Behaviour & Information Technology
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tbit20

A comparative analysis of different gender pair combinations in pair programming

Kyungsub Stephen Choi[a]
[a] Rhode Island College, Computer Information Systems, Providence, RI, USA

Published online: 24 Jul 2014.

Behaviour & Information Technology, 2015
Vol. 34, No. 8, 825–837, http://dx.doi.org/10.1080/0144929X.2014.937460

A comparative analysis of different gender pair combinations in pair programming

Kyungsub Stephen Choi*

Rhode Island College, Computer Information Systems, Providence, RI, USA

(*Received 18 November 2013; accepted 11 June 2014*)

Over the last decade, there has been a steady stream of pair programming studies. However, one significant area of pair programming that has not received its due attention is gender. Considering the fact that pair programming is one of the major human-centric software development paradigms, this is a gap that needs to be addressed. This empirical study conducted quantitative and qualitative analyses of different gender pair combinations within pair programming context. Using a pool of university programming course students as the experiment participants, the study examined three gender pair types: female–female, female–male, and male–male. The result revealed that there was no significant gender difference in the pair programming coding output. But there were significant differences in the levels of pair compatibility and communication between the same gender pair type: female–female and male–male, and the mixed gender pair type, female–male. The post-experiment comments provide additional insights and details about gender in pair interactions.

Keywords: gender; pair programming; extreme programming; group programming; pair learning

1. Introduction

Agile software development paradigm (http://agilemanifesto.org 2013) has been around for over a decade and it pithily serves the needs of software organisations that emphasise the shortest-time-to-market and practicality (Davenport and Short 1990; Smith and Reinertsen 1997). The paradigm offers a number of unorthodox software development approaches that centre on practical and human values, rather than structured methods and established models (Cockburn 2002). Among those approaches, Pair Programming of Extreme programming (Beck 2000) is the more visible and popular one. Pair programming is described as a programming technique where two programmers share one keyboard and one monitor, while cognitively collaborating to finish a programming task (Williams and Kessler 2002; Balijepally et al. 2009).

Given the pair formation, pair programming invites a number of tantalising research inquires. A number of pair programming studies have emerged, investigating the technical, economical, and human-centric dimensions of pair collaboration, with focuses on a range of topics, including effectiveness, productivity, costs and benefits, learning programming, pair types, virtual and distributed modes, programming proficiency, conflict management, cognitive disposition and personality (Cockburn and Williams 2001; Baheti, Gehringer, and Stotts 2002; Balijepally, Mahapatra, and Nerur 2003; Balijepally, Mahapatra, and Nerur 2006; Domino et al 2003; Heiberg et al. 2003; Padberg and Muller 2003; Mujeeb-u-Rehman et al. 2005; Preston 2005; Lui and Chan 2006; Dyba et al. 2007; Balijepally et al. 2009; Salleh et al. 2009; 2010; Walle and Hannay 2009; Hannay et al. 2010; Sennett and Sherriff 2010). Given such diverse topics, we expect to see a continuing increase in the number of studies.

Despite these streams of studies, one particular topic that has not received much attention is the gender topic – how men and women interact and collaborate in programming and dyad team work. Gender is an obvious variable; yet it is understudied in pair programming. Currently, there are several studies on gender-related pair programming (Werner, Hanks, and McDowell 2004; 2006; Werner, Denner, and Bean 2004; Werner et al. 2005; Werner, Campe, and Denner 2005; Werner and Denner 2009). While these studies provide an excellent source to learn about the topic, more studies that emphasise on the interactions between men and women are needed.

There are other significant reasons why further study of gender in pair programming is necessary. One is that the female–male, female–female, or male–male combinations present themselves as realistic choices in a workplace environment, in spite of the low number of women in programming. Given the resources and the demanding work schedules in a programming context, many managers will strategically deploy their best professional programmers to meet impending deadlines. Another reason is the fact that the government, the private sector, and educational institutions are steadfastly working together to increase the number of women in science, technology, engineering,

*Email: kchoi@ric.edu

and mathematics (STEM). The National Science Foundation ADVANCE programme (NSF 2013) specifically invites more female participation in the areas of science, engineering, and technology fields. The players in this initiative strongly believe that greater women participation will lead to a higher level of innovation, creativity, and competitiveness. This effort will ultimately yield a larger number of women in programming.

Pair studies from psychology, behavioural science, and cognitive science disciplines report the fact that there is a significant gender effect on human interaction and collaboration (Carli 1989; Derlega et al. 1989; Underwood, McCaffrey, and Underwood 1990; Sutter et al. 2009). These studies illuminate the impact of gender on socio-psychological, nonverbal, and cognitive behaviour, as well as on economic output; these areas of focus are similarly relevant to the complex social practices of pair programming.

The theoretical contribution is that this study's result will shed more light on the role of gender in pair programming which may lead to further studies on the significance of gender in programming. This study provides practical implications for programming shop managers, revealing information about the role of gender in pair programming as well as how such information might be applied in their operations to efficiently maximise their programming resources.

2. Literature review

Gender is a topical subject, which has been covered extensively throughout other disciplines with many different perspectives. Not to be too exhaustive, but still to delineate the scope, we sort and summarise relevant recent studies.

2.1. *Gender-focused pair programming studies*

Werner, Hanks, and McDowell (2004; 2006) and Werner et al. (2005) conducted studies that involved more than 500 students in programming courses and reported many favourable results for women who experienced pair programming than for women who experienced programmed alone. Having experienced pair programming, women exhibited a higher level of programming confidence, greater course completion and pass rates, program quality, test scores, confidence, enjoyment, and retention in computer-related majors; and they are more likely to persist in computer-related majors.

The studies also reason that pair programming presents an answer to a number of programming profession stigmas perceived by women. Many women view the programming profession as an unfriendly profession that fails to accommodate their lifestyle, family life, and personal safety. The perceptions are that (1) programming jobs are too competitive and the work is dominated by men, (2) the programming work is strictly solidary work in a confined location with not many friendly social interactions among coworkers, and (3) it poses safety and security concerns as programming work often stretches into long late night hours and sometimes into weekends in computer programming laboratories. Pair programming remedies such concerns through pair formation, thus resolving the anxieties associated with females programming alone. Moreover, it provides better group morale and social interactions, as well as fixed working hours since both programming partners must be present.

Another set of gender pair programming studies report how pair programming encourages young girls to pursue an interest in programming and increase IT fluency (Werner, Denner, and Bean 2004; Denner et al. 2005; Werner, Campe, and Denner 2005; Werner and Denner 2009). In the studies, 126 girls in sixth- through eighth-grade participated in interactive computer game design using Macromedia Flash MX. The findings are that pair programming positively affected the girls' level of IT fluency – their contemporary IT skills, fundamental grasp of IT concepts, and intellectual IT capabilities. McDowell et al. (2006) study had a participant group of 554 students from programming courses. Each participant was paired with a preferred partner. This naturally yielded three different group categories: male–male, male–female, and female–female. The study reported (1) no significant gender difference in average programming scores, (2) no significant gender difference in enjoyment and confidence, and (3) no interaction between gender and pairing. One particularly remarkable finding indicated that women showed a greater increase in confidence levels when working in pair programming in comparison to solo programming.

2.2. *Gender diversity*

Diversity is generally known to add a value on group's perspectives, problem-solving, information aggregation and more (Page 2007). A good example is gender diversity. Prior studies report that, instead of male dominant groups, groups with high percentage of women produced more favourable result. Francoeur, Labelle, and Sinclair-Desgagne (2007) study showed that, in a complex corporate environment, management teams with a high percentage of women officers generated positive and significant returns. Similarly, Campbell and MinguezVera (2007) studied the gender diversity in the management boardroom and firm's financial performance. They reported that the gender diversity significantly affects the high management's monitoring quality which consequently impacts the firm's financial performance. Adding it to this finding, Herring (2009) reported that gender diversity is significantly associated with increased sales revenue, more customers, and greater relative profits.

2.3. *IT fluency*

While other major disciplines are cautious and more reserved in claiming the existence of a gender gap (Damarin and Erchick 2010), the IS domain putatively presumes a gender difference (Adam, Howcroft, and Richardson 2004). With that, many IS studies that deal with the gender variable tend to fall under one of these two questions: (1) how is computing technology perceived and consumed differently by men as opposed to women? and (2) why is there such a low female population in the computing technology sector, what are the causes, and what can we do about this issue?

The gender perspective in IS generally lies with the notion that computing technology knowledge acquisition favours men (Adam 2002; Joshi and Kuhn 2007; Howcroft and Trauth 2008). In other words, men exhibit a significantly higher level of mastery of computing technology knowledge than women. To understand the reasons for this gender discrepancy, one may point to knowledge method and delivery (Gefen and Straub 1997; Venkatesh and Morris 2000), social and structural issues (Ahuja 2002), and psycho-social concerns (He and Freeman 2008).

In the knowledge method and delivery, Gefen and Straub (1997) emphasise the communication differences when men and women are introduced to technology. They report that men tend to 'focus discourse on hierarchy and independence, while women focus on intimacy and solidarity'. Cooper and Weaver (2003) and Barrett (2006) claim that knowledge method and delivery are more conducive to men than to women. For the social and structural issues, Ahuja (2002) and Cooper and Weaver (2003) mention the importance of social expectations, occupational culture, as well as the support and influence of peers and parents. They surmise that regardless of one's effort and attitude, our society and its social structure are not conducive to women achieving fluency in and enjoyment of computing technology as much as men. Psycho-social concerns constitute another reason that scholars have emphasised. Here, the major finding is that women exhibit a significantly higher level of anxiety than men in adopting computing technology (He and Freeman 2008).

A number of IS studies seek solutions to redress this gender gap. Given the report that we currently use computing technology that is a better fit for masculine users, the consensus in remedying the gender gap is to provide and accommodate the computing technology knowledge transfer method and delivery by placing an emphasis on feminist positivism (Adam 2002; Howcroft and Trauth 2008). One approach would be to shun today's authoritarian pedagogical style (Barrett 2006) and encourage more women participants by fostering an encouraging atmosphere for their skills (Cooper and Weaver 2003; Rosenberg-Kima et al. 2010). Some even suggest creating a women-only learning environment (Cooper and Weaver 2003). A prudent approach for both women and men would be to lower the identified 'barriers' that create the gender gap in the first place (Margolis and Fisher 2003), to instill current practices with feminist positivism (Adam 2002), to ensure a gender mix in group collaborations (Choi, Deek, and Im. 2009; Schiller et al. 2011), and to encourage and support women with effective persuasive messages (Rosenberg-Kima et al. 2010).

Werner and Denner (2009) have used pair programming to understand the women shortage problem in the science and engineering disciplines. Attributing this shortfall to a competitive and masculine culture, as well as to the low confidence levels of female individuals in problem-solving on the computer, the women shortage problem in those disciplines was microscopically examined through a group of middle school female students engaged in pair programming. A total of 126 girls were paired in groups of two, and given a computer programming task where each group had to make a set of decisions on the path to completing the given task. The pair programming sessions were also audio-recorded. The experiment results revealed that the girls lacked the computer-programming-task-required persistence, succumbing to the difficulties of the computer programming task. But through pair programming sessions, the girls complemented each other and supported one another to stay with the given task. The girls exercised metacognitive analysis in identifying and isolating a problem, determining and performing the necessary work, and testing the solution. These metacognitive steps were materialised through pair programming and team collaboration. This study underpins the notion that if female individuals are supported by their peers and others, and given safe-net facilitation, they can be productive even with cognitively conflicting tasks, such as a computer programming task.

3. Hypotheses development

For the pairing process, the available combinations are female–female, female–male, and male–male: two same-gender pairs and one mixed-gender pair. The sentiment of the literature review is that generally there are no significant differences between men and women (Murphy et al. 2006; Lau and Yuen 2009). Rather, the focus is on the interactive aspect of mixed-pair collaboration. The synergy between male–female would well complement each other in achieving a higher level of output.

A few studies report that men exhibit a greater propensity towards technology (Joshi and Kuhn 2007; Kulturel-Konak, D'Allegro, and Dickinson 2011); and men are also viewed as more action-oriented and goal-oriented (Adam, Howcroft, and Richardson 2004; Howcroft and Trauth 2008). Programming is considered a goal-oriented activity. Within a pair programming session, the intensity of goal-oriented behaviour escalates. The interaction between pair programmers, switching positions and continuous checks and balances on each other's work, places a substantial pressure on both programmers. If there are multi-tasking agile projects, which are often the case, the intensity escalates even more (Wijnands and Van Dijk 2007).

Under the demanding pair programming circumstances, conflicts and differences of opinion on coding strategy or practice are expected (Domino et al 2003). The situation also demands the compromise of two different programmers' problem solving styles, and can additionally pose 'pair fatigue' and 'pair pressure' (Wray 2010). To achieve a sound pair programming environment, these difficulties must be managed. Women demonstrate better skills than men to mitigate these difficulties. Research shows that women tend to focus on intimacy and solidarity (Gefen and Straub 1997), they foster an encouraging atmosphere (Cooper and Weaver 2003), and they support others with effective persuasive messages (Rosenberg-Kima et al. 2010), they show more concern about others as well as greater conformance to social norms (Gattiker and Nelligan 1998), and they manage well with ambiguity (Gutek and Larwood 1987).

Given the three pair categories, we believe that the synergy between female–male pair would enable productivity as well as manage potential conflicts to sustain a sound pair programming session.

> H1: The female–male pair would significantly achieve a higher score over other pair combinations in the following categories:
> H1.1: code design
> H1.2: code productivity

Different from the quantitative assessment, the qualitative assessment – confidence, compatibility, and communication – a different perspective takes place. The literature review points to the higher level of female–female harmonious relationship than the male–female or male–male combinations. Werner, Hanks, and McDowell (2004; 2006) and Werner et al. (2005) reported that female–female pairs exhibited a higher level of programming confidence, program quality, test scores, confidence, and enjoyment. The female–female pairs also complemented and supported each other to stay with the given task (Werner and Denner 2009). McDowell et al. (2006) study supports the idea that women showed a higher confidence level when pair programming than when programming solo.

The ability to communicate is particularly important. Wood (2001) discusses how women use it as a key instrument to bridge new relationships and to strengthen an existing relationship. Women communicate with others to learn more about them and also to share information about themselves (Johnson 1996). Women seek symmetry in communication: they seek equality between people (Aries 1987), where no one is feeling left out from a conversation, where everyone is invited to speak (Fishman 1978; Beck 1988), and they work to create a friendly atmosphere by connoting the intentions of others (Campbell 1973). Conversely, men primarily use communication as a tool to assert their ideas, defend their positions, preserve their independence, and enhance their status (Wood 2001). Compared to women, men exhibited a lesser degree of the qualities described above (Eakins and Eakins 1978; Beck 1988; Wood 1993; Stewart et al. 2002).

Based on these results, we expect a female–female pair combination would yield higher levels of confidence, compatibility, and communication than the two other pair combinations.

> H2: The female–female pair type would achieve significantly higher levels than the other two pair types in the following categories:
> H2.1: confidence
> H2.2: compatibility
> H2.3: communication

4. Methodology

The overall design of the experiment consisted of the initial preparations and the main experiment (see Figure 1). For the preparations, the following items were identified: secure the participant pool and pairing process, develop the programming tasks, administer the pilot experiment, identify the judges and the judging process, and develop the post-experiment questionnaire. The main experiment consisted of an information session, and two sessions of pair programming. The details of the experiment design are provided in the next section.

4.1. Preparations for the main experiment

Secure the participant pool and pairing process At a large state university, the introduction-to-programming course students were targeted for the experiment. The course instructors were briefed about the experiment and consulted for their cooperation. The experiment facilitator recruited the participants and, in return, the participants were given a course credit from their instructors for taking part. A total of 128 students – 93 males and 35 females – participated in this experiment.

In an information session, the participants tried out a practice session of pair programming, and filled out the consent form and programming background questionnaire. First, they attended a lecture about pair programming and its protocol, and then, they took part in a practice session.

In order to test the gender variable, other significant compounding variables were identified and controlled. The variables were: prior programming background, academic performance level, and familiarity. From the results of the earlier programming background questionnaire, no participant had any substantial programming background that appeared to be significant.

With the participants' consent, the participants' academic performance levels were acquired from the course instructors. The course instructors estimated the participants' grades were based on the participants' accumulated academic work. The familiarity variable in a small group

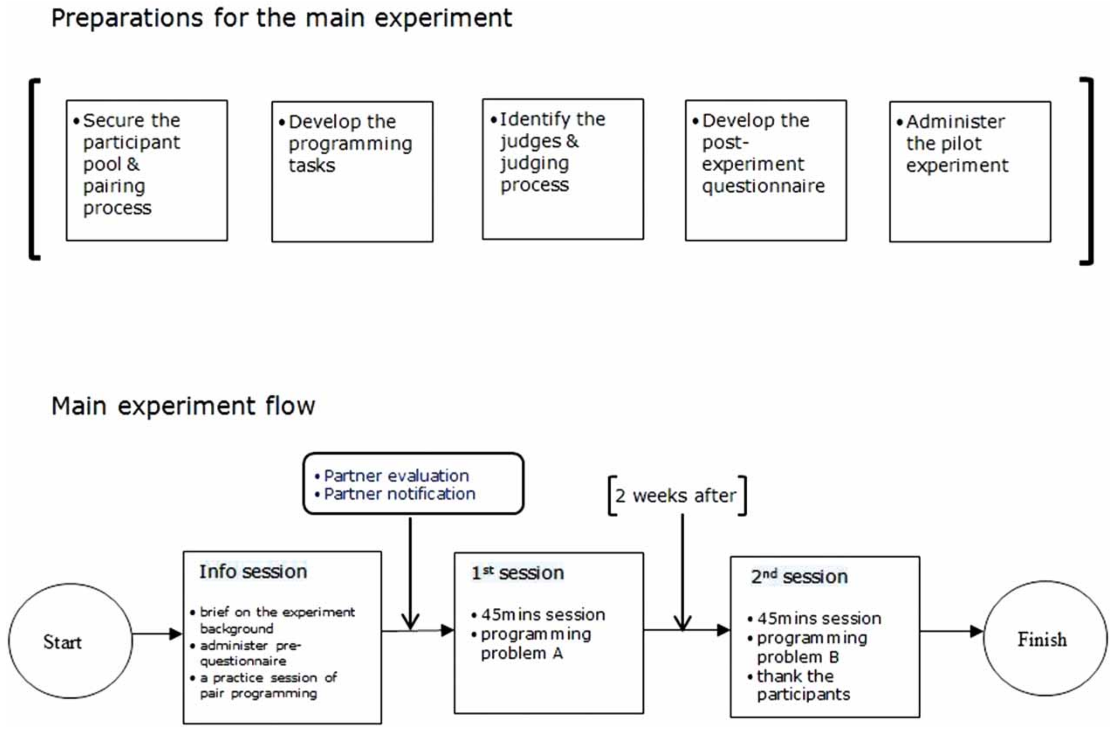

Figure 1. Experiment design.

or dyad is known to affect the process and its outcome (Orengo et al. 2000; Adams, Roch, and Ayman 2005; Janssen et al. 2009). To prevent familiarity within the pairs, the participants were asked about any friendship or close relationships. Also, the participants were asked for any significant level of familiarity with their partners.

After comparing and evaluating the participants' data and background, the pairing process was conducted. In reference to the above variables, the pairing process yielded 23 male–female pairs, 35 male–male pairs, and six female–female pairs. The low female–female pair number is due to the low female population enrolled in the programming course.

Develop the programming tasks The focus of devising the coding problems lies with the level of participants' programming ability. The problem needs to be doable for the participants; but at the same time, it has to be challenging enough to induce a high degree of pair interaction.

A collective effort, made up of the experiment facilitator, two outside judges, and course instructors, amassed and devised the coding problems. Three more external professional programmers were brought into the study and consulted for an objective evaluation. The validated problem set was used in the main experiment (Appendix 4). A few rounds of instrument fine-tuning the pilot experiments were executed prior to the main experiment. The session time duration, programming problem appropriateness, post-experiment questionnaire, and experiment protocol were identified for fine-tuning.

Identify the judges and judging process Two qualified independent programmers were asked to participate as judges. Both judges had an advanced degree in computer science and were fluent in coding. The judges contributed to the problem development, review, and modifications. The judges were given instructions on scoring (Appendix 1).

The judges reviewed and graded the subjects' work in two categories: (1) quantitative measurement (code productivity) and (2) qualitative measurement (code design). The grading scales are 0–10, with 0 being the lowest score and 10 being the highest score. If there is a difference in an item's scoring of more than 1 between the two judges then they are asked to reconcile the score.

An evaluation of how the judges scored the code output relatively to each other needs to be addressed. This is an inter-judge reliability. A bivariate correlation was conducted to generate the inter-judge reliability values. For the code productivity score correlation, we have 0.963, and for the code design score correlation, we have 0.967. These values indicate that the two judges are well within the tolerance range.

The participants' coding work and outputs from the two pair programming sessions were scored and averaged for the data analysis. The post-experiment questionnaire was administered after the completion of the second pair programming session.

Develop the post-experiment questionnaire The questionnaire was developed originally. The constructs are:

confidence, compatibility, and communication. The confidence refers to the level of confidence that the programmers have on their code output. The compatibility refers to how the two programmers get along during their coding session and how they manage any differences or conflicts. The communication refers to the intra-pair communication that takes place during coding.

4.2. Main experiment

Information session The experiment facilitator called all the participants to a common place for an information session. In the information session: (1) the subject's consent to the experiment form was signed and collected, (2) the participant's background questionnaire was administered (Appendix 2), (3) the participants were briefed on the experiment, and (4) the participants did a practice session of pair programming. In the practice session, the participants were alerted to role switching and reminded to switch the roles of 'pilot' and 'navigator'.

First session Before the first session, the partner information was given to each participant, and they were instructed to report to the experiment laboratory. The experiment was conducted at a large empty conference room equipped with PCs. Upon arrival, the experiment facilitator greeted the participants and led them to a pair programming workstation. One of the problem sets was issued for the first 45-min session. At each station, a camcorder was used to record the session by the experiment facilitator. The recording was used not only for verbal communication, but also to capture the seating proximity, the non-verbal communication, the role switching frequency, and any other nuanced and non-aural gestures. At the end of the session, the participants were told to turn in hardcopies of all coding work, including any error messages to the experiment facilitator.

Second session A repeat of the first session, the participants reported to the experiment laboratory and worked on their second problem. The session was video recorded and all hardcopies of their work were turned in. At the end, the experiment facilitator gave the full background of the experiment and answered any participant questions. The participants were thanked for their participation and time.

5. Data analysis and results

Analysis for H1 Table 1 presents the descriptive statistical analysis of the code output. The final data points or sample sizes (N) are: [FF] 12, [FM] 44, and [MM] 67. Due to an administrative error, two data points of [FM] and three data points of [MM] were lost.

Before determining the significant difference between the pairs, a data normality check was performed as routine. The result revealed that the data set was not normally distributed. This led to a non-parametric Kruskal–Wallis H (K–W) test, instead of ANOVA. Both code productivity (0.579, $p > 0.05$) and code design (0.443, $p > 0.05$) showed that there were no significant differences (see Table 2). In other words, although there were mean value differences between the pairs, those differences were *not statistically significant*. H1.1 and 1.2 were not supported.

Table 1. Code output descriptive statistics.

Dependent variable	Pair types	N	Mean	Std. dev.
Code prod.	[FF]	12	4.71	2.83
	[FM]	44	5.44	3.03
	[MM]	67	5.60	2.95
Code design	[FF]	12	4.67	2.63
	[FM]	44	5.42	3.01
	[MM]	67	5.60	2.88

Table 2. Coding output Kruskal–Wallis H test.

	Code prod.	Code design
χ^2	1.093	1.630
Df	2	2
Asymp. Sig.	.579	.443

Table 3. Kruskal–Wallis H test on the questionnaire data.

	Confidence	Compatibility	Communication
χ^2	.984	19.209	12.105
Df	2	2	2
Sig.	.612	.000	.002

Table 4. Mann–Whitney test on compatibility construct.

	[FM] vs. [MM]	[FF] vs. [FM]	[FF] vs. [MM]
Mann–Whitney U	3255.0	482.000	1136.000
Wilcoxon W	6576.0	3803.000	8639.000
Z	−4.123	−2.797	−.495
Sig.	.000	.003	.210

Analysis for H2 Similar to the H1 quantitative analysis, a data normality check was performed on the questionnaire data set. The check reported a failed data normality result. Kruskal–Wallis H test result (Table 3) revealed that only compatibility ($p = 0.000$) and communication ($p = 0.002$) were significantly different. Therefore, H2.1 was not supported. Further analyses on the compatibility and communication constructs were performed.

Mann–Whitney test was performed on each construct. The compatibility construct (Table 4) reported that (1) [MM] exhibited a significantly higher mean ($p = 0.000$, $p < 0.01$) than [FM], (2) [FF] showed a significantly higher mean ($p = 0.003$, $p < 0.01$) than [FM], and (3) no significance ($p = 0.210$, $p > 0.01$) between [FF] and [MM]. Same gender pairs, [FF] and [MM], exhibited significantly higher level of compatibility than mixed gender pair, [FM]. Hence, H2.2 was not supported.

Table 5. Mann–Whitney test on communication construct.

	[FM] vs. [MM]	[FF] vs. [FM]	[FF] vs. [MM]
Mann–Whitney *U*	3596.5	444.0	958.5
Wilcoxon *W*	6836.5	3684.0	8098.5
Z	−2.921	−2.808	−1.063
Sig.	.002	.003	.144

Table 6. Results analysis summary.

	Mixed-gender pair types		Same-gender pair type
	[FM] vs. [MM]	[FF] vs. [FM]	[FF] vs. [MM]
Confidence	No sig. diff	No sig. diff	No sig. diff
Compatibility	*Sig. diff* [FM] < [MM]	*Sig. diff* [FF] > [FM]	No sig. diff
Communication	*Sig. diff* [FM] < [MM]	*Sig. diff* [FF] > [FM]	No sig. diff

The communication construct (Table 5) reported that (1) [MM] showed a significantly higher mean ($p = 0.002$, $p < 0.01$) than [FM], (2) [FF] showed a significantly higher mean ($p = 0.003$, $p < 0.01$) than [MF], but (3) no significance ($p = 0.144$, $p > 0.01$) between [FF] and [MM]. Same gender pairs, [FF] and [MM], exhibited significantly higher level of communication than mixed gender pair, [FM]. Hence, H2.3 was not supported. Table 6 provides the summary.

The participants' comments In the post-experiment questionnaire, the participants' comments were collected, and a sample is presented here. The question was: 'Describe any conflicts or negative impacts from your partner that you have experienced during the pair programming which you think it (or they) influenced or caused the programming'. The following comments were from the females of the female–female pair type [FF]: (These comments are directly quoted):

> My partner was of the same sex (we are both female). I think this really helped because it made us feel more comfortable around each other. I really think that if I had worked with a guy, there would have been some initial awkwardness. Whether this would have been counter-productive or not would depend on the person and maybe if they came from a background that doesn't respect a woman's opinion'.

> I think it would have been more difficult if I were paired with a male instead. Some men like to monopolize a project like this. Also, some males make assumptions about your level before they get to see first-hand, so I was happy to get paired with a female.

> Gender is somewhat a factor. Since we are both females, it was easier to talk to and trust. If my partner was of the opposite sex, it may cause conflict. I may not be able to voice my suggestions or be too distracted.

> I think gender probably would play a role in this experiment and I felt more comfortable programming with another female.

The following were the comments from the female partners of female–male pair type [MF]:

> my partner was male, and tended to not explain what he was thinking or doing which was sometimes frustrating.

> I think gender caused us to be a little more reserved around each other. If I programmed with a female I probably would have been more open.

> I think men in general tend to view women as inferior in the technological arena, whether it is intentional or unintentional. I, myself, have had to prove myself on many occasions to my male counterparts. It is when the grades are announced that they 'accept' me, so to speak, as one of 'them'.

> I think this may have contributed to my partner's distrust. If I were male, I do think he would felt as though I was incompetent, but just may have thought that I was having a bad day or really needed a book for assistance. However, he made me feel as though even if I had a book, I still may not have known what to do. Of course this is speculation, and I do not think he meant it intentionally, but, I felt, his thoughts presented themselves unconsciously in his actions.

A similar effort was placed to locate comments from the male participants. But surprisingly, no similar comments were identified from the questionnaire forms. The males either had no comments or just a one-phrase comment such as 'no problem'.

6. Discussion

The coding output result supports the findings of earlier studies. McDowell et al. (2006) administered a pair programming experiment with similar gender pair types: female–female, female–male, and male–male. They found no significant gender difference in the programming scores. In other studies (Werner, Hanks, and McDowell 2004; 2006; Werner et al. 2005), female–female pair achieved satisfactory programming course completion and passing rates with high level of programming quality and test scores, and no gender difference. On the belief of gender difference in programming or science, technology, engineering, and mathematics (STEM), some point to the circumstantial behaviours or other-than-gender issues (Ahuja 2002; Hyde 2005; Hyde and Lindberg 2007; Kulturel-Konak, D'Allegro, and Dickinson 2011; Stoilescu and McDougall 2011) that caused the differences. The statistically not-significant result of this study and other earlier studies generally support the view that there is no difference between men's and women's capability in programming.

The questionnaire result, qualitative analysis, indicates that gender can be an issue for pair compatibility and pair intra-communication during a pair programming session or a collaboration for a cognitively challenging task. Compatibility and communication are in-process-related constructs whereas the confidence construct is after-process-related construct. Many of the participants' comments hint that gender can be an issue in pair programming context.

The female participants' comments vividly illustrate the level of compatibility: 'my partner was of the same sex (we are both female). I think this really helped because it made us feel more comfortable around each other…' 'my partner was male, and tended to not explain what he was thinking or doing which was sometimes frustrating…' 'my partner was male, and tended not to explain what he was thinking or doing which was sometimes frustrating,' 'since we are both females, it was easier to talk to and trust,' and 'if my partner was of the opposite sex, it may cause conflict. I may not be able to voice my suggestions or be too distracted'. These comments clearly expose the difficulties and conflicts that exist in mixed gender pair and support the high levels of compatibility and communication in the same gender pair.

Choi (2007) undertook a field survey on pair programming influencing factors with a group of professional programmers. The study includes a comment from a female professional programmer about the gender issue in pair programming that makes for instructive reading. Her comment states, 'Note that I am a female working in a team of males. I think that this is sometimes an issue when pairing.' Other male professional programmers add their comments to the question saying, 'I really don't have any experience pair programming with a female, but I don't know think gender should matter and if the team has respect for each other, and enjoy working with each other they will do well. Protocols, gender and skills will be completely eclipsed by a willingness to share ideas, and a desire to have conversations in voice and code at the same time.'

The gender-role theory (Connell 1987; Hardy 1995) talks about certain behaviours or norms that are generally expected in social relationship. Along with some of the existing information systems studies (Gutek and Larwood 1987; Gattiker and Nelligan 1998) the gender role theory may shed a light to why the women had behaved in such way and also why the women had commented in such way. Women are found to be more concerned about others and tend to conform more to societal norms, whereas men are driven more by individual attitudes and objectives.

One profound observation is that the female participants provided their colourful comments to illustrate the difficulties, but the male participants did not. Yet the male participants did reported similar significant compatibility and communication differences. This leads to the question of why there were no comments from the male participants in comparison to the female participants. One plausible explanation is that while men and women feel similar emotions, men generally do not overtly verbalise theirs emotions whereas women explicitly do. Wood (2001) mentioned that gendered verbal communication divide up men and women and allow each to identify with others of the same gender. More often, it is easier to talk to and collaborate with someone who shares the same or similar traits, without the aid of mental cues or verbal exchanges, the gender-role theory (Connell 1987; Hardy 1995). Johnson (1996) reported that talk is the essence of relationships for women than it is for men. We believe that there was a high level of intimacy between the two same gender partners, and the female partners were verbally explicit with their feelings after the programming sessions and the male partners were not.

Some of the comments infer that some female participants may have certain gender predispositions and they were more cognisant about gender. Those comments read, 'My partner was of the same sex (we are both female). I think this really helped because it made us feel more comfortable around each other,' 'I think it would have been more difficult if I were paired with a male instead,' and 'I think gender caused us to be a little more reserved around each other. If I programmed with a female, I probably would have been more open.' Again, no such comments from the male participants.

6.1. Limitations

The main limitation is the scant sample size of [FF] pairs. Clearly, six pairs is not a valid sample size in making a generalisable statement. But in today's environment, it is challenging to enlist enough female volunteering participants in making [FF] pairs in a study such as this. In a typical college programming course, it is hard to find female students. According to the US Census Bureau's 'Disparities in Science, Technology, Engineering, and Mathematics (STEM) Employment by Sex, Race, and Hispanic Origin' report (2013), there has been a steady decline in women's share of undergraduate degrees in computer science since 1980. There has also been a similar decline in women workforce in computer jobs; women represented 33% of mathematical and computer workforce in 1990, but it has dropped to 27% in 2011.

The scarcity of female programming students in college programming courses maybe is a reflection of this trend. Given these circumstances, it was difficult to identify enough female volunteering participants to create decent sample sizes for both [FF] and [FM] pair categories. Nonetheless, in statistical terms, the generalisation and inference from this study's result, specifically [FF] result, need to be interpreted appropriately with the aforementioned backdrop.

Arisholm et al. (2007) study revealed that the programmer's programming expertise may moderate the benefit of pair programming. Moreover, the level of prior programming experience may interact with the gender issue. The programming-related problem-solving and task completion abilities may influence the gender-difference conflicts. Another limitation is the use of students in the experiment. The students were mostly novice programmers. There can be a number of factors that may differ between a group of student programmers and a group of professional programmers. Professional programmers may transcend the gender differences with their maturity and professionalism.

7. Conclusion

This is a pair programming study that focused on gender difference. Using a pool of university programming course students as the experiment participants, the study examined three gender pair types: female–female, female–male, and male–male. On the quantitative analysis, the result reported no significant gender difference, but on the qualitative analysis, there were significant compatibility and communication level differences between same gender pair and mixed gender pair. From the post-experiment comments, the female participants were explicit in verbalising and sharing their partner experience.

This study contributes to the knowledge base of pair programming and agile development paradigm studies. The effort to increase the number of females in the programming world would also be helped by a better understanding of gender and its effect.

References

Adam, A. 2002. "Exploring the Gender Question in Critical Information Systems." *Journal of Information Technology* 17 (2): 59–67.

Adam, A., D. Howcroft, and H. Richardson. 2004. "A Decade of Neglect: Reflecting on Gender and IS, New Technology." *Work & Employment* 19 (3): 222–240.

Adams, S. J., S. G. Roch, and R. Ayman. 2005. "Communication Medium and Member Familiarity the Effects on Decision Time, Accuracy, and Satisfaction." *Small Group Research* 36 (3): 321–353.

Agilemanifesto.org. 2013. *Agilemanifesto* [online]. Accessed September 23, 2013. http://agilemanifesto.org/

Ahuja, M. 2002. "Women in the Information Technology Profession: A Literature Review, Synthesis and Research Agenda." *European Journal of Information Systems* 11 (1): 20–34.

Aries, E. 1987. "Gender and Communication." In *Sex and Gender*, edited by P. Shaver and C. Hendrick, 149–176. Newbury Park, CA: Sage.

Arisholm, E., H. Gallis, T. Dyba, and D. I. K. Sjøberg. 2007. "Evaluating Pair Programming with Respect to System Complexity & Programmer Expertise." *IEEE Transactions on Software Engineering* 33 (2): 65–86.

Baheti, P., E. Gehringer, and D. Stotts. 2002. "Exploring the Efficacy of Distributed Pair Programming." *Extreme Programming & Agile Methods – XP/AGILE Universe 2002*. Lecture Notes in Computer Science 2418, 387–410.

Balijepally, V., R. Mahapatra, and S. Nerur. 2003. "Cognitive Dispositions of IS Personnel and Implications for Pair Programming." Proceedings of Americas conference on information systems (AMCIS), Association of Information Systems, Tampa, FL, USA, August 4–6.

Balijepally, V., R. Mahapatra, and S. P. Nerur. 2006. "Assessing Personality Profiles of Software Developers in Agile Development Teams." *Communications of the Association for Information Systems* 18 (1): 55–75.

Balijepally, V., R. Mahapatra, S. Nerur, and K. Price. 2009. "Are Two Heads Better than One for Software Development? The Productivity Paradox of Pair Programming." *Management Information Systems Quarterly* 33 (1): 91–118.

Barrett, K. R. 2006. "Gender and Differences in Online Teaching Styles." In *Encyclopedia of Gender & Information Technology*, edited by E. M. Trauth, 372–377. Hershey: Idea Group Reference.

Beck, A. T. 1988. *Love is Never Enough*. New York: Harper & Row.

Beck, K. 2000. *Extreme Programming eXplained: Embrace Change*. 1st ed. Reading, MA: Addison-Wesley.

Campbell, K. K. 1973. "The Rhetoric of Women's Liberation: An Oxymoron." *Quarterly Journal of Speech* 59 (1): 74–86.

Campbell, K., and A. Minguez-Vera. 2007. "Gender Diversity in the Boardroom and Firm Financial Performance." *Journal of Business Ethics* 83 (3): 435–451.

Carli, L. L. 1989. "Gender Differences in Interaction Style and Influence." *Journal of Personality & Social Psychology* 56 (4): 565–576.

Choi, K. 2007. "Team Programming Influencing Factors: A Field Survey." *Journal of International Technology & Management* 18 (3–4): 1–13.

Choi, K. S., F. P. Deek, and I. Im. 2009. "Pair Dynamics in Team Collaboration." *Computers in Human Behavior* 25 (4): 844–852.

Cockburn, A. 2002. *Agile Software Development*. Boston, MA: Addison-Wesley.

Cockburn, A., and L. Williams. 2001. *The Costs & Benefits of Pair Programming, Extreme Programming Examined*. Boston, MA: Addison-Wesley Longman.

Connell, R. W. 1987. *Gender and Power: Society, the Person and Sexual Politics*. 1st ed. Stanford, CA: Stanford University Press.

Cooper, J., and K. D. Weaver. 2003, *Gender and Computers: Understanding the Digital Divide*. Mahwah, NJ: Lawrence Erlbaum.

Damarin, S., and D. B. Erchick. 2010. "Toward Clarifying the Meanings of Gender in Mathematics Education Research." *Journal for Research in Mathematics Education* 41 (4): 310–323.

Davenport, T., and J. Short. 1990. "The New Industrial Engineering: Information Technology & Business Process Redesign." *Sloan Management Review* 31 (4): 11–27.

Denner, J., L. Werner, S. Bean, and S. Campe. 2005. "The Girls Creating Games Program." *Frontiers* 26 (1): 90–98.

Derlega, V. J., R. J. Lewis, S. Harrison, B. A. Winstead, and R. Costanza. 1989. "Gender Differences in the Initiation & Attribution of Tactile Intimacy." *Journal of Nonverbal Behavior* 13 (2): 83–96.

Domino, M. A., R. W. Collins, A. R. Hevner, and C. F. Cohen. 2003. "Conflict in Collaborative Software Development." Proceedings of the 2003 SIGMIS conference on computer personnel research: freedom, in Philadelphia-leveraging differences and diversity in the IT workforce, 44–51. New York, NY: ACM.

Dyba, T., E. Arisholm, D. I. K. Sjoberg, J. E. Hannay, and F. Shull. 2007. "Are Two Heads Better than One? On the Effectiveness of Pair Programming." *IEEE Software* 24 (6): 12–15.

Eakins, B. W., and R. G. Eakins. 1978. *Sex Differences in Human Communication*. Boston, MA: Houghton Mifflin.

Fishman, P. M. 1978. "Interaction: The Work Women Do." *Social Problems* 25 (4): 397–406.

Francoeur, C., R. Labelle, and B. Sinclair-Desgagne. 2007. "Gender Diversity In Corporate Governance and Top Management." *Journal of Business Ethics* 81 (1): 83–95.

Gattiker, U. E., and T. Nelligan. 1998. "Computerized Offices in Canada & the United States: Investigating Dispositional Similarities & Differences." *Journal of Organizational Behavior* 9 (1): 77–96.

Gefen, D., and D. W. Straub. 1997. "Gender Differences in the Perception & Use of E-Mail: An Extension to the Technology Acceptance Model." *Management Information Systems Quarterly* 21 (4): 389–400.

Gutek, B. A., and L. Larwood. 1987. "Information Technology & Working Women in the USA." In: *Women & Technology*, edited by M. J. Davidson and C. L. Cooper, 71–94. Chichester: Wiley.

Hannay, J. E., E. Arisholm, H. Engvik, and D. I. Sjoberg. 2010. "Effects of Personality on Pair Programming." *IEEE Transactions on Software Engineering* 36 (1): 61–80.

Hardy, M. S. 1995. *The Psychology of Sexual Orientation, Behavior, and Identity: A Handbook*. Westport, CT: Greenwood Press.

He, J., and L. A. Freeman. 2008. "Are Men More Technology-Oriented Than Women? The Role of Gender on the Development of General Computer Self-Efficacy of College Students." *Journal of Information Systems Education* 21 (2): 203–212.

Heiberg, S., U. Puus, P. Salumaa, and A. Seeba. 2003. "Pair-Programming Effect on Developers' Productivity." In *Extreme Programming and Agile Processes in Software Engineering*, 215–224, Volume 2675/2003, 1016.

Herring, C. 2009. "Does Diversity Pay?: Race, Gender, and the Business Case for Diversity." *American Sociological Review* 74 (2): 208–224.

Howcroft, D., and E. M. Trauth. 2008. "The Implications of a Critical Agenda in Gender and IS Research." *Information Systems Journal* 18 (2): 185–202.

Hyde, J. S. 2005. "The Gender Similarities Hypothesis." *American Psychologist* 60 (6): 581–592.

Hyde, J. S., and S. M. Lindberg. 2007. "Facts & Assumptions about the Nature of Gender Differences and the Implications for Gender Equity." In *Handbook for Achieving Gender Equity through Education*, edited by S. S. Klein, 19–32. Mahwah, NJ: Lawrence Erlbaum.

Janssen, J., G. Erkens, P. A. Kirschner, and G. Kanselaar. 2009. "Influence of Group Member Familiarity on Online Collaborative Learning." *Computers in Human Behavior* 25 (1): 161–170.

Johnson, F. 1996. "Friendships Among Women: Closeness in Dialogue." In *Gendered Relationships*, edited by J. T. Wood, 79–94. Mountain View, CA: Mayfield.

Joshi, K. D., and K. M. Kuhn. 2007. "What it Takes to Succeed in Information Technology Consulting Exploring the Gender Typing of Critical Attributes." *Information Technology & People* 20 (4): 400–424.

Kulturel-Konak, S., M. L. D'Allegro, and S. Dickinson. 2011. "Review of Gender Differences in Learning Styles: Suggestions For STEM Education." *Contemporary Issues in Education Research* 4 (3): 9–18.

Lau, W. W. F., and A. H. K. Yuen. 2009. "Exploring the Effects Of gender & Learning Styles on Computer Programming Performance: Implications for Programming Pedagogy." *British Journal of Educational Technology* 40 (4): 696–712.

Lui, K. M., and K. C. C. Chan. 2006. "Pair Programming Productivity: Novice? Novice vs. Expert? Expert." *International Journal of Human–Computer Studies* 64 (9): 915–925.

Margolis, J., and A. Fisher. 2003. *Unlocking the Clubhouse: Women in Computing*. Boston, MA: MIT Press.

McDowell, C., L. Werner, H. E. Bullock, and J. Fernald. 2006. "Pair Programming Improves Student Retention, Confidence, & Program Quality." *Communications of the ACM* 49 (8): 90–95.

Mujeeb-u-Rehman, M., X. Yang, J. Dong, and A. M. Ghafoor. 2005. "Heterogeneous and Homogenous Pairs in Pair Programming: An Empirical Analysis." Canadian conference on electrical and computer engineering, 1116–1119. Silver Spring, MD: IEEE Press.

Murphy, L., B. Richards, R. McCauley, B. B. Morrison, S. Westbrook, and T. Fossum. 2006. "Women Catch Up: Gender Differences in Learning Programming Concepts." Proceedings of the 37th technical symposium on computer science education, 17–21. New York: ACM.

National Science Foundation (NSF). 2013, *ADVANCE: Increasing the Participation & Advancement of Women in Academic Science & Engineering Careers* [online] National Science Foundation. Accessed February 1, 2013 http://www.nsf.gov/crssprgm/advance/index.jsp

Orengo, C. V., A. A. M. Zornoza, F. Alonso, and J. M. Peiró Silla. 2000. "The Influence of Familiarity Among Group Members, Group Atmosphere and Assertiveness on Uninhibited Behavior through Three Different Communication Media." *Computers in Human Behavior* 16 (2): 141–159.

Padberg, F., and M. M. Muller. 2003. "Analyzing the Cost and Benefit of Pair Programming." Proceedings of ninth international software metrics symposium, 166–177. Silver Spring, MD: IEEE Press.

Page, S. E. 2007. *The Difference: How the Power of Diversity creates Better Groups, Firms, Schools, and Societies*. Princeton, NJ: Princeton University Press.

Preston, D. 2005. "Pair Programming as a Model of Collaborative Learning: A Review of the Research." *Journal of Computing Sciences in Colleges Archive* 20 (4): 39–45.

Rosenberg-Kima, R. B., E. A. Plant, C. E. Doerr, and A. L. Baylor. 2010. "The Influence of Computer-based Model's Race & Gender on Female Students' Attitudes & Beliefs Towards Engineering." *Journal of Engineering Education* 99 (1): 35–44.

Salleh, N., E. Mendes, J. Grundy, and G. S. J. Burch. 2009. "An Empirical Study of the Effects of Personality in Pair Programming using the Five-factor Model." Proceedings of the 2009 3rd international symposium on empirical software engineering and measurement, 214–225. Silver Spring, MD: IEEE Computer Society.

Salleh, N., E. Mendes, J. Grundy, and G. S. J. Burch. 2010. "The Effects of Neuroticism on Pair Programming: An Empirical Study in the Higher Education Context." Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement, 22. New York: ACM.

Schiller, S., F. Nah, B. Mennecke, and K. Siau. 2011. "Gender Differences in Virtual Collaboration on a Creative Design Task." Proceedings of the 32nd international conference of information systems (ICIS), Association of Information Systems, Shanghai, China, December 4–7.

Sennett, J., and M. Sherriff. 2010. "Compatibility of Partnered Students in Computer Science Education." Proceedings of the 41st ACM technical symposium on computer science education, 244–248. New York: ACM.

Smith, P. G., and D. G. Reinertsen. 1997. *Developing Products in Half the Time*. 2nd ed. New York: Wiley.

Stewart, L. P., P. J. Cooper, A. D. Stewart, and S. A. Friedley. 2002. *Communication & Gender*. 4th ed. Boston: Allyn & Bacon.

Stoilescu, D., and D. McDougall. 2011. "Gender Digital Divide & Challenges in Undergraduate Computer Science Programs. *Canadian Journal of Education* 34 (1): 308–333.

Sutter, M., R. Bosman, M. G. Kocher, and F. van Winden. 2009. "Gender Pairing and Bargaining—Beware the Same Sex!" *Experimental Economics* 12 (3): 318–331.

Underwood, G., M. McCaffrey, and J. Underwood. 1990. "Gender Differences in a Cooperative Computer-based Language Task." *Educational Research* 32 (1): 44–49.

US Census Bureau. 2013. "American Community Survey Reports: Disparities in STEM Employment by Sex, Race, and Hispanic Origin." US Department of Commerce, Economics and Statistics Administration, September. http://www.census.gov/prod/2013pubs/acs-24.pdf

Venkatesh, V., and M. G. Morris. 2000. "Why Don't Men Ever Stop to Ask for Directions? Gender, Social Influence, & their Role in Technology Acceptance & Usage Behavior." *Management Information Systems Quarterly* 24 (1): 115–139.

Walle, T., and J. E. Hannay. 2009. "Personality and the Nature of Collaboration in Pair Programming." *Empirical Software Engineering and Measurement, 2009. ESEM 2009. 3rd International Symposium on Digital Object Identifier*, 2009, 203–213. Silver Spring, MD: IEEE Press.

Werner, L., S. Campe, and J. Denner. 2005. "Middle School Girls + Games Programming = Information Technology Fluency." Proceedings of the 6th conference on information technology education, 301–305. New York: ACM.

Werner, L., and J. Denner. 2009. "Pair Programming in Middle School: What Does it Look Like?" *Journal of Research on Technology in Education* 42 (1): 29–49.

Werner, L., J. Denner, and S. Bean. 2004. "Pair Programming Strategies for Middle School Girls." Proceedings of the seventh international conference on computers & advanced technology in education, August 16–18. Kauai, Hawaii: ACTA Press.

Werner, L., B. Hanks, and C. McDowell. 2004. "Pair-programming Helps Female Computer Science Students." *ACM Journal on Educational Resources in Computing* 4 (1): 1–8.

Werner, L., B. Hanks, and C. McDowell. 2006. "Pair Programming & Gender." In *Encyclopedia of Gender & Information Technology*, edited by E. Trauth, 957–962. Hershey, PA: IGI.

Werner, L., B. Hanks, H. McDowell, H. Bullock, and J. Fernald. 2005. "Want to Increase Retention of Your Female Students?" *Computing Research News* 17 (2): 2.

Wijnands, R., and I. Van Dijk. 2007. "Multi-tasking Agile Projects: The Focal Point." In *Agile Processes in Software Engineering and Extreme Programming*, 222–225. Heidelberg: Springer.

Williams, L., and R. R. Kessler. 2002. *Pair Programming Illuminated*. Boston: Addison-Wesley.

Wood, J. T. 1993. "Engendered Relationships: Interaction, Caring, Power, and Responsibility in Close Relationships." In *Processes in Close Relationships: Contexts of Close Relationships*, Vol. 3. edited by S. Duck, 26–54. Beverly Hills, CA: Sage.

Wood, J. T. 2001. *Gendered Lives: Communication, Gender, & Culture*. 4th ed. Belmont, CA: Wadsworth.

Wray, S. 2010. "How Pair Programming Really Works." *IEEE Software* 27 (1): 50–55.

Appendix 1. Instruction sheet for the judges

Instruction for the Judges:

There are two categories; quantitative measure (How much of coding was done?) and qualitative measure (How are the codes?). The scale is 0–10, 10 being the highest score that a judge can give.

✤ *Quantitative Measurement* – Assign an appropriate score based on code productivity
  ➢ Coding activity that shows the programmer has attempted to solve the problem. However one or two lines are not acceptable.
  ➢ Coding activity that shows the programmer has completed almost half of the coding work
  ➢ Coding activity that shows the programmer has either completed all the coding work (including output) or close to completing the coding work

✤ *Qualitative Measurement* – Assess an appropriate score based on code design (structured, modular, or object-oriented approach), code efficiency, and code readability

Appendix 2. Subject programming background information

This form is used to collect the programming experience and background from the subjects.

Your Name:
Your ID:
School Year:
Major:
Your current programming course:

List all programming courses that you have taken
List all your professional programming experiences (Jobs, How long)
Have you practiced extreme programming before? If so, where and how long?
Have you practiced pair programming before? If so, where and how long?

Appendix 3

Survey Questions	Comm.	Conf.	Compat.
• My partner's active communication to me allowed me to be more active in expressing my views as well.	.790		
• My partner's hand gestures, eye gaze, body positions and other communication cues were NOT used adequately and poorly managed in our communication.	.765		
• My partner's voice tone was loud and clear which helped our communication	.742		
• My partner did not express nor communicated much (too quiet) which made PP very difficult	.735		
• My partner's message delivery was unclear which made programming very difficult	.732		
• My partner described his or her point very well for me to fully understand	.692		
• I really enjoyed my partner's partnership (willing to work as a team and open mind)	.511		
• I believe that our code is readable that others can follow and understand it with no problem		.856	
• I am confident that our work (output) is done correctly.		.791	

Survey Questions	Comm.	Conf.	Compat.
• I believe that our code is efficiently written (with less number of lines).		.783	
• I believe that I had an easier time reaching the solutions.		.557	
• My partner and I disagreed frequently in reaching a solution			.785
• There were times when I was withdrawn (or maybe upset) because of the disagreements from the partner			.689
• My partner insisted on doing things his way and/or did not collaborate			.649

Extraction method: principal component analysis.
Rotation method: Varimax with Kaiser normalisation.

Appendix 4. Problem set for programming session

Problem A: Online Supermarket

Your team is asked to create a program that is to be used by an online supermarket. You need to observe the following;

1. Each item in the store has a standard price that is up to two decimal places in precision.
2. A state sales tax (7%) is charged on all items except: bottled water, toilet paper, and fresh fruit.
3. A customer must buy at least five items, which includes two or three promo items and one non-taxed item.
4. A customer may buy more than one quantity per item.
5. The extended price (unit price*qty) for an item shall never be less than zero.
6. The store may run a promotion where the customer can buy three for a special price. If the customer buys less than three while the promotion is in effect the price is equal to the special price divided by the quantity rounded down to two decimal places.

Items	*Price* ($)	*Promotions*?	*Promo Price* ($)
A loaf of bread	1.99		
A box of orange juice	5.99		
One pound of apples	2.99		
A box of ice cream	3.99		
A set of two rolls of paper towel	1.99	Yes	3 for $4.99
A bottled water	0.99		
A bottle of ketchup	4.99		
Pecan pie	3.99		
A box of laundry detergent	8.99	Yes	3 for $14.99
A bottle of olive oil	2.99		
One bag of grapes	4.99		
A set of 12 rolls of toilet paper	9.99	Yes	3 for $16.99
A case of tofu	1.99		
A bag of imported black beans	2.99		

A shopper would take the printout and go to the supermarket to pay and pick up the grocery items.

Problem B: Employee Pay

Write a program called Pay, which calculates the pay for an employee's. Assume the user inputs employee name, status (i.e. full/part time), number of hours worked and pay rate on the command line in that order.

Your program should output the employee name, number of hours worked and pay.

When the employee worked 40 hours or less he/she should get regular pay

When the employee worked between 40 and 60 h (from 41 to 60) and he/she is a full time employee he/she should get double overtime, if he/she is a part time employee then he/she should get time and a half.

When the employee worked between 60 and 80 h (from 61 to 80) and he/she is a fulltime employee he/she should get triple overtime, if he/she is part time then he/she should get double overtime.

When he works more than 80 h, nothing is calculated and your program should print an error message that states, 'meet with employee'.

Appendix 5. Sample code work result of female–female pairs

Pairs	Sessions		Problem A			Problem B	
			P	D		P	D
7222 and 4959	1	Judge 1	6.0	6.0			
		Judge 2	5.0	5.0			
		Avg.	5.5	5.5			
	2				Judge 1	8.0	7.0
					Judge 2	8.0	8.0
					Avg.	8.0	7.5
7729 and 8472	1	Judge 1	1.0	1.0			
		Judge 2	2.0	1.0			
		Avg.	1.5	1.0			
	2				Judge 1	1.0	1.0
					Judge 2	1.0	1.0
					Avg.	1.0	1.0
2350 and 9566	1	Judge 1	6.0	5.0			
		Judge 2	5.0	5.0			
		Avg.	5.5	5.0			
	2				Judge 1	8.0	7.0
					Judge 2	9.0	8.0
					Avg.	8.5	7.5
1375 and 2103	1	Judge 1	2.0	2.0			
		Judge 2	1.0	2.0			
		Avg.	1.5	2.0			
	2				Judge 1	4.0	7.0
					Judge 2	4.0	6.0
					Avg.	4.0	6.5
1120 and 5058	1	Judge 1	2.0	3.0			
		Judge 2	3.0	4.0			
		Avg.	2.5	3.5			
	2				Judge 1	3.0	2.0
					Judge 2	2.0	2.0
					Avg.	2.5	2.0
6074 and 7734	1	Judge 1	8.0	6.0			
		Judge 2	7.0	7.0			
		Avg.	7.5	6.5			
	2				Judge 1	9.0	8.0
					Judge 2	8.0	8.0
					Avg.	8.5	8.0