



UNIVERSITY OF GOTHENBURG



## Analyzing the Efficiency of Pair Programming in Education

*Bachelor of Science Thesis in the Programme Software Engineering & Management*

XINRAN HE  
YUWEI CHEN

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
Göteborg, Sweden, January 2014

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

### **Analyzing the Efficiency of Pair Programming in Education**

XINRAN HE,  
YUWEI CHEN,

© XINRAN HE, January 2014,  
© YUWEI CHEN, January 2014.

Examiner: MORGAN ERICSSON

University of Gothenburg  
Chalmers University of Technology  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Cover:  
Pair programming

Department of Computer Science and Engineering  
Göteborg, Sweden January 2014

# Analyzing the Efficiency of Pair Programming in Education

## **Abstract:**

*Pair programming has been widely implemented in education because of the benefits it brings to fresh students in university. However, the efficiency of implementing pair programming in education is affected by many factors. This research investigates the most common factors which influence pair programming effectively and gives recommendations to teachers implementing pair programming in an efficiency way. Based on a systematic literature review and semi-structured interviews with 5 students, we investigate the most common benefits and factors of pair programming in education. As a consequence of our findings, we propose the "full pair programming process" to improve the quality of pair programming implementation.*

**Keywords:** Pair programming in Education, Pair programming in classroom, Pair programming

## **1. Introduction**

The Talmud, the ancient authoritative body of Jewish tradition, states that three things are needed to learn: a text, a teacher, and a learning partner [17]. Traditionally, learning to program is a solitary activity [19]. However, students need to use others as resources even when they work alone. Pair programming offers students a learning partner at beginning and transforms the solitary activity into collaborative learning.

Pair programming is a key component of the Extreme Programming software development methodology [17]. It rises with popularization of extreme programming in the industrial and education field. As the name suggests, two programmers work side-by-side at the same machine. One programmer plays the role called "driver" and who is responsible for operating the keyboard. While the driver is writing code, the other programmer (the "navigator") observes the driver's work and offers suggestions. They should switch roles frequently [2].

Research [5, 12, 13, 14, 27, 30] shows that pair programming provides significant benefits to programming: improving confidence of programmers; making the programming process enjoyably; producing code with higher quality. Compared with pair programming in industry, where additional factors such as double labor cost and simple task ineffectiveness have to be considered, pair programming in education earns more benefits. The main goal for students is to learn/gain knowledge so that the labor cost does not matter as much. For example, regarding the simple task issue, a professional programmer doing solo programming is very effective [5]. But students are more junior programmers, and during their education, get tasks that are challenging with respect to their programming level.

This research focuses on education. We applied systematic literature review and semi-structured interviews method with five undergraduate students to collect data in order to discuss the factors that affect the effectiveness of pair programming in education most. We find that the way the instructors implement pair programming influence the effectiveness of pair programming. And based on this finding, we suggest instructors implementing *full pair programming process* to improve the teaching quality, maximize the benefits of pair programming in education.

## 2. Research Design

### 2.1 Research Strategy

This research is based on a qualitative approach: a systematic literature review and semi-structured interviews. This qualitative research aims to gather an in-depth understanding of pair programming and the factors that related to the effectiveness of pair programming.

Systematic literature review is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest [11]. Systematic literature review is implemented as a research method in this report which aims to find the most common factors influence the efficiency of pair programming.

In addition, we relied on semi-structured interviews with undergraduate students. Intended to ensure that the same general areas of information are collected from each interviewee, this method provides more focus on the topic rather than “going with the flow”, but still allows a degree of freedom and adaptability to get information from the interviewee [25].

### 2.2 Research Question

Because of the benefits that pair programming brings; it has been implemented in education for many years. This research focuses on understanding and identifying the most common factors that influence the efficiency of implementing pair programming in education. Therefore, this systematic literature review focus on answering the following research questions:

- Which the factors influence the efficiency of pair programming in classroom/education?
- How to improve the efficiency of pair programming process?

### 2.3 Data Collection Procedures

#### **Systematic literature review:**

In order to search the most appropriate literatures which relate to the research questions, IEEE Xplore and ACM Library are used as searching database. The IEEE Xplore digital library is a powerful resource for discovery and access to scientific and technical content published by the IEEE (Institute of Electrical and Electronics Engineers) and its publishing partners<sup>1</sup>.

ACM is the world’s largest educational and scientific computing society, delivers resources that advance computing as a science and a profession. ACM provides the computing field’s premier Digital Library and serves its members and the computing profession with leading-edge publications, conferences, and career resources<sup>2</sup>. In addition, Google Scholar is also

---

<sup>1</sup><http://ieeexplore.ieee.org/xpl/aboutUs.jsp>

<sup>2</sup><http://www.acm.org/>

implemented in order to complement the data collection of systematic literature review. The following keywords are searched in IEEE Xplore, ACM Library, and Google Scholar database:

- Pair programming & Classroom
- Pair programming & Education
- Pair programming & Course
- Pair programming & Learning
- Pair programming & Teaching

In order to find the primary studies relating to our research topics, we designed an inclusion and exclusion criteria which is based on the research question. After searching the keywords in both databases, we include the literatures which have these keywords in abstract or introduction part. In addition, in our inclusion criteria, every literature mentions gender, personality, partner choose, factors which influence the efficiency of pair programming or the benefits of pair programming are all needed to study. Any literature studies pair programming in industry or compares the efficiency of pair programming and solo programming through counting the labor cost as their primary study are all excluded. What's more, some other criteria are found after the full texts of literature have been studied that are not relevant to our research topic, such as, a) literature introduces a tool which can improve the efficiency of pair programming. b) The experiment sample are relatively small/limited that cannot provide the most correct result. c) The distributed pair programming are excited which we will not consider as well.

82 literatures are listed after first searching based on IEEE Explore and ACM Library, 43 are excluded according to our exclusion criteria, and it also contains the literatures which are published in both databases. The final number of the literature which relates to our research topics is 39.

### **Interviews:**

Five bachelor students in Gothenburg University were interviewed, four males and one female students of which one student was in his first year, two students in their second year, and two students in their third year. We interviewed them in Gothenburg University face to face and discussed about the pair programming related to the topics listed below. The interview lasted between 10 and 30 minutes, three of the interviews were recorded.

The first question asked students about some basic personal information, for example: name, gender, name of the course which implementing pair programming. This question was used as a warm-up question and the results were not reported in the Part 3. We discussed the pair programming around the following topics which were derived from the results of our literature review:

1. Describe pair programming in your own words and how do you implement pair programming?
2. Did you have any training in pair programming?
3. How did you get your partner (teacher assigned or random or choose by yourself)?
4. Were you monitored by instructors (do you have supervision during pair programming)?
5. Compare your and your partner's programming capability and did you enjoy the pair programming process?

6. Did you get any benefits from implementing pair programming and could you describe it please?
7. Did you meet any troubles when using pair programming?
8. What was your best/worst moment when implement pair programming?
9. Would you recommend using pair programming in the class and could you offer some suggestions to improve pair programming process?

In the program Software Engineering and Management in Gothenburg University, pair programming were not required in the courses. Some students heard it from the course or the internet and then they used it for their assignment or project as their own choice. So none of them had training about pair programming from a teacher. They used pair programming in different courses for different project, for instance: Java, Embedded and Erlang Project.

### 3. Results

The following part we collect data from both systematic literature review and interview which aim to find the benefits of pair programming and how many factors influence the efficiency of pair programming in education

#### 3.1 Systematic Literatures Review

In systematic literatures review, the data are separated into two parts, which are benefits of pair programming and factors which influence the efficiency of pair programming.

##### 3.1.1 Benefits of Pair Programming

There are a lot of benefits for implementing pair programming in education which are mentioned in the literatures which are selected. The following are the most significant benefits:

No.	Benefits	Reference
<b>B1</b>	Program with higher quality	<ul style="list-style-type: none"> <li>- Williams &amp; Kessler, 2000 [27]</li> <li>- Lai &amp; Xin, 2011 [12]</li> <li>- Dyb ået al, 2007 [5]</li> <li>- Ramli &amp; Fauzi, 2008 [18]</li> <li>- Nagappan et al, 2003[15]</li> <li>- McDowell et al, 2003 [13]</li> </ul>
<b>B2</b>	Students have higher confidence	<ul style="list-style-type: none"> <li>- McDowell &amp; Werner et al, 2003 [14]</li> <li>- Williams &amp; Kessler, 2000[27]</li> <li>- Lai &amp; Xin, 2011 [12]</li> <li>- Ramli &amp; Fauzi, 2008 [18]</li> </ul>
<b>B3</b>	Students enjoy programming	<ul style="list-style-type: none"> <li>- McDowell &amp; Werner et al, 2003 [14]</li> <li>- Cliburn, 2003 [4]</li> <li>- Williams &amp; Kessler, 2000 [27]</li> <li>- Hanks, 2006 [7]</li> </ul>

<b>B4</b>	Students have higher average exam score	<ul style="list-style-type: none"> <li>- Cliburn, 2003 [4]</li> <li>- Lai &amp; Xin, 2011 [12]</li> <li>- Ramli &amp; Fauzi, 2008 [18]</li> <li>- Nagappan et al, 2003[15]</li> <li>- McDowell et al [13]</li> </ul>
<b>B5</b>	Students have higher percentage to go to continue the course	- McDowell & Werner et al, 2003 [14]
<b>B6</b>	Students deliver their products on time/spent less time on task	<ul style="list-style-type: none"> <li>- Williams &amp; Kessler 2000 [27]</li> <li>- Dyb ået al, 2007 [5]</li> </ul>
<b>B7</b>	Reduce workloads for the teaching staff	<ul style="list-style-type: none"> <li>- Williams &amp; Kessler, 2000 [27]</li> <li>- Cliburn, 2003 [4]</li> </ul>
<b>B8</b>	More efficiency when changing request	<ul style="list-style-type: none"> <li>- Xu &amp; Chen, 2005 [30]</li> <li>- Xu &amp; Rajlich, 2005 [31]</li> </ul>

### **B1. Program with higher quality**

According to Williams & Kessler [27], software product with high quality can be measured by the followed three points 1) the product is what they want, 2) they get it when they want it, 3) the product is defect-free. Based on their conclusion, "pair-pressure" seemed to have a positive effect on each of these and all projects were of very high quality. Toll III et al. [24] had interviewed with paired students which considered about the pair-pressure, they said "We both felt that we couldn't let our partner down so we stayed on task more that if we had been working alone". In addition, Xu & Rajlich [31], indicated that the programs written by pairs have more meaningful variable names, which proves that with the help from the partner, programmer pairs are able to write higher quality code.

### **B2. Students have higher confidence**

McDowell et al. [14] suggest that students who paired reported significantly higher confidence in their program solutions than students who worked independently. Moreover Williams & Kessler [27] report that the collaboration made students confident in their work - giving them a "We nailed that one!" feeling. This sentiment made them feel more positive about the class overall. In addition, Lai & Xin [12] conclude from the data they analyzed that the majority of the pair students accept the view that the pair programming is beneficial for improving their self-confidence.

### **B3. Students enjoy programming**

Another benefit which is mentioned a lot by several authors is that the paired students enjoy the programming process. According to Cliburn's [4] interview, there was evidence to show that most students enjoy programming in pairs more than programming alone. What's more, Williams & Kessler [27] had an anonymous survey; it shows that 84% of the class agreed with the statement "I enjoyed doing the assignments more because of pair programming".

### **B4. Students have higher average exam score**

Pair programming improves the student's average exam score which based on Cliburn's [4] interview, most students thought pair programming improved their grade (61.5%), while 30.8% thought there was no change. Students most commonly cited having a partner to help them catch mistakes and debug programs as the primary factor in making their grades better.

Furthermore, according to Lai & Xin [12], experiment, the pair programming students' average score was greater than the non-pair programming students' average score.

#### **B5. Students have higher percentage to go to continue the course**

McDowell et al. [14] propose that a significantly higher percentage of the students who had paired had gone on to attempt the subsequent programming course than had the non-pairing students.

#### **B6. Students deliver their products on time/spent less time on task**

Regarding the benefit, students deliver their products on time, Dybå et al. [5] gave the positive analysis on this. According to their meta-analysis, it suggested that pair programming reduced the time to deliver the finished product, compared with individual programming.

#### **B7. Reduce workloads for the teaching staff**

Pair programming also has benefit for teachers. As Williams & Kessler [27] mentioned, collaboration makes the instructor feel more positive about the class. Their students are happier and the assignments are handed in on-time and are of higher quality. There is one additional very positive effect for the teaching staff -- less questions! According to the Cliburn's [4] interview, pair programming also reduced workload for the teaching staff.

#### **B8. More efficiency when changing request**

The last common benefit in the table above is that pair programming is very efficient when students face changing requirements. This can be proved by Xu & Chen [30]. According to their experiment, the results indicated that the pairing can reduce the time used during software evolution, especially for moderate programs. Furthermore, they suggest that the knowledge accumulation for the pair is much faster than that of individuals in the course of carrying on change requests.

### **3.1.2 Factors which Influence Pair Programming**

However, pair programming is not always as efficient as those literatures said. Lots of benefits of implementing pair programming in classroom are influenced by one or more specific factors. After reviewing the selected literature, 4 common factors can be found which are listed below:

No.	Factors	Reference
F1	Partner choose	- Toll III et al, 2007 [24] - Thomas et al, 2003 [23] - Sennett & Sherriff, 2010 [20] - Braught et al, 2008 [1] - Jacobson & Schaefer, 2008 [9] - Katira et al, 2004 [10] - Hannay et al, 2010 [8]
F2	Task complexity/size	- Sison, 2009 [22] - Dybå et al, 2007 [5] - Giri & Dewangan, 2012 [6]
F3	Efforts in supervision	- Williams et al, 2011 [2] - Jacobson & Schaefer, 2008 [9]

		- Bevan et al, 2002 [3] - Williams, 2007 [26]
<b>F4</b>	Gender	- Braught et al, 2008[1] -McDowell et al, 2003 [14]

### **F1. Partner choose**

How to choose a perfect partner is the most important factor which would influence the efficiency of implementing pair programming in the classroom. According to the literatures, there are 3 common ways to choose partners. They are:

- Assigned partner by teacher

#### a. Students are paired depends on their program skill levels

Toll III et al. [24] got the result according to their experiment, the best learning environment was created when one partner had slightly more, or slightly less skill than the other. However, when the difference in skill is too great, the pair programming ideal can break down. In addition, Braught et al. [1] also had an experiment, they paired students by similar assignment scores in the course and the result indicated that students with lower SAT scores who pair-programmed in lab tended to perform better on individual programming tasks than students with similar SAT scores who programmed individually in lab.

#### b. Students are paired depends on their confidence toward program

Thomas et al. [23] did two experiments which are opposite and similar experience toward students. In the first experience, they paired code-warrior (“I have had no trouble at all completing programming tasks to date, I love to program and anticipate no difficulty with this course”) to the “opposites” (Code-a-phobe, “I don’t like programming and I don’t think I am any good at it”) in terms of the attitude questionnaire and get the result that only 53% of the warriors reported enjoyment of the experience and only 47% of them thought that pair programming led to a better solution. The second experience, they paired two students with “the same”. 60% of the phobes said they enjoyed this experience more and the warriors had equal number reported liking it more and less than the previous time.

#### c. Students are paired depends on their personality

Personality and learning style are proved that had less significant correlations towards the efficiency of pair programming, the evidence can be found in Sennett & Sherriff [20]. Furthermore, Hannay et al. [8] reported on a study of the impact of the Big Five personality traits on the performance of pair programmers together with the impact of expertise and task complexity through three countries forming 98 pairs. They also got the conclusion that the personality do not affects pair programming.

- Choose partner randomly

According to the experiment of Katira et al. [10], it suggests that pairs will be highly compatible and successful if we pair them randomly, without necessarily considering personality type, skill level, or self-esteem.

- Choose partner by students themselves

Jacobson & Schaefer [9] indicated that encouraging students to choose their own partners results in a very high rate of compatible pairings. Katira et al. [10] reported that students

prefer to pair with someone they perceive to be of similar technical competence. Williams et al. [28] got the similar result based on their experiment - pairs are more compatible if students with similar perceived skill level are grouped together.

### **F2. Task complexity/size**

Another factor which affects the efficiency of pair programming is the task complexity. Sison [22] did two experiment and the result indicate that pair programming can increase software quality when the software being built is relatively complex and on the contrary, pair programming can decrease programmer productivity when writing programs that are relatively simple. In addition, Dybå et al. [5] give the best answer on whether two heads are better than one - it depends. They highly recommend that junior programmers need use pair programming whenever the task is complexity or not. Also, senior programmer do not need paired with someone unless the task is too complex to be solved satisfactorily by an individual senior programmer.

### **F3. Efforts in supervision**

Hanks [7] got an unexpected result of this study which is student attitudes toward pair programming may be influenced by the instructor. So that the efforts of TA are also an important factor which influence the efficiency of pair programming. For example, Jacobson & Schaefer [9] present that courses should use closed labs, scheduled times and each has a teaching assistant and a peer lab tutor present. The TA is polite but adamant that switching occurs.

### **F4. Gender**

McDowell et al, 2003 B/24 reported that the gender of the partner was not related to performance on the final exam. And according to Hanks' [7] analysis on student attitudes toward pair programming, it indicates that women tended to have more positive attitudes toward pair programming than men did. Since there are only two literature study gender as their primary work, and then indicate that gender is not influence the efficiency of pair programming at all. Some other literature, however, report that gender may the factor that does matter. We cannot give the most appropriate suggestions on the gender factor, and we will not take this factor into our final result.

## **3.2 Interview**

Interview 5 students which have implemented pair programming in their studies.

<b>Students</b>	<b>Summary</b>	<b>Related to Findings</b>
<b>S1</b>	<ul style="list-style-type: none"> <li>● Gives correct definition of pair programming and knows it by book and internet, no training before implementing it</li> <li>● Dislikes pair programming because observer may focus on other staff</li> <li>● The benefits of pair programming depends on the size of project</li> <li>● Assigned partner by group manager</li> <li>● Felt bad at first time when using pair programming and got better when he was familiar with pair programming</li> </ul>	<p><b>F1</b> <b>F2</b> <b>F3</b></p>

S2	<ul style="list-style-type: none"> <li>● Enjoy pair programming depends on the project</li> <li>● Chosen partner by himself</li> <li>● Pair programming is not bad</li> </ul>	<b>F1</b> <b>B3</b>
S3	<ul style="list-style-type: none"> <li>● Feels “pain” because he must always keep track on the partner</li> <li>● Pair programming is good mostly, sometimes they may have different opinions but finally come up with something good</li> <li>● No training of pair programming and knows pair programming based on software process course</li> <li>● Choose his partner himself with similar program skill</li> <li>● Partner’s feedback is helpful</li> <li>● Recommend to using pair programming at university level</li> <li>● <b>Suggestion:</b> make sure that everything is clear between the people involved and everybody is on the same page</li> </ul>	<b>F1</b> <b>F3</b>
S4	<ul style="list-style-type: none"> <li>● Gives incorrect definition of pair programming (divided task)</li> <li>● Choose his partner himself</li> <li>● His program skill is better than his partner</li> <li>● Have schedule conflict</li> <li>● Enjoy pair programming process</li> <li>● <b>Suggestion:</b> it’s better that two programmer complement each other and dislike the partner with the similar compatibility</li> </ul>	<b>F1</b> <b>F3</b> <b>B3</b>
S5	<ul style="list-style-type: none"> <li>● Gives correct definition of pair programming</li> <li>● No training of pair programming and most trainings were targeting agile or scrum in software process course</li> <li>● They were a team of 5 students and ended up into two pairs and one single guy</li> <li>● She chosen the one with similar programming capabilities and she enjoyed it</li> <li>● Benefits: became more calm and confident, saved time, higher up code quality</li> <li>● worst moment when they had completely different directions</li> <li>● Schedule isn’t really a big problem</li> <li>● Recommend pair programming if both developers are of roughly the same programming levels</li> </ul>	<b>F1</b> <b>F3</b> <b>B1</b> <b>B2</b> <b>B3</b> <b>B6</b>

## 4. Discussion

### 4.1 Full Pair Programming Process

Based on the result of systematic literature review and interview, we conclude that a *full pair programming process* includes three parts: preparation stage, implementation stage and

grading stage. Literatures [4, 27] indicate that pair programming reduces the workload for teaching staff. This is one benefit we found in literature review. However, we believe that if the instructors want to implement a *full pair programming process*, from prepare stage and process monitor to the grade marking, it requests more works for instructors. The effort in supervision is one important factor which influencing the efficiency of pair programming. *Full pair programming process* is listed below and shown in Figure 1:

### **Prepare Stage:**

Provide good environment: instructors should provide a lab for students to do pair programming [4, 25, 29]. A lab containing computers with double monitors, mice, and keyboards is recommended. In this case, student can easily access to the hardware.

Provide sufficient training: training for students is very important. Instructors should introduce pair programming to students and make sure they understand how to make it work. This can be seen by the fact that one of the students in our interviews misunderstood the definition of pair programming. He thought pair programming is two person share one task. So he just divided the tasks and did solo programming. This student heard about pair programming in a course. But he did not check the explanation of it and used it in the wrong way.

Provide pro-active help with challenges: a complete training includes not only introduce the definition to students, instructors should also give students some early warning for conflict schedule problem (if the instructors implement pair programming after class), unreliable partner and so on. They should give suggestions for the issues which students may meet during the course. Pair with schedule conflict partner or unreliable partner will reduce the benefits of implementing pair programming in classroom. For instance, student who pair with an unreliable partner may feel less enjoyment or they cannot deliver their products on time. In additional, instructors should introduce the benefits of pair programming and encourage students to use it at the beginning of the course. The warm up training will improve the confidence of students who take pair programming in their study.

Align Task complexity with pair programming: the tasks or assignments of pair programming should be well-designed. We found that the complexity/size of task is one factor which influencing the efficiency of pair programming in education [5, 6, 22]. Two interviewees also point out that the benefits of pair programming depend on the size of project. For students with different programming levels (junior or senior), for students with different learning experiences (first time of using pair programming or experienced with pair programming); instructors should design various assignments or projects.

### **Implementation Stage:**

When implementing Pair programming, instructors are released from basic programming questions because students are paired together and they can complement each other. For this point, some literatures indicate one benefit: reduce workload for the teaching staff [4, 27]. Instant of answering some basic simple questions, instructors should observe and visit the students, ensure they are working together effectively. They should: a) Encourage students to communicate which improves the knowledge transfer. b) Let them understand that talking is not cheating which help them transform solo pair programming to collaborative learning. c) Help students display humility and understand the respect between partners is significant [27].

### Grading Stage:

In earlier time, many teachers and administrators view collaboration as an unwieldy grading problem, and prefer to avoid it entirely [3]. But time makes the marking system of pair programming mature. Even in pair programming learning process, an individual test is essential [17]. Collaborative learning research indicates that individual accountability occurs when all team members take individual tests and receive individual grades. The individual test should encourage students to stay actively involved in project development. For example, the individual test should require students to develop code which relates to the pair assignments. And the test scores for individuals should be more heavily weighted (individual scores should take sixty percent of the course grade, and the result of pair programming assignments should take forty percent). Research also introduces the peer evaluation to complement assignment grading part [10, 20, 28]. The right marking system of pair programming may reduce cheating in pair programming assignments.

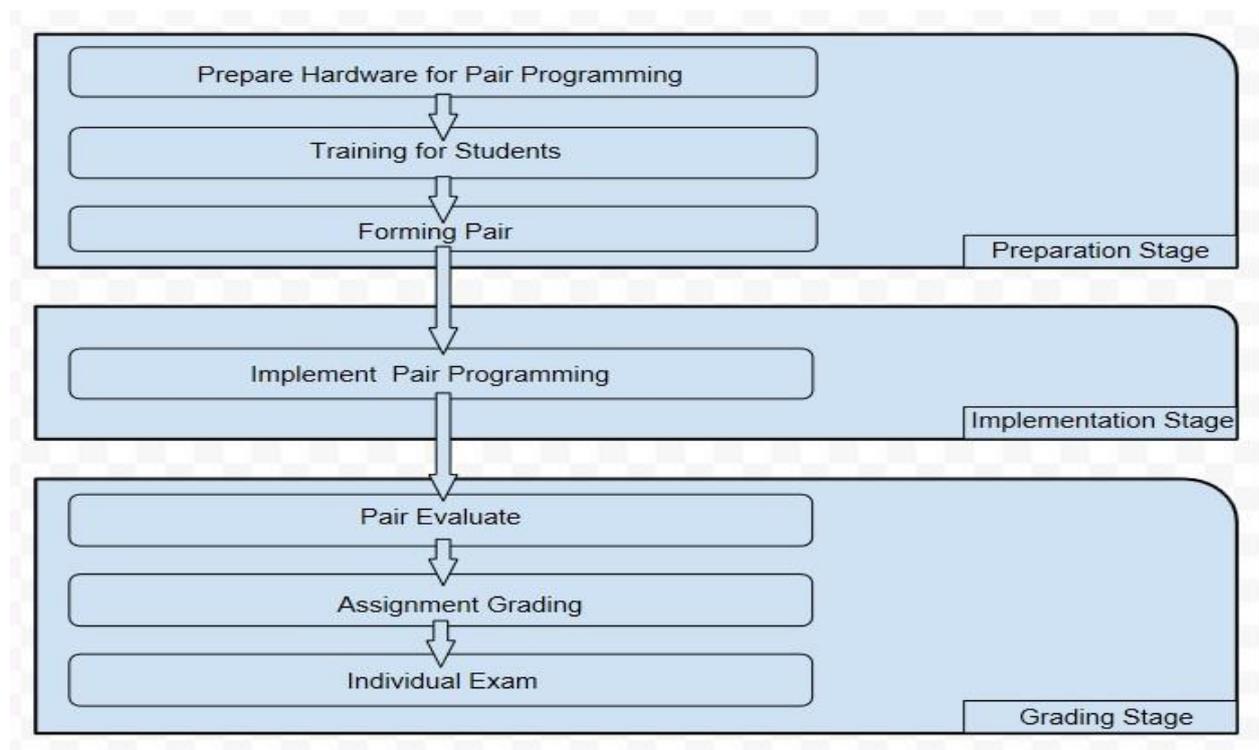


Figure1: *full pair programming process* flow chart.

### 4.2 Obstacle and Efforts in Supervision

Successful implementation of pair programming in education depends on supervision effort. In practice, real education environments may hinder instructors to implement a *full pair programming process*, e.g. through the following behavior: assignment is homework and instructors do not provide a lab or class hour for pair programmers so that students meet and pair program without control. This lack of control increases the likelihood to encounter one challenge obstacle which called schedule conflict occurred [3, 7, 14, 31]. It is hard for some students to meet after class because of conflicting class and work schedules or family obligations [7].

In real education environments, instructors may find it difficult to deal with this challenge when implementing pair programming in their class. Related work indicates that schedule conflicts can seriously impair effectively of pair programming in education. As Williams [26]

mentioned undergraduate students should have extensive experience (probably two full courses) with pair programming in a closed laboratory setting before they can realistically pair on their own outside of a class. Thus, students should pair during class hours [3]. Instructors should implement pair programming in class hours, offer a lab and supervision, and also introduce policies for attendance and tardiness as well [29]. During the pair programming process they can identify the unreliable partners and take steps to cut loss.

If pair programming is after class, schedule conflict problem is really hard to predict or avoid. Before pair programming process really start, in the training part of pair programming, the students should be told that under the circumstances which the two partners have insurmountable scheduling conflicts, students should alert the instructor as soon as possible. Then the instructor should prepare for re-pairing. In Bevan's experiment [3], some pairs tried to overcome scheduling conflicts for several weeks before reporting the problem, which made re-pairing much more difficult.

If re-pairing is difficult, the precarious pair still can be saved by distributed pair programming. Distributed pair programming allows students who had difficulty with collocation to still enjoy the benefits associated with pair programming. Hanks [7] had conducted a study and indicated that students in pair programming groups and in distributed pair programming groups receive similar scores on their assignments and examinations, have similar levels of confidence, and express similar attitudes toward pair programming. Note, that Hank's study offers only limited quantitative evidence that indicates the benefits of implementing distributed pair programming in education. This study recommends students sit and work together and propose distributed pair programming as an alternative choice when students have schedule conflict sometimes and find it hard to re-pair.

The instructor should allow the students to do solo programming learning as the last resort. Sometimes instructors should encourage the partners who are facing break-ups to “divorce” to avoid cheating in assignment summit and unhappy pair programming experience. Bevan [3] had found something surprising that the willingness of students to submit an assignment with both partners' names attached, even if one partner had not contributed at all. Pair programming is just a method for learning. Instructors recommend this method to students and at the same time they should offer a comfortable learning environment as well. The students should know that the final goal in a classroom is learning, if pair programming is not such benefit for some students, they can do solo programming as well.

### 4.3 How to Form Pair

There are a lot of studies that focus on the forming pair factor. And we propose *partner choose* as one important factor which influencing the efficiency of pair programming in result part. Basically, this research offers three methods for forming pair: assigned by teacher, choose partner randomly and choose partners by students themselves.

#### **Assigned by Teacher**

Instructors assigning two students as a pair according to one or more factors, for instance: different personality type, actual skill levels, perceived skill level, self-esteem, learning style, work ethic, time management skill [10, 20, 21, 23, 24, 28].

Ranking of the number of studies with corroborating findings relating to pairing formation, the *perceived skill level* was ranked highest, followed by the *actual skill level* [10, 20, 28].

Perceived skill level means the student's perception of their partner's technical competence which differs with their partner's real competence. This result is different with Salleh [21] while they proposed that the actual skill level was ranked highest. Anecdotally, students indicate that they like best to work with students of similar or higher skill level relative to themselves [28]. Actually students like best to work with the one who they believed that he or she has similar or higher skill relative to themselves. Research proves that pairs are more compatible if students with similar perceived skill levels are grouped together [10, 20, 28]. But it is very hard to predict the student's perceptions of each other's competency based on available objective information. So the actual skill level is the most significant reference factor to improve the compatibility between partners.

### **Choose Partner Randomly**

Based on examining the compatibility of student pair programmers, Williams indicates that pairs will be highly compatible and successful even if teachers pair them randomly without necessarily considering personality type, skill level, self-esteem, work ethic, or time management skills [10, 28]. This is good news for teaching staff. Apparently, randomly pairing reduces the workload and does not reduce the effectiveness of pair programming.

### **Students choose partners**

Jacobson [9] have learned that encouraging students to choose their own partners results in a very high rate of compatible pairings. After providing guidance, instructors should offer a buffering time for students to find "the one", maybe two weeks for students to get to know each other in the lab. This approach takes virtually no instructor time, minimal teaching assistant time. The results of interviews also prove that students are more likely to enjoy working with the partner they choose

We suggest "short pair programming" method in order to choose a compatible partners. Instructors assign several small tasks to students and allow them to change their partners every assignment during the first two weeks. Students try to pair with different partners and find out the one they prefer to work with which normally student would choose the one who they believe that he or she has the similar or higher programming skill. It is one way to predict the student's perceptions of each other's competency which is the ranked highest factor that affect the pairing formation.

If the instructor cannot implement a *full pair programming process* which means they implement pair programming after class hour and not provide a lab, they may face the schedule conflicting as mentioned in part 4.2. By students choosing their partners themselves, they can find a partner who has less schedule conflict during the first two weeks before official partner settling down.

Additionally, Williams [29] indicate that a small segment of students will always desire to work alone. Most often, these are the top students who do not want to be "slowed down" by another student and who do not see benefit in teaching others. Letting students choose their own partners will help improve the willingness of using pair programming. They can choose a partner of equal strength to avoid being slowed down. But if instructors want to get to the root of the problem, i.e. how to encourage students accept pair programming, they need help students to know that making the transition from solo programming to pair programming involves breaking down some personal barriers [27]. Communication and sharing is the ladder of the human progress.

## 5. Conclusions & Future work

Based on our results, we argue that if instructors implement the *full pair programming process* as suggested in this report, the efficiency and quality of programming study will be quite improved. Furthermore, if the pair programming is implemented after class, adequate preparation can still make sure it works well.

Pair programming changes the individual activity in traditional software development into collaborative work [12]. It is a bridge from individual programming to team collaboration and there is strong evidence that students have no difficulty working independently after having worked in a pair programming team [9]. It is a smart tool which helping the explorer open up wasteland of programming.

Based on our findings we suggest five future works: a) there should be more experiments about workload reducing. Quantitative data about workload reducing should complement the existing qualitative results reported in literature. b) Pair programming may help reducing the effects caused by the difference of teachers. More research should take into devote. This is a significant factor for reducing educational gap. c) Additionally, an in-depth empirical study should investigate the optimal complexity and size of tasks for pair programming in education. d) The perspective of students on the grading system of pair programming needs to be taken into account, e.g. based on surveys or interviews. e) Two students share one machine may reduce the requirements of hardware. This maybe one benefits of implementing pair programming especially in developing countries.

## Reference List

- [1] Braught, G., Eby, L. M., & Wahls, T., 2008. *The Effects of Pair-Programming on Individual Programming Skill*. ACM SIGCSE Bulletin, 40(1).New York, NY, USA.P: 200-204
- [2] Braught, G., & Wahls, T., 2011. *The Case for Pair Programming in the Computer science Classroom*.ACM Transactions on Computing Education, 11(1). New York, NY, US.
- [3] Bevan, J., Werner, L., & McDowell, C., 2002. *Guidelines for the Use of Pair Programming in a Freshman Programming Class*. Software Engineering Education and Training. Covington, KY. P: 100-107
- [4] Cliburn, D. C., 2003. *Experiences With Programming at A Small College*. Journal of Computing Sciences in Colleges, 19(1), Consortium for Computing Sciences in Colleges. USA. P: 20-29
- [5] Dyb å T., Arisholm, E., et al, 2007. *Are Two Heads Better than One? On the Effectiveness of Pair Programming*. Software, IEEE, 24(6). P: 12-15
- [6] Giri, M., & Dewangan, M., 2012. *A Study of Pair Programming in the context of Facilitating the Team Building*. Advanced Computing & Communication Technologies. Rohtak, Haryana. P: 20-23

- [7] Hanks, B., 2006. *Student Attitudes toward Pair Programming*. ACM SIGCSE Bulletin, 38(3). New York, NY, USA, P: 113-117
- [8] Hannay, J. E., Arisholm, E., et al, 2010. *Effects of Personality on Pair Programming*. Software Engineering, IEEE Transactions, 36(1). P: 61-80
- [9] Jacobson, N., & Schaefer, S. K., 2008. *Pair Programming in CS1: Overcoming Objections to Its Adoption*. ACM SIGCSE Bulletin, 40(2). New York, NY, USA, P: 93-96
- [10] Katira, N., Williams, L., et al, 2004. *On Understanding Compatibility of Student Pair Programmers*. ACM SIGCSE Bulletin, 36(1). New York, NY, USA, P: 7-11
- [11] Kitchenham, B., 2007. *Procedures for Performing Systematic Reviews*. EBSE Technical Report version 2.3, Univ. of Durham
- [12] Lai, H., & Xin, W. J., 2011. *An experimental research of the pair programming in java programming course*. e-Education, Entertainment and e-Management. Bali, P: 257-260
- [13] McDowell, C., Hanks, B., & Werner, L., 2003. *Experimenting with Pair Programming in the Classroom*. ACM SIGCSE Bulletin, 35(3). New York, NY, USA, P: 60-64
- [14] McDowell, C., Werner, L., et al, 2003. *The Impact of Pair Programming on Student Performance, Perception and Persistence*. Software Engineering. 25th International Conference. P: 602-607
- [15] Nagappan, N., Williams, L., et al, 2003. *Improving the CS1 Experience with Pair Programming*. ACM SIGCSE Bulletin, 35(1). New York, NY, USA, P: 359-362
- [16] Preston, D., 2005. *Pair Programming As A Model of Collaborative Learning: A Review of the Research*. Journal of Computing Sciences in Colleges, 20(4). Consortium for Computing Sciences in Colleges, USA, P: 39-45
- [17] Preston, D., 2006. *Using Collaborative Learning Research to Enhance Pair Programming Pedagogy*. 3(1), ACM SIGITE Newsletter, 3(1). New York, NY, USA, P: 16-21
- [18] Ramli, N., & Fauzi, S. S. M., 2008. *The Effects of Pair Programming in Programming Language Subject*. Information Technology. International Symposium. Kuala Lumpur, P: 1-4
- [19] Simon, B., 2008. *First-Year Students' Impression of Pair Programming in CS1*. Journal on Educational Resources in Computing, 7(4), Article No. 5. New York, NY, USA
- [20] Sennett, J., & Sherriff, M., 2010. *Compatibility of Partnered Students in Computer Science Education*. SIGCES'10 Proceedings of the 41st ACM technical symposium on Computer science education. New York, NY, USA, P: 244-248
- [21] Salleh, N., Mendes, E., et al, 2011. *Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review*. 37(4), July/August, IEEE Computer Society: 509- 525

- [22] Sison, R., 2009. *Investigating the Effect of Pair Programming and Software Size on Software Quality and Programmer Productivity*. Software Engineering, IEEE Transactions, 37(4). P: 187-193
- [23] Thomas, L., Ratcliffe, M., & Robertson, A., 2003. *Code Warriors and Code-a-Phobes: A Study in Attitude and Pair Programming*. ACM SIGCSE Bulletin, 35(1). New York, NY, USA, P: 363-367
- [24] Toll III, T. V., Lee, R., & Ahlswede, T., 2007. *Evaluating the Usefulness of Pair Programming in a Classroom Setting*. Computing and Information Science, 6th IEEE/ACIS International Conference. Melbourne, Qld, P:302-308
- [25] Weiss, R. S., 1994. *Learning from Strangers: The Art and Method of Qualitative Interview Studies*.
- [26] Williams, L., 2007. *Lessons Learned from Seven Years of Pair Programming at North Carolina State University*. ACM SIGCSE Bulletin, 39(4). New York, NY, USA, P: 79-83
- [27] Williams, L., & Kessler, R. R., 2000. *The Effects of "Pair-Pressure" and "Pair-Learning" on Software Engineering Education*. Software Engineering Education & amp. P: 59-65
- [28] Williams, L., Layman, L., et al, 2006. *Examining the Compatibility of Student Pair Programmers*. Agile Conference
- [29] Williams, L., McCrickard, D. S., et al, 2008. *Eleven Guidelines for Implementing Pair Programming in the Classroom*. Agile Conference. Toronto, ON, P: 445 - 452
- [30] Xu, S. C., & Chen, X. H., 2005. *Pair Programming in Software Evolution*. Electrical and Computer Engineering. Saskatoon, Sask, P: 1846-1849
- [31] Xu, S. C & R, V., 2005. *Pair Programming in Graduate Software Engineering Course Projects*.6/05, IEEE Frontiers in Education Conference: F1G-7-F1G-12