

ABSTRACT

Agile Software Development is being considered as one of the best software development models, in which the family of development processes, discusses the various ways of creating software in a lighter, faster and more people-centric way. It is a fast growing industrial practice due to the documented benefits of increased software quality, shared programmer expertise, and user satisfaction. Extreme Programming (XP) is one of the Agile Methods. Pair Programming (PP) is one of the best practices of XP. PP is a coding activity, in which two programmers collaborate continuously on the program development in a single computer.

Pair programming is practiced in IT industry by software professionals as a style of programming which allows developers to produce superior code in less time, and with very less defects, than code produced by individuals. Reported experiments which obtain quantitative results have typically been performed in an IT industry, and may not reflect actual academic conditions.

Direct or indirect mentoring is the most common training method. This thesis reports on an experimental investigations using pair programming in an academic setup. Its goal is to determine if this pair programming style can be used to increase the technical and programming knowledge of the students, and verify the benefits of pair programming is achieved or not.

Anecdotal and qualitative evidence from the software industry indicates that in pair programming two programmers working side-by-side on one computer, collaborating on the same design, algorithm, code, or test, perform substantially better than the two persons working alone. Due to the nature of pair programming, communication should occur continuously. So,

effective communication is essential between pair for the success of programming.

The first study provides the support to encourage investigations in pair programming, to determine the best time frame to introduce pair programming to the students either in the first half or in the second half of the semester. The experiment was conducted with 60 students in the computer science laboratory to investigate whether better performance and satisfaction outcomes are achieved when the pair programming is performed. In this study, the students are divided into two groups and one of the group's of the course completed programming exercises in pairs, whereas students in the other groups of the course completed programming exercises in solo. This process repeated vice versa between two programming laboratory courses in the same semester. In this investigation, the students were worked in solo and in pairs throughout the semester and completed programming exercises. The experimental data was analyzed based on the various lab tests and final lab exams. This research supports that the adoption of pair programming in the second half better than the first half due to the toughness of the course increases. It also improves the learning ability, reduces time taken to do a laboratory exercise and increases the knowledge and overall programming skills.

The second investigation was conducted towards the effects of pair programming with student context in terms of duration, effort and correctness of the output. It also to check whether the task complexity like low complexity or high complexity of the problem affects the effort differences between solo and pair programming. The experiment has been conducted in the programming course in the laboratory environment. This experiment is also used to analyze whether there is a dependency between the task complexity and students' capability. The results of this experiment show that

there is a significant difference in the duration of task completion, the effort to perform the task correctly between the students involved in solo and pair programming groups.

Third investigation was conducted with students in non-programming courses with high and low complexity of problems. In this work tutorials were conducted in mathematics courses in the classroom environment using pair mode and solo mode towards the effects of pair made in a student context similar to the third chapter experiments of programming courses. This work also checks whether the task complexity affects the effort differences between solo and pair mode. This experiment is also used to analyze whether there is a dependency between the task complexity and students' capability for solving the non-programming courses. The results of this experiment also confirm that there is a significant difference in the duration of task completion, the effort to perform the task correctly between the students involved in solo and pair groups.

The fourth investigation was conducted to experiment the influence of personality differences among paired students using the five-factor model as a personality measurement framework. This is used to improve the effectiveness of Pair Programming as a pedagogical tool for computer science education. The personality measurement framework used in this work was the Five Factor Model, and it focused on the impact of the personality trait 'Conscientiousness' on paired students' academic performance. The students are categorized based on the scores from the personality test for 'Conscientiousness' trait. Based on personality scores they could be paired up with students who are low, medium and high, students are categorized. Pair programming experiments was conducted based on their personality score. When comparing similar and mixed personality pairs, the test scores between these two groups were somewhat same, but mostly a mixed

personality group pairs obtained higher marks than same personality students. The performance of students who engaged in Pair Programming during laboratory classes with those who went pair with same personality level and mixed personality levels were recorded and compared. The grounds for this investigation indicates that irrespective of the variation in students' personality disposition, PP not only caused the increase of satisfaction and trust level, but also brought enjoyment and students' learning motivation.

The focus of this research is to analyze the various factors of pair programming in the academia. The possible benefits include higher scores on graded assignments, program correctness, better code quality, increased learning satisfaction, increased confidence level, reduced student frustration, increased course completion rate. It helps to reduce the workload for the teaching staff indirectly because the students more often look to each other for technical support and advice during their pair programming sessions.

All the experiments were conducted successfully in both solo programming, and pair programming and the result of the experiments satisfy the proposed benefits of pair programming in the academia.