams ym. 2008). Eri ominaisuuksien vaikutuksesta parin väliseen kommunikaatioon löytyy tutkimusta, ja seuraavaksi esitellään näitä tuloksia. Osa käsitellyistä tutkimuksista koskee parin yhteensopivuutta, mutta koska kommunikaation helppouden voi tyypillisesti olettaa olevan tekijä yhteensopivuuden tunteessa, käsitellään myös yhteensopivuuteen liittyvät tulokset.

4.3.1 Taitotaso

Parien muodostamiseen liittyvistä ominaisuuksista taitotaso on todennäköisesti kaikista tutkituin. Moni tutkija, kuten Bevan, Werner ja McDowell (2002) ja Bowman ym. (2019), suosittelee tutkimuksessa saamiensa tulosten pohjalta parien muodostamista siten, että parin jäsenten taitotaso on suunnilleen sama. Myös opiskelijat itse toivovat, että heidän parinsa olisi suunnilleen samalla taitotasolla (Melnik ja Maurer 2002; Williams ym. 2008). Katira, Williams ja Osborne (2005) huomasivat parien yhteensopivuutta käsittelevässä tutkimuksessaan, että pariohjelmointikokeilun jälkeen opiskelijat raportoivat suurempaa yhteensopivuuden määrää, kun he olivat työskennelleet samalla taitotasolla olevan henkilön kanssa.

Vaikuttaa siltä, että parien muodostaminen siten, että taidoiltaan samantasoiset opiskelijat työskentelevät yhdessä, auttaa etenkin heikompia opiskelijoita. Braught, MacCormick ja Wahls (2010) muodostivat pariohjelmointitutkimuksessaan kolme ryhmää: taitotason mukaan muodostetut pariohjelmointiparit, sattumanvaraisesti muodostetut pariohjelmointiparit

sekä yksin työskentelevien ryhmän. Heidän tuloksistaan kävi ilmi, että heikommat opiskelijat, jotka työskentelivät taitotason mukaan muodostetuissa pareissa, menestyivät paremmin kuin heikommat opiskelijat kahdessa muussa ryhmässä. Myös Bowman ym. (2019) tutkivat aiemman kokemuksen ja pariohjelmoinnin yhteyttä, ja heidän tuloksistaan kävi ilmi, että opiskelijat, jotka saivat parikseen itseään kokeneemman henkilön, kokivat tyypillisesti ponnostavansa pariaan vähemmän tehtäviin ja ymmärsivät heikommin kurssin konsepteja.

Moni tutkija selittää taitotasoeroon liittyviä tuloksia sillä, että parin jäsenten taitojen ollessa eri tasoilla parista taitavampi voi päätyä joko tietoisesti tai tiedostamattomasti tekemään kaiken työn, jolloin taidoiltaan heikompi jäsen päätyy vain seuraamaan työntekoa sivusta (Melnik ja Maurer 2002; Braught, MacCormick ja Wahls 2010; Bowman ym. 2019). Tämän kaltaisessa tilanteessa kommunikaatio jää hyvin vähäiseksi, eikä se muistuta toivottua puheliasta ja yhdessä pohtivaa pariohjelmoinnin ilmapiiriä.

4.3.2 Sukupuoli

Toinen tutkimusten kartoituksessa esiin noussut ominaisuus oli sukupuoli. Se, ovatko parin jäsenet samaa sukupuolta vai eivät, voi vaikuttaa parin kommunikaatioon ja yhteensopivuuteen. Aiheesta on tehty tutkimusta, ja osa tuloksista on ristiriidassa keskenään.

Choi (2015) tutki pariohjelmointia eri sukupuolista sekä samoista sukupuolista koostuneilla pareilla. Hänen tutkimuksessaan kävi ilmi, että sukupuoli ei ollut yhteydessä ohjelmointikykyyn, mutta samaa sukupuolta olevien parien keskuudessa ilmeni tilastollisesti merkittävästi korkeampaa kommunikaation ja yhteensopivuuden määrää. Saman tutkimuksen kokeen jälkeen kerätyistä kommentteista, joita etenkin naiset antoivat, kävi ilmi, että naiset tyypillisesti kokivat työskentelyn toisen naisen kanssa helpommaksi ja että miesten kanssa he kokivat joutuvansa puolustamaan osaamistaan enemmän.

Katira, Williams ja Osborne (2005) saivat Choin (2015) kanssa samankaltaisia tuloksia yhteensopivuuden osalta. He tutkivat parien yhteensopivuuden ja eri ominaisuuksien yhteyttä kahdella eri ohjelmointikurssilla. Tuloksien perusteella parit, jotka koostuivat eri sukupuolien edustajista, raportoivat epätodennäköisemmin yhteensopivuutta.

Kuttal, Gerstner ja Bejarano (2019) huomasivat pariohjelmointia ja sukupuolia käsitteleväs-

sä tutkimuksessaan, että naiset ilmaisivat haluavansa mieluummin työskennellä samaa sukupuolta olevan henkilön kanssa. Tämä tulos sopii yhteen Choin (2015) tutkimuksen tuloksien kanssa. Tuloksien perusteella naiset käyttivät myös enemmän non-verbaaleja viestejä miehiin verrattuna. Tässä tutkimuksessa huomiotavaa on kuitenkin se, että se toteutettiin etänä eli parin jäsenet eivät olleet samassa huoneessa vaan kommunikoivat videopuhelun välityksellä.

Kaikissa tutkimuksissa ei ole löydetty yhteyttä sukupuolen ja parin jäsenten yhteensopivuuden tai kommunikaation välillä. Demir ja Seferoglu (2021) tutkivat eri ominaisuuksien mukaan muodostettujen heterogeenisten ja homogeenisten parien yhteensopivuutta, flow-kokemusta ja suoriutumista. Sukupuoli oli yksi tutkittavista ominaisuuksista. Tutkimuksen tuloksien perusteella sukupuoleltaan heterogeenisten ja homogeenisten parien välillä ei ollut merkittävää eroa yhteensopivuuden tunteessa. Merkittävää eroa ei huomattu myöskään flow-kokemuksessa tai tehtävistä suoriutumisessa.

Zhong, Wang ja Chen (2016) ovat saaneet Demirin ja Seferoglun (2021) kanssa samanlaisia tuloksia. Myös heidän tutkimuksessa tutkittiin sukupuolen vaikutusta yhteensopivuuden tunteeseen, ja tuloksien perusteella parin jäsenten sukupuolella ei ollut merkittävää yhteyttä yhteensopivuuteen. Tutkimuksessa mitattiin myös tehtävistä suoriutumista ja itsevarmuutta, ja tuloksien perusteella parin jäsenten sukupuoli ei ollut myöskään näihin tekijöihin merkittävästi yhteydessä. On kuitenkin tärkeää huomata, että siinä missä muiden käsiteltyjen tutkimusten osallistajat olivat tyypillisesti täysi-ikäisiä yliopistojen opiskelijoita, tämän tutkimuksen osallistajat olivat alaikäisiä peruskoulun opiskelijoita. Zhong, Wang ja Chen (2016) toteavat, että osallistujien ikä voi mahdollisesti selittää erilaisia tuloksia ja että aihe vaatii lisää tutkimusta tulevaisuudessa.

4.3.3 Persoonallisuus

Viimeinen kartoituksessa esiin noussut ominaisuus parien muodostuksessa on persoonallisuus. Tutkimus persoonallisuuden yhteydestä pariohjelmointiin on tuottanut sukupuolen lailla erilaisia tuloksia. Suurimmassa osassa käsiteltävistä tutkimuksista persoonallisuuden määrittelyyn käytettiin Myers–Briggsin tyyppi-indikaattoria (MBTI). Myers–Briggsin tyyppi-

indikaattori jakaa persoonallisuuden neljään ulottuvuuteen, ja jokainen ulottuvuus koostuu kahdesta vastakkaisesta piirteestä (Sfetsos ym. 2009). Ulottuvuudet kuvaavat ihmisten tapaa saada energiaa, kerätä tietoa, tehdä päätöksiä ja työskennellä.

Sfetsos ym. (2009) tutkivat persoonallisuuden ja temperamentin vaikutusta pariohjelmoinnin tehokkuuteen, jota mitattiin parin suorituskyvyn ja yhteistyön toimivuuden avulla. He hyödynsivät persoonallisuuden ja temperamentin määrittämisessä David Keirseyn kehittämää Keirseyn Temperament Sorter -testiä (KTS), joka perustuu Myers–Briggsin tyyppi-indikaattoriin. Heidän tuloksensa viittasivat siihen, että persoonaltaan ja temperamentiltaan heterogeeniset parit ovat tehokkaampia. Yksi tekijä, jolla parin suorituskykyä mitattiin, oli kommunikaation määrä, ja tuloksien perusteella heterogeenisissä pareissa se oli merkittävästi suurempi. Yhteistyön toimivuutta mitattiin kokeen jälkeen täytettävän kyselyn avulla, ja yksi kysytyistä osa-alueista oli tyytyväisyys kommunikaatioon. Heterogeeniset parit raportoivat merkittävästi parempia tuloksia sen osalta. Heterogeeniset parit pärjäsivät myös monissa muissa mitattavissa asioissa homogeenisiä pareja paremmin: he muun muassa saivat paremmat pisteet tehtävistä ja olivat merkittävästi nopeampia.

Williams ym. (2008) saivat myös tutkimuksessaan tuloksia, jotka viittasivat siihen, että persoonallisuuksiltaan erilaiset henkilöt muodostavat yhteensopivampia pareja kuin persoonallisuuksiltaan samanlaiset henkilöt. Heidän tutkimuksessa Myers–Briggsin tyyppi-indikaattorin neljästä ulottuvuudesta tiedon keräämisen tapaan liittyvä ulottuvuus on merkittävä. Tämän ulottuvuuden vastakohtat ovat tosiasiallinen (engl. *sensing*) ja intuitiivinen (engl. *intuiting*). Tosiasialliset ihmiset keräävät tietoa enemmän suoraan kokemusten ja aistiensa kautta, kun taas intuitiiviset ihmiset keräävät tietoa enemmän tiedostamattomilla ja epäsuoremmilla tyyleillä (Carlson 1985). Williams ym. (2008) raportoivat, että tosiasiallisen ja intuitiivisen henkilön muodostama pari raportoi merkittävästi todennäköisemmin olevansa erittäin yhteensopiva.

Kaikkien tutkimusten tulokset eivät tue yhtä vahvasti persoonaltaan heterogeenisten pariin parempaa yhteensopivuutta. Katira ym. (2004) tutkivat pariohjelmoinnin pariin persoonallisuuden ja yhteensopivuuden yhteyttä hyödyntämällä edellisten tutkimusten tavalla Myers–Briggsin tyyppi-indikaattoria persoonallisuuden määrittelyssä. Huomattavaa on se, että Katira ym. (2004) tekivät tutkimuksen sekä uusille opiskelijoille tarkoitettulla ohjelmoin-

nin johdantokurssilla sekä opinnoissa pidemmälle päässeille tarkoitettulla ohjelmistotuotantoa käsittelevällä kurssilla. Heidän tuloksissaan persoonaltaan heterogeeniset parit raportoivat korkeampaa yhteensopivuutta johdantokurssilla. Samaa ilmiötä ei kuitenkaan esiintynyt opinnoissa pidemmälle päässeiden kurssilla.

Katira, Williams ja Osborne (2005) jatkoivat yhteensopivuuden tutkimusta edellisen tutkimuksen (Katira ym. 2004) pohjalta. He hyödynsivät yhä Myers–Briggsin tyyppi-indikaattoria persoonan määrittelyssä. Jatkotutkimuksen tuloksien perusteella persoonallisuudella ei ollut merkittävää yhteyttä parin jäsenten yhteensopivuuteen, ja Katira, Williams ja Osborne (2005) toteavat, että persoonallisuus ei ole kriittinen tekijä yhteensopivuuden kannalta. Huomioitavaa on, että tässä tutkimuksessa Katira, Williams ja Osborne (2005) tekivät tutkimuksen yhteensopivuuden kyselyn kahdella kurssilla, jotka oli suunnattu opinnoissa pidemmällä oleville opiskelijoille, kun taas edellisessä tutkimuksessa (Katira ym. 2004) merkittävä yhteys persoonallisuuden ja yhteensopivuuden välillä ilmeni nimenomaan aloittelijoille suunnatulla johdantokurssilla. Pariohjelmoinnin käyttö eri vaiheissa opintoja on aihe, jota kannattaa yhä jatkossa tutkia.

4.3.4 Johtopäätökset parien muodostamisesta

Parin jäsenten ominaisuuksien vaikutuksesta pariohjelmointiin ja parin väliseen kommunikaatioon löytyy jonkin verran tutkimusta, mutta se on myös alue, jota kannattaa tulevaisuudessa tutkia lisää. Tällä hetkellä etenkin persoonallisuuden ja sukupuolen osalta on mahdollista löytää ristiriidassa olevia tuloksia.

Kartoituksen pohjalta vaikuttaa siltä, että käsitellyistä ominaisuuksista taitotasoon liittyy eniten tutkimusta, ja monissa sitä koskevissa tutkimuksissa on huomioitu, että eri tasoisten opiskelijoiden muodostamassa parissa kommunikaatio voi jäädä hyvin vähäiseksi ja että taitavampi opiskelija voi dominoida työskentelyä. Ohjelmointiparien muodostaminen taitotaso huomioon ottaen vaikuttaa siis kannattavalta, vaikka tämä todennäköisesti aiheuttaa lisätyötä opettajille, jotka joutuvat tällöin kartoittamaan ennen parien muodostamista opiskelijoiden taitotasoa.

Sukupuolen ja persoonallisuuden huomioon ottaminen ei vaikuta yhtä kriittiseltä pariohjel-

moinnin kommunikaation kannalta. Parien muodostaminen sukupuolen mukaan tasa-arvoisessa nykymaailmassa ei olisi mahdollisesti sosiaalisesti hyväksyttävää. Persoonallisuuden kartoittamiseen liittyisi todennäköisesti suuri työmäärä, ja tällä hetkellä aiheesta ei ole tutkimusta niin paljon, että sitä voitaisiin pitää välttämättömänä. Nämä kaksi ominaisuutta voivat kuitenkin vaikuttaa parin kommunikaatioon, ja pariohjelmointia opettavien on hyvä olla tietoisia niihin liittyvistä tutkimustuloksista.

5 Yhteenveto

Tässä tutkielmassa tehtiin kirjallisuuskartoitus pariohjelmoinnin kommunikaation tukemiseen liittyvistä keinoista. Käsitellyjä keinoja olivat istumisjärjestely, parin yhteinen tavoite, kommunikaatio-ohjeet sekä parien muodostaminen taitotason, sukupuolen ja persoonallisuuden mukaan. Näistä keinoista parien muodostamiseen taitotason perusteella vaikuttaa liittyvän eniten tutkimusta ja sitä puoltavaa näyttöä. Myös muista menetelmistä on saatu lupaavia tuloksia, mutta joidenkin menetelmien kohdalla tutkimus on vielä vähäistä, kun taas joidenkin osalta löytyy huomattava määrä ristiriitaisia tuloksia. Mainituista menetelmistä olisi hyvä saada tutkimusta tulevaisuudessa, jotta mahdollisimman tehokkaiden tukikeinojen kehittäminen on mahdollista. Lisäksi itse pariohjelmoinnin kommunikaatioprosessista olisi suotavaa saada lisää tutkimusta, jotta kommunikaation mahdolliset hyvät ja huonot tekijät voitaisiin tunnistaa paremmin.

Tässä tutkielmassa esitetyt menetelmät valittiin siksi, koska ne nousivat esiin aiheeseen liittyvän kirjallisuuden kartoituksessa. On kuitenkin todennäköistä, että esitettyjen menetelmien lisäksi on muita tapoja tukea pariohjelmoinnin kommunikaatiota. Toivottavaa olisi, että tulevaisuudessa pyritäisiin löytämään ja kehittämään mahdollisesti uusia keinoja kommunikaation tukemiseen.

Yksi mielenkiintoinen tutkimuskohde, johon tulevaisuudessa kannattaisi perehtyä, on etäpariohjelointi (engl. *distributed pair programming*). Tämä tutkielma keskittyi pariohjelmoinnin kommunikaation tukemiseen lähiopetuksessa, ja vaikka istumisjärjestelyä lukuun ottamatta tässä tutkielmassa esitettyjä menetelmiä voi soveltaa myös etäpariohjelmoinnissa, on mahdollista, että etäpariohjelointiin liittyy sille uniikkeja kommunikaatio-ongelmia, jotka vaatisivat toisenlaisia ratkaisuja. Vuonna 2020 koronapandemia ja sen mukana tullut etäopetus on tehnyt etäpariohjelmoinnista vielä ajankohtaisemman aiheen, ja aiheesta saadaan toivottavasti tämän seurauksena lisää tutkimusta.

Lähteet

Begel, Andrew, ja Nachiappan Nagappan. 2008. “Pair Programming: What’s in It for Me?” Teoksessa *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, 120–128. ESEM ’08. Kaiserslautern, Germany: Association for Computing Machinery. ISBN: 9781595939715. <https://doi.org/10.1145/1414004.1414026>.

Bevan, Jennifer, Linda Werner ja Charlie McDowell. 2002. “Guidelines for the use of pair programming in a freshman programming class”. Teoksessa *Proceedings 15th Conference on Software Engineering Education and Training (CSEE&T 2002)*, 100–107. IEEE. <https://doi.org/10.1109/CSEE.2002.995202>.

Bowman, Nicholas A., Lindsay Jarratt, K.C. Culver ja Alberto Maria Segre. 2019. “How Prior Programming Experience Affects Students’ Pair Programming Experiences and Outcomes”. Teoksessa *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 170–175. New York, NY, USA: Association for Computing Machinery. ISBN: 9781450368957. <https://doi.org/10.1145/3304221.3319781>.

Braught, Grant, John MacCormick ja Tim Wahls. 2010. “The Benefits of Pairing by Ability”. Teoksessa *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 249–253. SIGCSE ’10. Milwaukee, Wisconsin, USA: Association for Computing Machinery. ISBN: 9781450300063. <https://doi.org/10.1145/1734263.1734348>.

Carlson, John G. 1985. “Recent assessments of the Myers-Briggs type indicator”. *Journal of personality assessment* 49 (4): 356–365. https://doi.org/10.1207/s15327752jpa4904_3.

Celepku, Mehmet, ja Kristy Elizabeth Boyer. 2018. “The Importance of Producing Shared Code Through Pair Programming”. Teoksessa *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 765–770. SIGCSE ’18. Baltimore, Maryland, USA: Association for Computing Machinery. ISBN: 9781450351034. <https://doi.org/10.1145/3159450.3159506>.

Choi, Kyungsub Stephen. 2015. “A comparative analysis of different gender pair combinations in pair programming”. *Behaviour & Information Technology* 34 (8): 825–837. <https://doi.org/10.1080/0144929X.2014.937460>.

Demir, Ömer, ja Süleyman Sadi Seferoglu. 2021. “The Effect of Determining Pair Programming Groups According to Various Individual Difference Variables on Group Compatibility, Flow, and Coding Performance”. *Journal of Educational Computing Research* 59 (1): 41–70. <https://doi.org/10.1177/0735633120949787>.

Katira, Neha, Laurie Williams ja Jason Osborne. 2005. “Towards increasing the compatibility of student pair programmers”. Teoksessa *Proceedings. 27th International Conference on Software Engineering, 2005. ICSE 2005*. 625–626. IEEE. <https://doi.org/10.1109/ICSE.2005.1553618>.

Katira, Neha, Laurie Williams, Eric Wiebe, Carol Miller, Suzanne Balik ja Ed Gehringer. 2004. “On Understanding Compatibility of Student Pair Programmers”. *SIGCSE Bull.* (New York, NY, USA) 36, numero 1 (maaliskuu): 7–11. ISSN: 0097-8418. <https://doi.org/10.1145/1028174.971307>.

Kuttal, Sandeep Kaur, Kevin Gerstner ja Alexandra Bejarano. 2019. “Remote Pair Programming in Online CS Education: Investigating through a Gender Lens”. Teoksessa *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 75–85. IEEE. <https://doi.org/10.1109/VLHCC.2019.8818790>.

Lima Salge, Carolina Alves de, ja Nicholas Berente. 2016. “Pair Programming vs. Solo Programming: What Do We Know After 15 Years of Research?” Teoksessa *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 5398–5406. IEEE. <https://doi.org/10.1109/HICSS.2016.667>.

McChesney, Ian. 2016. “Three Years of Student Pair Programming: Action Research Insights and Outcomes”. Teoksessa *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, 84–89. SIGCSE '16. Memphis, Tennessee, USA: Association for Computing Machinery. ISBN: 9781450336857. <https://doi.org/10.1145/2839509.2844565>.

- McDowell, Charlie, Linda Werner, Heather Bullock ja Julian Fernald. 2002. “The Effects of Pair-Programming on Performance in an Introductory Programming Course”. *SIGCSE Bull.* (New York, NY, USA) 34, numero 1 (helmikuu): 38–42. ISSN: 0097-8418. <https://doi.org/10.1145/563517.563353>.
- McDowell, Charlie, Linda Werner, Heather E. Bullock ja Julian Fernald. 2006. “Pair Programming Improves Student Retention, Confidence, and Program Quality”. *Commun. ACM* (New York, NY, USA) 49, numero 8 (elokuu): 90–95. ISSN: 0001-0782. <https://doi.org/10.1145/1145287.1145293>.
- Melnik, Grigori, ja Frank Maurer. 2002. “Perceptions of Agile Practices: A Student Survey”. Teoksessa *Extreme Programming and Agile Methods — XP/Agile Universe 2002*, toimittanut Don Wells ja Laurie Williams, 241–250. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-45672-8. https://doi.org/10.1007/3-540-45672-4_27.
- Muller, M. M., ja W. F. Tichy. 2001. “Case study: extreme programming in a university environment”. Teoksessa *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001*, 537–544. <https://doi.org/10.1109/ICSE.2001.919128>.
- Rodríguez, Fernando J., Kimberly Michelle Price ja Kristy Elizabeth Boyer. 2017. “Exploring the Pair Programming Process: Characteristics of Effective Collaboration”. Teoksessa *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, 507–512. SIGCSE '17. Seattle, Washington, USA: Association for Computing Machinery. ISBN: 9781450346986. <https://doi.org/10.1145/3017680.3017748>.
- Sfetsos, Panagiotis, Ioannis Stamelos, Lefteris Angelis ja Ignatios Deligiannis. 2009. “An experimental investigation of personality types impact on pair effectiveness in pair programming”. *Empirical Software Engineering* 14 (2): 187–226. <https://doi.org/10.1007/s10664-008-9093-5>.
- Simon, Beth, ja Brian Hanks. 2008. “First-Year Students’ Impressions of Pair Programming in CS1”. *J. Educ. Resour. Comput.* (New York, NY, USA) 7, numero 4 (tammikuu). ISSN: 1531-4278. <https://doi.org/10.1145/1316450.1316455>.

Umapathy, Karthikeyan, ja Albert D. Ritzhaupt. 2017. “A Meta-Analysis of Pair-Programming in Computer Programming Courses: Implications for Educational Practice”. *ACM Trans. Comput. Educ.* (New York, NY, USA) 17, numero 4 (elokuu). <https://doi.org/10.1145/2996201>.

Watson, Christopher, ja Frederick W.B. Li. 2014. “Failure Rates in Introductory Programming Revisited”. Teoksessa *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, 39–44. ITiCSE '14. Uppsala, Sweden: Association for Computing Machinery. ISBN: 9781450328333. <https://doi.org/10.1145/2591708.2591749>.

Werner, Linda L., Brian Hanks ja Charlie McDowell. 2004. “Pair-Programming Helps Female Computer Science Students”. (New York, NY, USA) 4, numero 1 (maaliskuu): 4–es. ISSN: 1531-4278. <https://doi.org/10.1145/1060071.1060075>.

Williams, Laurie, D Scott McCrickard, Lucas Layman ja Khaled Hussein. 2008. “Eleven guidelines for implementing pair programming in the classroom”. Teoksessa *Agile 2008 Conference*, 445–452. IEEE. <https://doi.org/10.1109/Agile.2008.12>.

Williams, Laurie, Charlie McDowell, Nachiappan Nagappan, Julian Fernald ja Linda Werner. 2003. “Building pair programming knowledge through a family of experiments”. Teoksessa *2003 International Symposium on Empirical Software Engineering, 2003. ISESE 2003. Proceedings*. 143–152. IEEE. <https://doi.org/10.1109/ISESE.2003.1237973>.

Williams, Laurie, ja Richard L. Upchurch. 2001. “In Support of Student Pair-Programming”. *SIGCSE Bull.* (New York, NY, USA) 33, numero 1 (helmikuu): 327–331. ISSN: 0097-8418. <https://doi.org/10.1145/366413.364614>.

Williams, Laurie, Eric Wiebe, Kai Yang, Miriam Ferzli ja Carol Miller. 2002. “In Support of Pair Programming in the Introductory Computer Science Course”. *Computer Science Education* 12 (3): 197–212. <https://doi.org/10.1076/csed.12.3.197.8618>.

Williams, Laurie A., ja Robert R. Kessler. 2001. “Experiments with Industry’s “Pair-Programming” Model in the Computer Science Classroom”. *Computer Science Education* 11 (1): 7–20. <https://doi.org/10.1076/csed.11.1.7.3846>.

Zarb, Mark, ja Janet Hughes. 2015. “Breaking the communication barrier: guidelines to aid communication within pair programming”. *Computer Science Education* 25 (2): 120–151. <https://doi.org/10.1080/08993408.2015.1033125>.

Zarb, Mark, Janet Hughes ja John Richards. 2012. “Analysing communication trends in pair programming videos using grounded theory”. Teoksessa *The 26th BCS Conference on Human Computer Interaction* 26, 1–4. <https://doi.org/10.14236/ewic/HCI2012.106>.

Zhong, Baichang, Qiyun Wang ja Jie Chen. 2016. “The impact of social factors on pair programming in a primary school”. *Computers in Human Behavior* 64:423–431. ISSN: 0747-5632. <https://doi.org/10.1016/j.chb.2016.07.017>.