

## **Pair Programming “Potential Benefits and Threats”**

**Supriya Gupta<sup>1</sup>, Vandana Bhattacharya<sup>2</sup>, Madhumita Singha<sup>3</sup>**

Department of Computer Science - St. Xavier's College, Ranchi<sup>1</sup>

Department of Computer Science - Birla Institute of Technology, Lalpur<sup>2</sup>

Department of Information Technology Xavier Institute of Social Service, Ranchi<sup>3</sup>

### **Abstract**

*Agile software methodologies are preferred over traditional methods in a variety of industry projects. Pair programming is a key practice of Extreme Programming; an agile software development technique, in which two students/ programmers/ co-workers work together at one terminal. One, the writer, writes code while the other, the analyst, reviews each line of code as it is typed in. The two co-workers swap roles frequently. For the IT students who are the future IT professionals of a country, Pair Programming may be a good method of training, as they need to inculcate the necessary soft skills that are essential in software development besides the necessary technical background. Students need to learn how to work in a team, one of the key aspects of sustenance in an industry. This helps them to develop the art of communication and interaction. This indeed makes them progress towards the path of making themselves better IT managers. Experiencing team work i.e. working in tandem with team members is an essential preparation for professional software development. In this process of learning the Mentor plays a vital role. Mentor doesn't only guide the students towards the completion of the project or assignment but also helps the students in their mental conditioning. The objective of this paper is to give a comparative study of a few authors, where the challenges faced by the students and how Pair Programming or collaborative learning can benefit and pose threats to students are explored.*

### **Keywords**

*Agile Methods, Pair Programming, Unified modeling language, learning , self-organizing team.*

### **1. Introduction**

Pair programming [1] is an intensive style of programming co-operation where two people working together in resolving programming scenarios. Educators are interested in this style of teaching approach as it enhances learning experience

and improves programming performance. In most of the universities and colleges teaching computers at graduate or at masters level typically introduce students to the social and managerial side of the computer science discipline. Projects are shared i.e. students share not only the responsibility but also their grades [2].The concept of collaboration and team spirit is very essential for professional software development.

Agile software development represents a new approach for creating platform and controlling software projects. Agile manifesto [3], a statement that expresses a set of basic principles and rules:

- (1) individuals and interactions over processes and tools;
  - (2) working software over comprehensive documentation;
  - (3) customer collaboration over contract negotiation;
  - (4) Responding to change over following a plan.
- Professors/Mentors have to actively participate in students projects/assignments, this enables creating a more professional environment for students. Mentors have to give importance to the impact of collaborative assignments and understand how students perceive teamwork which is an essential element for professional success. Pair programming a key element of XP (Extreme programming) is a way where students of a team are allowed to work on one computer for coding, thereby encouraging students to learn collaborative aspect of a team. In this paper, we seek to understand what are the challenges faced by the students of computer science discipline, how pair programming and group work helps students to move towards more professional culture in an organization. Mentors guidance to students will have positive effects on students, as mentors will not only guide strong coders but also students who are not so confident about the coding aspect of the project thereby reducing students negative experiences.

This paper is methodized as follows:

- Section 2 : Challenges faced by Students
- Section 3 : Introduction to Pair programming and Related work.
- Section 4 : Methodology of Pair Programming.
- Section 5 : Potential Benefits of Pair programming.
- Section 6 : Threats of Pair Programming.
- Section 7 : Future works and Conclusion.

## **2. Challenges faced by Students**

Undergraduate students face lot of challenges, they are not in position to correctly estimate the workload of their project or assignment [4] i.e. they lack vision due to lack of self-confidence. Students don't take design phase too seriously or they are less willing to expand beyond the immediate requirements. They don't even concentrate on understanding the specifications; they don't even use UML (Unified Modeling Language) to detail their blue-print phase. Students just try to document the phase without any elaborated diagrams. Moving ahead with this halfhearted approach to the coding phase they don't get the code right and then students don't have the courage to discard the code and start again, i.e. their less willingness to accept change. They don't even effectively use standard testing framework. In this process students develop slackness and casualness in their attitude.

## **3. Introduction to Pair Programing and Related Works**

This section provides an introduction to pair programming, and various related empirical studies.

### **3.1 Pair Programming**

Pair programming is a practice, whereby two programmers work side by side at the same computer, continuously collaborating on the same design, algorithm, code, or test[5,1].When students working in pair feel responsible for their partner's success or failure. The two members of the team using the same computer for coding where one member acts as the driver who is actually the coder and the other member who is looking at the screen is the navigator or we can say the visionary focuses on fault finding and problem. Two students sharing the same computer communicate almost every minute which gives them better understanding of the problem and the courage to move in right direction.

### **3.2 Related work**

Lucas Layman [1] has discussed pair programming with respect to students in one of his paper.

According to him an increase in the organization of the students' and pairing those up will be beneficial when collaborating on complex tasks in other classes or in a professional career. Collaborative assignments make students save time and help them to get better insight of the project.

Nils Brede Moe, TorgeirDingsoyr and Tore Dyba [6, 7, and 8] have reviewed in their paper; a team following a plan driven model often consists of independently focused self-managing professionals, who resist change. A transition from high individual and low group autonomy to a high level of individual and group autonomy is probably the biggest challenge when introducing change-driven development based on self-organizing teams. Software development process depends significantly on team work, as does any process that involves human interaction. Two important channels to achieve team performance are assessment and dissemination.

HamedYaghoubiShahir, ShervinDaneshpajouh, and Raman Ramsin[9] also have suggested that pair programming makes two people responsible for a task and they can cover each other's weaknesses. They suggest that light analysis and design approach which brings agile approach closer to traditional approach. Traditional approach harps on heavy design and documentation which is generally not adopted by the students, they want to quickly get over with design with very little documentation, so this light method design would indeed readily be accepted by the students.

Laurie Williams, D. Scott McCrickard , Lucas Layman , Khaled Hussein suggested [10] certain guidelines for students to work in pair. These guidelines are in the context of the HCIcourse undertaken by 22 students in the Fall 2007. The two guidelines that were added are as follows. The first suggested that it is important for students in a pair to be working for a common goal. The second additional guideline suggests that the mentor should refrain from telling the answers but instead point them in the right direction and encourage them to find answers together.

Pearl Brereton, Mark Turner and RumjitKaur[11] in their paper mentioned studies reports on five aspects of a student w.r.t pair programming , Exam marks/ Assignment marks, Assignment Quality, Pass/success rate, Retention, confidence and enjoyment attitude .They found that there was an effective improvement in the pass rates of undergraduate students, Significant improvement in the retention rates of

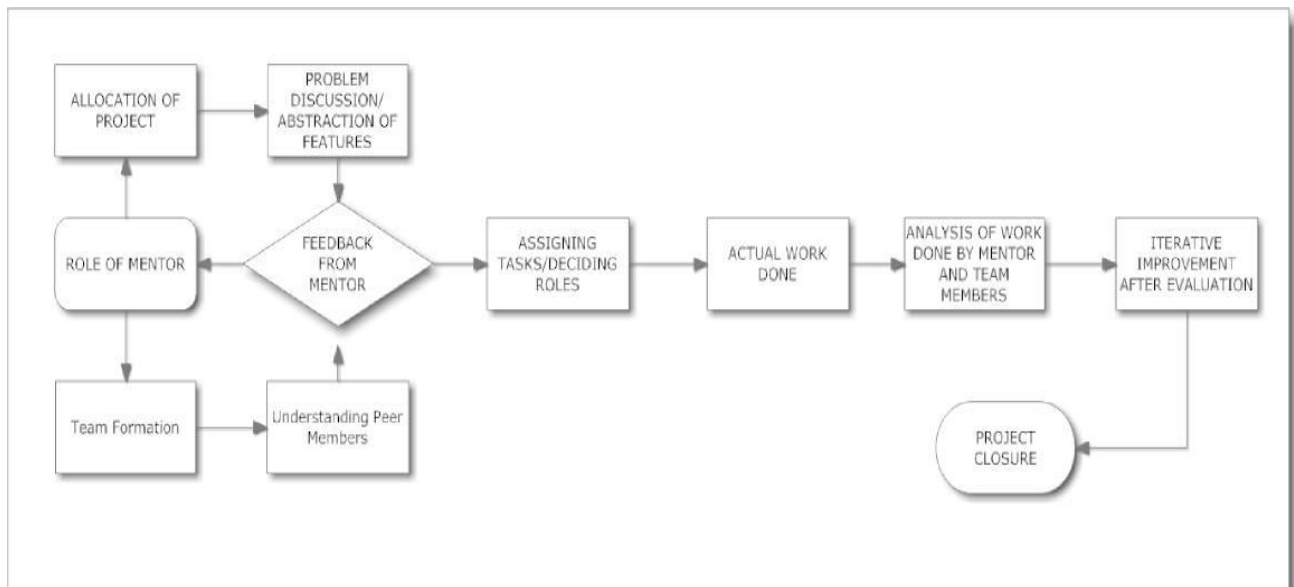
programming modules. There was a great sense of enjoyment and had a better command on the computer language. There was improvement in the quality of work and thereby increasing the level of confidence. All the studies showed positive reports except one study report stated that pair programming had an effect on non-Computer Science majors and had no significant effect on the Computer Science Majors.

#### **4. Methodology of Pair Programming**

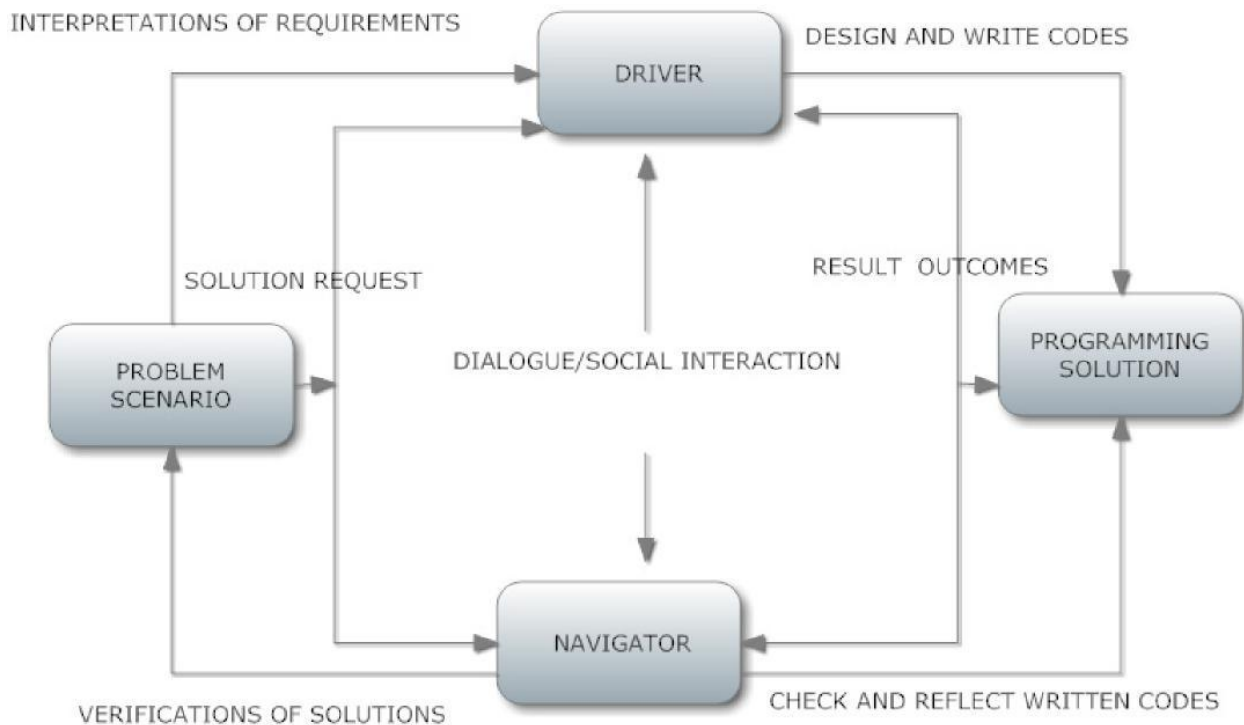
Steps in the process of pair programming with mentor in the scenario. (FIG 1)

1. Mentor begins the process by forming pairs according to student's personality types and subsequently allocating the project.
2. After the formation of teams, peer members spend time knowing each other and simultaneously abstracting the details of the project.
3. If members are not compatible then they
4. consult the Mentor; who then accommodate
5. the individuals in different teams. After

6. contemplation teams get their details assessed by the mentor; or they can ask for a different project if the group members are not complacent with the initial project allocation.
7. After the strenuous discussion on abstraction of the project details, then the team decides roles for each peer member.
8. The team starts the process of actual work i.e. designing, coding etc. Team members discuss with each other frequently, subsequently mentor guides them in the right direction after evaluating their work.
9. If there is scope of iterative improvement then teams are asked by the mentor to refine their work. This helps students to work in future with their peer members in the organization and also build a favorable environment for work thereby experiencing enriched human interaction.
10. After evaluating the project the mentor asks for submission for grading and assessment and the project closes.



**Fig 1: Process Flow of Pair Programming (Mentor's Role Elaborated)**



**Fig 2: Process of flow of Pair Programming Dialog among Team Members  
(Chong, Plummer, Leeifer, Klemmer, Eris, & Toyer, 2005)**

In the pair programming software development paradigm [12], the driver and navigator act on the Specifications in tandem and develop code. The two co-workers swap roles frequently during the assignment. In this alternating dynamic, we postulate that the navigator's focus on higher level conceptual relationships and goals allow him/her to take on a coaching role where he/she observes the driver's interaction with the code, identifies needs and opportunities, and intervenes to supply needed information and/or strategy to arrive at the desired goal, while the driver attends to the immediate coding task at hand. Inclusion of the navigator introduces multiple feedback paths for knowledge creation and error correction. This view is consistent with statement that the navigator is performing tasks with a longer time constant than those of the driver.

## **5. Potential benefits of Pair Programming**

Pair programming is a crucial XP process that Williams and Kessler [13, 14] describe as follows: "Pair programming is an approach of programming in which two programmers /students /co-workers work side by side at one terminal, continually

participating on the same design, algorithm, code, or test. One of the pair, called the driver /writer, is typing at the computer or writing down a design. The other partner, called the navigator /analyst, has many jobs, one of which is to review the work of the driver/ writer, looking for sensitive and crucial bugs. Sensitive bugs are syntax errors, defects, calling the wrong method, and so on. Crucial bugs occur when the driver / writer are directed down the wrong path—what is implemented just won't achieve what needs to be achieved. The navigator / analyst are the strategic, comprehensive thinker... Another great thing is that the driver / writer and the navigator / analyst can enter into a discussion on demand at any time .A competent pair programming relationship is very alive. The driver / writer and the navigator / analyst communicate every minute. It's also very important to swap roles frequently between the driver / writer and the navigator / analyst." When programming in pair, the drivers'/ writers' work is constantly evaluated. This is called "Pair Analysis" [21]. This helps students to debug the problem faster and discussing their work with the peer member makes their tasks simpler. In this entire process the students achieve a higher level of satisfaction and the team members get to know each other better

which in turn helps them to do the task in a much better collaborated way. Students learn to be better team members i.e. they learn to be a part of team and work towards more professional culture.

Working in pair leads to exchange of knowledge and if they have any doubt or problem in their assignment they have the courage to ask to their respective mentor. This is called "Pair Mettle" [15]. Students who are less confident about any programming language tend to learn more from their partners, thereby having a stronger knowledge base. It's a constant learning process, which helps them evolve as a better working professional and have the right sense of soft skills which makes them more appropriate to work in an organization. Strong coders who are generally introverts and don't like working in a team, when paired with right person with proper guidance of the mentor tend to come out of their own shell.

## **6. Threats of Pair Programming**

If the pairing is not right then there is difference of opinions due to which members tend to withdraw from the project and loose interest and focus, so it is very important to do the right pairing of students, thereby increasing the work of the mentor. Firstly the mentor has to understand the students' personality traits and academic soundness of each student then take decision of pairing students. Secondly mentor has to constantly guide them, get effective feedback and then keep evaluating them.

It is a time taking procedure not only for the mentor but also for the students. Mentor spends time understanding his/her students and constantly evaluating them. Students have to expend time on understanding his/her peer members. Students tend to rely on their peer members and tend to deviate from the path of self-learning. Students tend to regress to their old habits i.e. their slack and casual attitude if constant pressure of the mentor is not there.

## **7. Future work and Conclusion**

The study of Pair Programming has emphasized the importance of learning as it helps in learning the basic programming concepts. Pair programming helps students to learn the process of abstracting the details of a project. This method of learning helps students to develop self-reasoning and enhanced programming skills. Students are paired in class room teaching so that there can be interactive learning i.e. by learning-

by-doing paradigm which inculcates in them social interactive skills. It makes them better suited for an organizational culture when they complete their education. Threats posed by pairing up students can be taken as one time investment of time by the concerned faculty as far as pairing is concerned. This will help the faculty/mentor to know his/her students better and in turn conduct a better lecture as he/she is in a better position to take their queries in a positive stride. Finally as teaching and training tool Pair Programming may open up new vistas for active learning with peers which is faster, deeper and unleashing unbound energy and enjoyment of achievement through team work. However, the role of the mentor in this process is vital and cannot be over emphasized.

## **References**

- [1] Tie Hui Hui, Irfan Naufal Umar," Pair Programming and LSs in Computing Education: It's Impact on Students' Performances",2011.
- [2] Lucas Layman ,"Changing Students' Perception: An Analysis of the supplementary Benefits of Collaborative Software Development, North Carolina State University,2006.
- [3] Abrahamsson, P.,Salo, O.,Ronkainen,J., and Warsta,J., Agile software development methods: Review and analysis,2002, VTT Technical report.p 107.
- [4] Peter Maher,"Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum",Department of Mathematics and Computer Science , 2009 Sixth International Conference on Information Technology: New Generations.
- [5] L.Williams and R.Kessler,"Pair Programming Illuminated,Addison Wesley, Reading, Massachusetts,2003.
- [6] Nils Brede Moe, TorgeirDingsoyr , Tore dyba, " Understanding Self Organizing Teams in Agile Software Development",SINTEECT,Norway, 2008.
- [7] Neur,S.andBalijepally, V.,Theoretical reflections on agile development methodologies- The traditional goal of optimization and control is making way for learning and innovation. Communications of the ACM,2007. 50(3): p.79-83.
- [8] Guzzo,R.A and Dickson, M.w., Teams in Organizations :Recent research on performance and effectiveness. Annual Review of Psychology, 1996 ,47 :p 307-338.
- [9] HamedYaghoubiShahir, ShervinDaneshpajouh, and Raman Ramsin, "Improvement Strategies for Agile Processes: A swot Analysis Approach", Department of Computer Engineering Sharif University of Technology Tehran, Iran, 2008.

- [10] Laurie Williams , D. Scott McCrickard , Lucas Layman , Khaled Hussein, “Eleven Guidelines for Implementing pair Programming in the Classroom”, North Carolina State University ,Department of Computer Science,2008.
- [11] Pearl Brereton, Mark Turner, Rumjit Kaur “Pair Programming as a teaching tool: a student review of empirical studies”,school of computing and Mathematics ,Keele University,2009.
- [12] Jan Chong, Robert Plummer, Larry Leifer, Scott R. Klemmer, Ozgur Eris, and George Toye,”Pair Programming: When and Why it Works”, 2005.
- [13] L. Williams and R. Kessler, Pair ProgrammingIlluminated, Addison Wesley, 2002.
- [14] RaymundSison,” Investigating Pair Programming in a Software Engi-neering Course in anAsian Setting”, College ofComputer Studies,De La Salle UniversityManila, Philippines,2008.
- [15] Chih-wei Ho, Kelli Slaten, Laurie Williams and Sarah Berenson,”Work in Progress - Unexpected Student Outcome from Collaborative Agile Software DevelopmentPractices and Paired Programming in a Software Engineering Course”,2004.



My home town is Ranchi; my date of birth is 3rd November 1979. I have done my Bsc (Economics Honors) from St. Xaviers college Kolkata, Masters in Computer Application from Institute Of Man-agement And Technology Ghaziabad, Pursuing final semester in M.Tech(Computer Science) from Birla Institute of technology Mesra. I am currently working as “Assiatant professor ” in St. Xaviers College ,Ranchi