

An Experimental Analysis of pair programming

KulbirKaur*, AnureetKaur**

*PG Dept of Computer Science & Applications, Khalsa College, Amritsar

** PG Dept of Computer Science & Applications, Khalsa College, Amritsar

Abstract:

Pair programming is one of the practices of extreme programming. It's a technique in which two programmers work as a pair together on one workstation. One, the driver, writes code while the other, the observer, pointer or navigator, reviews each line of code as it is typed in. The programmer types the code is called driver and the person review the code is called observer or navigator. Some benefits that can expect from pair programming are: better code, higher morale, better time management, higher productivity and shared knowledge throughout team.

In this paper nine experiments are conducted on students to evaluate three parameters: programming skills, number of errors in the program and completion time of program. Results indicate that programming skills increases and number of errors in the program reduces when students adopt pair programming but in completion time there is little bit variations. The feedback from the students show the positive result that pair programming improves knowledge transfer and enjoyment of work.

Keywords —Pair programming, solo programming, extreme programming, Agile mythology

I. INTRODUCTION

Agile Methodology

Agile software development is a style of software development that emphasizes customer satisfaction through continuous delivery of functional software. It is a group of software development methodology based on iterative and incremental development. It focuses on keeping code simple, testing often and delivering functional bits of application as soon as they are ready. [1]

Extreme Programming

It is one of many agile processes. It is also known as XP. The basic advantage of extreme programming is that whole process is visible and accountable. It is an important new methodology due to two main reasons. First, it re-examine software development methodologies that have become standard operating procedures. Second, it is one of the lightweight methodologies created to help reduce the cost of software development. Extreme programming include programming in pairs or doing extensive code review, unit testing of

all code, avoiding programming of features until they are actually needed, flat management structure, simplicity and clarity in code, expecting changes in the customer's requirements as time passes and the problem in better understood, and frequent communication with the customer and among programmer.[2]

Pair Programming

Pair programming is an agile software development technique in which two programmers work as a pair together on one workstation. One, the driver, writes code while the other, the observer, pointer or navigator, reviews each line of code as it is typed in. All code to be sent into production is created by two people working together at a single computer. Pair programming increases software quality without impacting time to deliver. It is counter intuitive, but 2 people working at a single computer will add as much functionality as two working separately except that it will be much higher in quality. With increased quality comes big savings later in the project. Two programmers are called driver and observer. Both programming

switch roles frequently. One of the programmers (the driver) has control of the keyboard/mouse and actively implements the program. The other programmer (the observer) continuously observes the work of the driver to identify tactical (syntactic, spelling, etc.) defects, and also thinks strategically about the direction of the work. On demand, the two programmers can brainstorm any challenging problem. Because the two programmers periodically switch roles, they work together as equals to develop software. [3][4].

II. PROBLEM DEFINITION

This paper is focused on comparing pair programming with solo programming. Solo programming means where only one student is involved in the development of a program. Pair programming is an agile software development technique in which two programmers work together. Both programmers switch roles frequently. [5]

The objective of this paper is to find out number of errors in the program during both solo and pair programming techniques. Second objective is to compare the completion time of program in both techniques and third and last objective is to compare learning efficiency of students with help of questionnaire which is given before and after both techniques easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

III. EXPERIMENTAL DESIGN

A In the experimental design three different parameters are considered and to evaluate these three parameters three hypotheses are set. To prove these hypotheses eight experiments are conducted.

Three hypotheses considered are given below:

1. Programming skills: Programming skills means individual gained knowledge and it is measured with the help of questionnaire which is given to the students before and after solo and pair programming.

2. Number of errors in the program: Number of errors in the program like syntax error occurred in the program during first compilation and noted down in both solo and pair programming

3. Completion time of program: Completion time is defined as total development time between start and completion of the programming task and also included when program runs correctly. This completion time is noted down in both solo and pair programming.

In order to evaluate these three parameters following hypotheses are set:-

1. The subject knowledge and programming skill increased when students adopt pair programming

2. There is significant increase in correctness to perform task when students adopt pair programming

3. There is significant difference in the duration when students adopt pair programming and solo programming

To prove these hypotheses three different kinds of programs are considered:

1. Simple problem

2. Medium problem

3. Complex problem

In this experiment, more than 100 students from B.Tech 6th semester and M.C.A 4th semester class are considered. Also to prove these hypotheses 8 experiments are conducted. In each experiment number of students, their completion time and number of errors are noted down. After that mean time, standard deviation and average number of errors in both solo and paired group are calculated. On the basis of these calculation graph is designed which easily describe the difference in above mentioned hypotheses.

Choice of Pairs:

1. Expert – expert pairing

2. Average-average pairing

3. Expert-average pairing

According to their class record M.C.A 4th semester as expert class is taken because they are post graduate students and B.Tech 6th semester as average class because they are under graduate students.

IV. RESULTS AND ANALYSIS

SIMPLE PROBLEM

A. Experiment 1

Expert solo programming and expert pair programming:

45 students from M.C.A class are considered. A small introduction about pair programming is given. The students were allowed to program in C and C++. They perform solo programming and pair programming. When they finished the problem it was evaluated immediately and the completion time was noted down. Number of students, syntax errors and their completion time has been noted. Following table show the problem name, number of students, completion time etc for M.C.A class which is considered as expert group.

TABLE 1
DATA FOR EXPERT GROUP DURING SIMPLE PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Expert	Solo Programming	To search element in an array if not present then insert it	27	11
Expert	Pair programming	To merge two array then sort it	32(16) pairs	15 pairs

B. Experiment 2

Average solo programming and Average-average pair programming. In this experiment, 38 students from average group and B.Tech class are taken as average group.

TABLE 2
DATA FOR AVERAGE GROUP DURING SIMPLE PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Average	Solo Programming	To print the following output	16	11

		1 2 3 4 5 6 7 8 9 10		
Average	Pair programming	To print the following output 1 2 3 2 3 4 5 4 3	24(12 pairs)	12 pairs

TABLE 3
COMPLETION TIME AND SYNTAX ERRORS OF BOTH GROUPS

Group	Paired Mean Time	Paired Standard Deviation	Solo Mean Time	Solo Standard Deviation	Average no. of errors in solo programming	Average no. of errors in pair programming
Average	22 min	9.39	15.67 min	6.3	3.78	2.08
Expert	30.66	14.48	31.36	14.09	2.63	0.94

C. Experiment 3

In this part of experiment 26 students from both groups are taken.

TABLE 4
COMPLETION TIME AND SYNTAX ERRORS OF BOTH GROUPS

Group	Technique	Name of program	Total no. of students	No. of students complete the problem	Avg completion time	Avg errors
Expert - average	Pair programming	To reverse a string and check whether its palindrome	26(13 pairs)	9 pairs	55.66 min	4.27

This table shows that when we combine expert & average group together and perform pair programming then mean time, average number of errors and standard deviation are more as compared to pair programming with their own classmates. It means they are not comfortable when paired two different classes together.

LEARNING EFFICIENCY DURING SIMPLE PROBLEM

The learning efficiency of students measured with the help of questionnaire. In which there are total 20 questions and given to the students before and after solo and pair programming. After that their marks were evaluated which is shown in the table below:

TABLE 5
LEARNING EFFICIENCY OF STUDENTS DURING SIMPLE PROBLEM

Group	Number of students	More marks than previous	Less marks than previous	Same marks as previous
Solo1	27	6	8	13
Pair1	32	19	3	10
Solo2	16	5	1	10
Pair2	24	15	4	5

MEDIUM PROBLEM

Now a medium problem is considered as a second experiment. In this part a problem given to the students was more complex than first simple problem.

A. Experiment 1

Average solo & pair programming for medium problem is taken. First 35 students from average group are considered.

TABLE 6
DATA FOR AVERAGE GROUP DURING MEDIUM PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Average	Solo Programming	To search element in an array if	19	12

		not present then insert it		
Average	Pair programming	To merge two arrays then sort it	28(14pairs)	14 pairs

The above table shows that in solo programming only 12 students complete the task where as in pair programming 14 pairs complete the task.

B. Experiment 2

Expert solo & pair programming for medium problem are taken. First 27 students from expert group are considered.

TABLE 7
DATA FOR EXPERT GROUP DURING MEDIUM PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Expert	Solo Programming	Write a function that compare two integer arrays to see whether they are identical	15	11
Expert	Pair programming	To create linked list & print out total no. of elements in it	24(12pairs)	11 pairs

This table 7 shows that in solo programming only 11 students complete the task where as in pair programming 11 pairs complete the task.

TABLE 8
COMPLETION TIME, SYNTAX ERRORS AND STANDARD DEVIATION OF BOTH GROUPS

Group	Paired Mean Time	Paired Standard Deviation	Solo Mean Time	Solo Standard Deviation	Average no. of errors in solo programming	Average no. of errors in pair programming
Average	43.64 min	6.84	56.83 min	5.87	2.89	2.71
Expert	37.10 min	4.74	42.70 min	5.09	3.65	0.85

C. Experiment 3

Expert-average pair programming for medium problem is taken. In this part of experiment 24 students from both groups are considered.

TABLE 9
COMPLETION TIME, SYNTAX ERRORS AND STANDARD DEVIATION OF BOTH GROUPS

Group	Technique	Name of program	Total no. of students	No. of students complete the problem	Avg completion time	Avg errors	Std. deviation
Expert-average	Pair programming	To delete specified node in the list	24(12 pairs)	8 pairs	52.21 min	2.96	7.3

This table 9 shows that when combining expert & average group together and perform pair programming then mean time, average number of errors and standard deviation are more as compared to pair programming with their own classmates. It means they are not comfortable when paired two different classes together.

LEARNING EFFICIENCY DURING MEDIUM PROBLEM

TABLE 10
LEARNING EFFICIENCY OF STUDENTS IN MEDIUM PROBLEM

Group	Number of students	More marks than previous	Less marks than previous	Same marks as previous
Solo1	19	4	2	13
Pair1	28	18	5	5
Solo2	15	1	2	12
Pair2	24	15	5	4

COMPLEX PROBLEM

Now Complex problem is considered as a third part of experiment. In this part of problem is more complex than first two problems given to the students.

A. Experiment 1

Average-solo & pair programming for complex problem is taken. In this 26 students are considered from average group.

TABLE 11
DATA FOR AVERAGE GROUP DURING COMPLEX PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Average	Solo Programming	To show method of overloading	18	7
Average	Pair programming	Given a list of numbers and count number of primes and display it	28(14 pairs)	11 pairs

This table 11 shows that in solo programming only 11 students complete the task where as in pair programming 11 pairs complete the task it means that as problem move towards complex problem from simple and medium one then in both cases less number of students complete the task.

B. Experiment 2

Expert solo & pair programming for complex problem is considered. First 18 students from both the groups are taken.

TABLE 12
DATA FOR EXPERT GROUP DURING COMPLEX PROBLEM

Group	Technique	Name of program	Total no. of Students	No. of students complete the program
Expert	Solo Programming	Write a function that traverse a linear singly list in reverse & write out contents in reverse order	12	7
Expert	Pair programming	Given a two ordered singly linked list & write a function that will merge them into third ordered list	18(9pairs)	5 pairs

This table 12 shows that in solo programming only 7 students complete the task where as in pair programming 5 pairs complete the task.

TABLE 13
COMPLETION TIME, SYNTAX ERRORS AND STANDARD DEVIATION OF BOTH GROUPS

Group	Paired Mean Time	Paired Standard Deviation	Solo Mean Time	Solo Standard Deviation	Average no. of errors in solo programming	Average no. of errors in pair programming
Average	31.09 min	9.03	33.01 min	8.82	2.38	2.04
Expert	44.53	6.38	45.76	7.85	4.38	4.36

V. CONCLUSIONS

In this paper 8 experiments are conducted and the data collected from the experiments has been used to analyses about programming skills, quality of programming, and completion time of program and efficiency of pair programming. The results show

that there is a significant increase in the knowledge when using pair programming. It was also observed that when student are grouped together in pairs the number of errors was very less as compared to solo programming. Completion time of pair programming is fast as compared to solo programming. Learning efficiency of students also increases when they adopt pair programming.

REFERENCES

- [1] www.agile methodology.org
- [2] Khaled R., Barr P., Noble J., Biddle R. System metaphor in extreme programming: A semiotic approach. Presented at The 7th Internat. Workshop Organ. Semiotics (2004) Setúbal, Portugal
- [3] Daniel C. Cliburn, Experiences with pair programming at a small college, Journal of Computing Sciences in Colleges, v.19 n.1, p.20-29, October 2003
- [4] Hahn, J. H., Mentz, E., & Meyer, L. (2009). Assessment strategies for pair programming. Journal of Information Technology Education, 8, 273-284.
- [5] www.indicthreads.com