Effective Pair Programming Practice- An Experimental Study

Venkata Vinod Kumar Padmanabhuni, Hari Praveen Tadiparthi,
Muralidhar Yanamadala, Sagar Madina
Master of Science in Software Engineering, Blekinge Tekniska Hogskola, Sweden.
Email: (hata10, muya10, samc09)@student.bth.se

ABSTRACT
Pair programming is a practice, where two programmers work on a single task. Long research has been carried out in industries, and efficient results were achieved. Research is also being done in implementing the pair programming techniques in academics and good results are seen. In this paper, we suggested a new approach of pairing students from different engineering levels an experiment was conducted on a sample of students. In this research, both qualitative and mixed approaches are used.

Keywords: pair-programming, PP, experiment.

1. INTRODUCTION
Pair programming is an eXtreme Programming (XP) practice and an agile software development method [1]. Pair programming is a programming technique where two programmers work on same computer on a single task. One programmer is designated as the driver while the other is a navigator. This is different from collaborative programming, in a pair programming the driver works on the computer while the navigator checks for errors, thinks of alternatives and other resources [2]. The two programmers switch their responsibility periodically [3].

Many empirical studies and research show that pair programming is an efficient technique used in the development of software projects to improve the software quality. In software industry pairs produce code with low defect rates; development time is short and is cost efficient [4]. Research suggests that pair programming has increased the students’ self-confidence and interest in the field of Information Technology [5].

Pair programming helps in knowledge transfer among the student and emphasizes the benefits of the pair designing [6]. Pair programming creates an environment in encouraging the students to learn actively, leading students more confident and developing social interaction [7].

Research has been done in pair programming with students of different skills. The results showed that the best learning environment occurs when one person is slightly more or less skilled. Here the students participated are of same educational background and are undergraduate students. No research study was done by pairing persons with different engineering levels like undergraduate, post graduate and PhD.

2. BACKGROUND AND RELATED WORK
A lot of research on pair programming has been conducted in the field of software engineering, which showed better results when compared to individual programming in Industrial context. Tacit knowledge was shared among the employees, which helps them to increase their learning skills. Previous studies related to pair programming in the field of academics helped both the teaching faculty and students in obtaining good results.

2.1. Literature review
Pair Programming has a great impact in industries and students education. In Software Industry, it has increased the productivity and knowledge transfer between employees. The advantages of pair programming motivated industries and higher educational institution students in the fields of Computer Science (CS) and Software Engineering (SE) to implement this practice.

Pair programmers spent 60 more hours of time in programming to complete the task but after the adjustment period, i.e. pair jelling it was found that compared to solo students paired students completed their task 40% to 50% faster, and many of the students reported that they enjoyed the experience of using pair programming [8].

Data Analysis is done based on the research question, and it was proven that compatibility factors influence the students when they program in pairs. There are three studies who specifically addressed the compatibility of pair programmers [9][10][11]. The results are varied among the compatibility of different groups, students from different courses with different personalities and same self-esteem and same skill are paired [11].

Another three studies [12][13][14] also concentrated on the personality type, Myers-Briggs personality test (MBTT) was the most commonly used instrument to measure the personality type. NEO-PI is also used to measure the personality type [14]. It was discovered that personality of a single programmer does not affect the results of pair programmers but there is a difference when pairs of different personalities work together.

There are also articles that discuss about the compatible factors affect the pair programming [15], [16], [17], [18]. There was a conflict in the idea of pair programming, the two members have to divide the work and at the end
integrate their results but the concept is different where two programmers work on the same task, and their jobs are different.

The advantages of pair programming over individual programming are pairs produce code with fewer defects, produces good quality of code within half the time of an individual programmer, develop good knowledge, builds team work and finally the rotation of partners helps in knowing the overall system [20].

Pairing of students is a great challenge; only the factor of academics is considered by leaving gender and race. The factors should be considered properly. There are also many guidelines to be followed while pairing the students in order to achieve high effectiveness. The students should be supervised by staff or instructors in order to resolve the problem between the pairs. Partners of the students should also be rotated so they can develop communication skills and prevent them from intolerance of same partner [21].

Experience of pair programming shows good results for the Information Systems’ students [19]. Industries programmers will come from different educational backgrounds, i.e. people not only come from computer science but also from economics, arts, etc [6]. In these aspect Authors E.Bellini et al. conducted an experiment by pairing students of Master of Technologies of Software and Master of Management and Technologies of Software which had shown better results of knowledge transfer between them [6]. Hence, we intend to do research in this field of pair programming by not only implementing on the students of same degree but also extending this concept to pairing students of different degree levels, which may show a great impact in the students’ thesis or projects.

3. RESEARCH QUESTIONS

Based on the findings of our literature review and other related work we formulated two research questions for this research paper.

**RQ1:** How the effectiveness of pair programming increases by pairing students of different education levels?

**RQ2:** What are the benefits and challenges of pairing students?

4. RESEARCH METHODOLOGY

Research is the systematic way to collect data, analysis and find a solution to the problem [22]. We are following mixed method approach [28] based upon the research questions we have formulated. The RQ1 is answered by using interviews and questionnaire at end of experiment. The RQ2 is answered by analyzing the experiment results.

### 4.1. Qualitative Research

#### 4.1.1. Context

To understand the benefits and challenges of pair programming we opted for a qualitative procedure interview. We performed this interview process with ten professors who are having interest and experience of research in pair programming. The professors are from Norway, Germany, Sweden and Austria. Among the selected sample professors are related to both academia and industry since they also work as consultants for software companies.

#### 4.1.2. Data Collection

The data collection is done through interviews. We have communicated with professors through communication tools like Skype, Instant messenger and mails. We thought this Interview is useful for collecting the information related to pair programming. Such as benefits and challenges of pair programming. The interview Questionnaire is given in Appendix A.

#### 4.1.3. Data Analysis

The data collected from professors is analyzed by tabulation of a questionnaire survey and narrative analysis. The list of benefits and challenges collected from literature review [5][7][19][25][26][2][30][31][32][33][34][35] and interviewers are shown in Table 1:

<table>
<thead>
<tr>
<th>Benefits</th>
<th>No. of professors Supported</th>
<th>No. of Professors not supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Productivity increases</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>Knowledge transfer</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Increase in communication skill</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Problem solving skill</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>Quality increases</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>PP is enjoyable (work satisfaction)</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Builds trust and increase team work</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Increase in student Academic performance</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>New coding idioms</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Challenges</th>
<th>No. of professors Supported</th>
<th>No. of Professors not supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deciding on pairs</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>Assigning the complexity tasks</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>Bonding between team members</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>
Motivating the students continuously towards the goal

| 4 | 6 |

4.2. Quantitative method

For this quantitative method, an experiment was conducted among the students of different levels, and their results are compared. Claes Wholin defined experiment as “A formal, rigorous and a controlled investigation and the experimentation is a process of systematic, disciplined, quantifiable and controlled way of evaluating the human based activities” [39].

4.2.1. Purpose of the study

The research methods are selected based on the research questions. The quantitative method is used to show the effectiveness of pair programming over individual programming and pair programming among the students of same degree to the students of different educational levels.

4.2.2. Experiment Planning

The experiment was planned to be conducted in our home university Blekinge Teknik Hogskola, Sweden. A total of 45 students were taken from Computer Science and Information Technology branches. Among them 15 members from under graduation (UG), 15 members from post-graduation (PG) and other 15 are from PhD. For this experiment, a Java program is considered as it was a single subject of the course in the UG level that was known to the other two levels (PG and PhD).

The objects needed for conducting the experiment are a class room with good infrastructure, eight computers, and a tutor to guide them all. The interview is conducted after the completion of an experiment so printed material with questions is needed. The experiment is conducted on a Java program. The computers should be provided with Java IDE tools.

4.2.3. Hypothesis

There are two hypothesis null hypothesis and alternative hypothesis.

- Null Hypothesis: This defines that there is no effect in the measurement we have considered.

  \( H_0 \): Effectiveness of pair programming doesn’t increase by pairing students from different level.

- Alternative Hypothesis: This converse to the null hypothesis and to prove that the observations considered are to be true.

  \( H_1 \): Effectiveness of pair programming increases by pairing students from different level.

4.2.4. Variable Selection

For conducting an experiment the variables are considered. There are two types of variables independent variables and dependent variables. The variable selection is as follows:

- **Independent Variable**: It includes the same or different personalities of students, no. of students, time, and infrastructure.

- **Dependent Variable**: It includes pair performance, pair participation, code design and communication among the students.

4.2.5. Experiment Design

In experiment design effectiveness was measured. Generally, the related concepts for measuring effectiveness are project duration, Effort and Quality [22]. Since the experiment is designed for a developing a program code the project duration will be the time taken for completing the task. The measure of effort is the person-hours required to complete the task. The quality depends on how good the code was developed and the measure is number of test cases passed. Two experiments were conducted on students’. First Experiment is concerned about the effectiveness of pair programming to individual programming, and second experiment is about the effectiveness of pair programming of students of same level to pairing the students of different level.

Firstly, process metrics like pair speed advantage and defect density are used for measuring the effectiveness of pair programming compared to solo programming [23]. First Experiment was conducted on graduate students. Their time of completions of individual and pair programming were recorded as shown in Table 2.

### Table 2: Time duration of first experiment

<table>
<thead>
<tr>
<th>Programming Type</th>
<th>Time Duration (in person hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual</td>
<td>2.58</td>
</tr>
<tr>
<td>Pair</td>
<td>2</td>
</tr>
</tbody>
</table>

It was noticed that pair programmers on average require 29 percent less time than individual programmers.

**Pair speed advantage** = \( \frac{100}{100-29} = 1.4 \)

Hence pair speed advantage was accepted as it lies in the range between 1.3 to 1.8 [24][25][26]. According to W.Humphery good conventional software eliminates upto 70 percent defects and Defect Density was assumed to be 0.03 defects per line of code. Defect Density of the first experiment was shown in Table 3.

**Defect Density** = Lines of code / defects
Finally, from the results of process metrics pair speed advantage and defect density in first experiment shows that pair programming is more effective than individual programming.

Secondly program duration, quality and effort are considered in second experiment to examine the pair programming effectiveness of students from same level to the students from different level. The students are formed pairs among them in all three levels and while pairing the students of same level they all are of same skill and it does not show much influence on the program effectiveness. The time duration taken by the students to complete the program, the no defects found which indicates the quality and no. of test cases the program passed are tabulated in Table 4. According to our requirements, the program developed by the students has to satisfy 15 test cases and basing on this programming effectiveness of students was represented using the graphs as shown in Figure 1.

hence programming effectiveness based on test cases for UG-UG is 55%, PG-PG 65% and PhD – PhD is 95%. Now the students of different level are paired like under graduate – post graduate (UG-PG), under graduate – PhD (UG-PhD), PhD – post graduate (PhD-PG). To this mixed personality pair a Java program is given, and their program efficiency is measured and are tabulated in Table 5. Similarly as above the program effectiveness of mixed personality pairs can be seen in Figure 2.

**Table 3: Defect Density measurements**

<table>
<thead>
<tr>
<th>Programming type</th>
<th>Lines of code</th>
<th>Defects</th>
<th>Defect Density</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual</td>
<td>120</td>
<td>6</td>
<td>0.05 defect per LOC</td>
</tr>
<tr>
<td>Pair</td>
<td>90</td>
<td>3</td>
<td>0.03 defect per LOC</td>
</tr>
</tbody>
</table>

**Table 4: Measured values of students paired in same level**

<table>
<thead>
<tr>
<th>Pair level</th>
<th>Person hours</th>
<th>No. of defects</th>
<th>No. test cases passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>UG-UG</td>
<td>13</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>PG-PG</td>
<td>10.5</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>PhD-PhD</td>
<td>8.25</td>
<td>4</td>
<td>14</td>
</tr>
</tbody>
</table>

**Table 5: Measured values of students paired in different level**

<table>
<thead>
<tr>
<th>Pair level</th>
<th>Person hours</th>
<th>No. defects</th>
<th>No. test cases passed</th>
</tr>
</thead>
<tbody>
<tr>
<td>UG-PhD</td>
<td>9.5</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>PG-PhD</td>
<td>8</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>PG-UG</td>
<td>11</td>
<td>9</td>
<td>10</td>
</tr>
</tbody>
</table>

**Figure 1:** Programming Effectiveness = (no. test cases passed / (based on test cases) expected test cases to be Satisfied) * 100
programming. The Questionnaire sends to students is in Appendix.

4.2.6. Data Analysis

4.2.6.1. Hypothesis Testing

To reject a null hypothesis we use two-way ANOVA test to test the hypothesis [37] in section 4.2.3. Firstly, students’ feedback is analyzed and the benefits were compared to the benefits drawn in qualitative analysis and are shown in Table 6. For our convenience assume pairs:

A: PG-PG, B: PhD-PhD, C: PG-PhD
D: UG-PhD, E: UG-UG, F: UG-PG

Table 6: Benefits of PP supported by student pairs and professors

<table>
<thead>
<tr>
<th>Benefits</th>
<th>No. of professors Supported</th>
<th>Pairs supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Productivity increases</td>
<td>9</td>
<td>A,B,C,D,E,F</td>
</tr>
<tr>
<td>Knowledge transfer</td>
<td>6</td>
<td>A,B,C,D,F</td>
</tr>
<tr>
<td>Increase in communication skill</td>
<td>7</td>
<td>A,B,C,D</td>
</tr>
<tr>
<td>Problem solving skill</td>
<td>8</td>
<td>B,C,D</td>
</tr>
<tr>
<td>Quality increases</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>PP is enjoyable (work satisfaction)</td>
<td>10</td>
<td>A,B,C,D,E,F</td>
</tr>
<tr>
<td>Builds trust and increase team work</td>
<td>5</td>
<td>B,C,D</td>
</tr>
<tr>
<td>Increase in student Academic performance</td>
<td>6</td>
<td>A,B,C,D</td>
</tr>
<tr>
<td>New coding idioms</td>
<td>7</td>
<td>C,D</td>
</tr>
</tbody>
</table>

Table 6 signifies that the pair of different education levels has experienced more benefits than other pairs.

5. THREATS TO VALIDITY

5.1. Internal Validity

Maturation: The students participated in the experiment may lack concentration or enthusiasm and this might be a threat to experiment result. To prevent this threat experiment is executed with only interested participants.

Instrumentation: The instruments used in experiment like computers can affect the results of an experiment if the systems are not in good working condition, while measuring the time taken by programmers to complete the task. To avoid this threat instructor are advised to check the systems [19].

Selection: Selection of a particular programming language like Java in this experiment could affect the results. As some participants may not have a basic idea of programming in Java, and they may take a lot of time to understand. To minimize this threat prerequisite for the student is to be of academic background of computer science/software engineering and who has completed the basic course of Java in their academics.

5.2. External Validity

Selection-treatment interaction: The experiment is conducted manually and no tools are used and also there is no industrial environment. For result validation this can be a threat.

5.3. Construct Validity

Hypothesis guessing: In an experiment if test subjects are not properly instructed to the students then the outcome of an experiment is affected and the students probably guess the hypothesis and false assumptions may deviate from the main aim. Hence, to avoid this threat the students are given 20 minutes brief instructions about what we want to do them in the experiment [36].

Investigator bias: Information from professors may be indirect and is concluded through the perspective of interviewee’s views and also they have answered the questions with some designation but not from natural setting. This threat is to be handled by preparing the questions easily understandable to both interviewer and interviewee and can be easily answered by an interviewer.

5.4. Conclusion Validity

Heterogeneity of participants: The limitation of an interview with professors is that people are not equally articulate and perceptive [28]. The selection of professors from different expertise level and area of research may be a threat. To minimize this threat, we used selection criteria for professors whose area of interest is in pair programming.

Reliability measure: The answers given by interviewers are descriptive. This is a threat while we are analyzing the interview questions. To minimize this threat it is better to provide an option for answers on an ordinal scale which will be easy for measurement purpose.
6. CONCLUSION AND FUTURE WORK

In this research, we finally conclude that pair programming is an efficient technique in programming. While pairing the students from different levels benefits the students in sharing their knowledge, develop communication skill, low level students can gain more knowledge from the higher level students, and also they learn much better than by reading books.

As Researchers suggest that running experiments with students reduce the validity of study and more advantage than an industrial environment (Kitchenman 2002). Hence future work we recommend is to use this pair programming practice of different levels in an educational environment to an industrial environment. We suggest in industries the pairing should be done among the senior programmer and a fresher so that the fresher can easily learn the new technologies with just an initial idea about the technology. Industries can also cut off their costs involved in training the fresher and also the cost required to fix errors and debug them.

7. ACKNOWLEDGEMENT

We would like to thank Dr. Mikael Svahnberg, Jan Kasper Martinsen, Michael Unterkalmsteiner, Criag Lindley and Kranthi Kumar Lakum for spending their valuable time in discussing the ideas and reviewing our research paper.

REFERENCES


APPENDIX A

Questionnaire for Professors
Q1. What is your designation?
Q2. How many years you are working on the field of pair programming?
Q3. What are the benefits of pair programming?
Q4. What are challenges of pair programming?
Q5. Which programming practice is best solo or pair programming?
Q6. What is your future work on pair programming?

Questionnaire for Students
Q1. How was the co-operation with your team member?
Q2. How was the communication with your team member?
Q3. How is Quality of the program by doing in pairs?
Q4. What is your satisfaction level of knowledge sharing in pair programming?

Q5. How is the work environment in pair programming?

Q6. What benefits you observed while doing pair programming?

Q7. What challenges do you face while working in pairs?

APPENDIX B

Data Entry: ANOVA

<table>
<thead>
<tr>
<th>Group</th>
<th>Data Entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1 2 3 4 5 6</td>
</tr>
<tr>
<td>B</td>
<td>7 8 9 10 11</td>
</tr>
<tr>
<td>C</td>
<td>12 13 14 15 16</td>
</tr>
<tr>
<td>D</td>
<td>17 18 19 20 21</td>
</tr>
<tr>
<td>E</td>
<td>22 23 24 25 26</td>
</tr>
</tbody>
</table>

ANOVA: Results

The results of a ANOVA statistical test performed at 12:21 on 17-MAR-2012

<table>
<thead>
<tr>
<th>Source of Variation</th>
<th>Sum of Squares</th>
<th>d.f.</th>
<th>Mean Square</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between</td>
<td>6.906</td>
<td>5</td>
<td>1.381</td>
<td>0.042</td>
</tr>
<tr>
<td>Error</td>
<td>131.4</td>
<td>12</td>
<td>10.95</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>140.3</td>
<td>17</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The probability of this result, assuming the null hypothesis, is 0.972

Group A: Number of items=3

Mean = 7.07
95% confidence interval for Mean: 5.814 thru 8.328
Standard Deviation = 2.52
Hi = 8.00 Low = 6.00
Median = 8.00
Average Absolute Deviation from Median = 0.833

Group B: Number of items=3

Mean = 9.17
95% confidence interval for Mean: 7.746 thru 10.592
Standard Deviation = 1.66
Hi = 10.50 Low = 8.00
Median = 9.00
Average Absolute Deviation from Median = 0.833

Group C: Number of items=3

Mean = 8.75
95% confidence interval for Mean: 7.554 thru 9.958
Standard Deviation = 2.02
Hi = 10.00 Low = 7.00
Median = 9.00
Average Absolute Deviation from Median = 1.667

Group D: Number of items=3

Mean = 9.50
95% confidence interval for Mean: 8.333 thru 10.667
Standard Deviation = 2.90
Hi = 12.00 Low = 7.00
Median = 9.50
Average Absolute Deviation from Median = 1.667

Group E: Number of items=3

Mean = 6.67
95% confidence interval for Mean: 5.404 thru 7.938
Standard Deviation = 2.03
Hi = 8.00 Low = 4.00
Median = 7.00
Average Absolute Deviation from Median = 3.33

Group F: Number of items=3

Mean = 10.0
95% confidence interval for Mean: 8.833 thru 11.167
Standard Deviation = 2.00
Hi = 11.00 Low = 9.00
Median = 10.0
Average Absolute Deviation from Median = 0.667
AUTHORS BIOGRAPHY

- Venkata Vinod Kumar Padmanabhuni
  Masters of Science in Software Engineering
  Blekinge Institute of Technology-Sweden

- Hari Praveen Tadiparthi
  Masters of Science in Software Engineering
  Blekinge Institute of Technology-Sweden

- Muralidhar Yanamadala
  Masters of Science in Software Engineering
  Blekinge Institute of Technology-Sweden

- Sagar Madina
  Masters of Science in Software Engineering
  Blekinge Institute of Technology-Sweden