Parametric Pair Programming - A Way towards Optimum Output

Malay Tripathi[*], Ashish Agrawal[*], L.S.Maurya[**], Dr.Himanshu Hora[***]

[*] Master Of Technology, SRMS CET Bareilly
[**] CS/IT Deptt. ,SRMS CET Bareilly
[***]MCA, SRMS CET Bareilly

Abstract- A new set of software development methodologies has become prevalent in previous years, commonly known as Agile techniques. Pair programming (an activity from agile techniques)- where two programmers work on the same module at one computer; with one programmer takes responsibilities of writing the actual code and the other one takes the responsibility for the monitoring of strategies, designing and defect related issues. The incorporation of pair programming leads to a better software development process resulting in a high quality product. The purpose of this paper is to demonstrate some parameters for building an effective team for performing pair programming.

Index Terms- Agile, Agile manifesto and values, quality, pair- programming, parameters.

I. INTRODUCTION

To build an optimum project within cost and schedule is the ultimate goal of all software development processes and for achieving this many models like waterfall, spiral, CMMI, CMM etc. has been evolved in past and are still evolving. For a high scale result, a methodology has to be a pool of sub-methodologies which helps in to keep focus on various life cycle factors.

In February 2001, 17 software professionals published a manifesto that was based on some light weight processes (iterative and incremental like SCRUM, extreme programming etc.) formally known as "manifesto for agile software development" to define the approach agile software development. Agile methodologies encompasses many techniques like whole team involvement, continuous integration, automated testing etc. and pair programming is one of them which works on the principle *two are better than one.* As we know pair means a group of two persons; in pair programming "driver" and "navigator" makes pair. Where driver do source coding and navigator keep close watch on code being written.

In this paper , we first look at pair programming as defined by Agile Alliance a non-profit organization that promotes software development according to the principles of Agile Manifesto and then we demonstrate some parameters for having effective pair-programming teams.

II. LITERATURE REVIEW

Agile methods are a subset of iterative and evolutionary methods and are based on iterative enhancement and opportunistic development processes. In all iterative products, each iteration is a self-contained, mini-project with activities that Span requirements analysis, design, implementation, and test [1]. A key difference between agile methods and past iterative methods is the length of each iteration. In the past, iterations might have been three or six months long. With agile methods, iteration lengths vary between one to four weeks [1]. The most important technique in Agile methodology is pair-programming. It is arise as an alternative to the traditional programming. As described by Beck in 2000 pair programming "two people working at one machine, with one keyboard and one mouse".

Lui and Chan (Lui 2006) developed a single measurement method Relative Effort Afforded by Pairs (REAP) formula, to measure productivity achieved over a period of time when undergoing a pair-programming experiment [2].

REAP= [(FTP)*2-(FTI) / (FTI)]*100%

Where,
FTP means finish time of pair
FTI means finish time of individual

Another experiment known as repeat programming, in which subjects were asked to write the same program several times, has been conducted to assess the productivity of pair- programming versus solo programming [3].Hitherto pair programming reached far with course of time but no effective parameters are described yet for making good pair for pair-programming. This paper is proposing some parameters for making good pairs.

Erik Arisholm, Member, IEEE, Hans Gallis, Tore Dyba, Member, IEEE Computer Society, and Dag I.K. Sjøberg, Member, IEEE defines in their conceptual model, the effects (in our case given by duration, effort, and correctness of the maintained program) of PP (versus individual programming) will depend on the moderating variables, system complexity and programmer expertise[4].

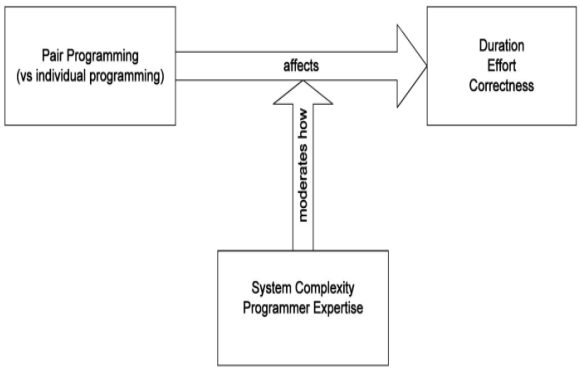

Figure 1: Conceptual research model of the hypothesized pair programming

III. RESEARCH QUESTIONS

After reading many references and papers we found that effectiveness of Pair Programming can still be modified to facilitate agility in any work and to provide highly matured process.

RQ-1:-How the team selection based on some parameters can increases the effectiveness of pair programming?

RQ-2:- What should be the basic parameters for the team selection in pair programming?

IV. RESEARCH METHODOLOGY

IV.I Research design-The research work consisted of an experiment which is described below. The experiment was conducted with 10 under-graduate students. We follow some guidelines given by Jari Vanhanen and Casper Lassenius[5].

IV.II Experiment setting-For conducting experiment, five teams of two students (formed randomly) been identified who did similar computer programs using similar computing language, specifications and tools. A two hours pair programming lecture and presentation was given to the students.

We had 10 students for this experiment all at least 3rd and 4th year B.Tech students at D.N.S. College of engineering and technology, J.P.Nagar affiliated to Uttar Pradesh Technical University. All of them were familiar with software development process, coding phase & practices. All students were used to work in Java technology. Firstly the experiment was conducted normally as pair programming do and then according to our parameters the pairs were selected. The result after that obtained was compared with the previous one.

Likert Scale: - A Likert scale is a psychometric scale commonly involved in research that employs questionnaires. It is the most widely used approach to scaling responses in survey research, such that the term is often used interchangeably with rating scale, or more accurately the Likert-type scale, even though the two are not synonymous.

The scale is named after its inventor, psychologist Rensis Likert. Likert distinguished between a scale proper, which emerges from collective responses to a set of items (usually eight or more), and the format in which responses are scored along a range research design [7].

Table 1. Likert Scale

Rating	Observation
1	Strongly disagree
2	Disagree
3	Neutral
4	Agree
5	Strongly Agree

V. PARAMETERS OF EFFECTIVE PAIR-PROGRAMMING

A. *Team Capability*:-Group of individuals is team, teams make department and on combining departments we form an organization so overall efficiency of an organization depends upon team's efficiency. Now for a good team at least one member should have good experience and knowledge. Among members there should be no ego-problem, no communication gap, and no linguistic barrier. "Even when one programmer is significantly more experienced than the other, it is important to take turns "driving," lest the observer become disjoint, feel out of the loop or unimportant"[6].

B. *Switching flexibility*:-Switching means hopping of members from one team to another. Switching plays vital role in pair programming because it helps in enhancing knowledge of members in different domains, later on which provide an extra-edge to members during promotion and doing quality work in different domain.

C. *Level of compatibility*:-Although understanding and compatibility has different meaning in dictionary but they are much related. If there is no understanding between members compatibility will definitely not exist, so for compatibility, understanding is the only requirement. Everyone has their own thinking and their own ideas, a member should respect ideas of others.

D. *Knowledge about application domain*: - Persons, who are working on any module, should know about all functional and non-functional requirements so that it can easily point out in use-cases that this part is not relevant for this domain. So in one line we can say that persons should have vein of sense about every aspect of the project scope of product and its users, so that they can add functionality, if missed.

E. *No involvement in other lifecycle activities*: - Members should be focused for one work and there should not any condition in which they have to run from one floor to another for completing the tasks. A coder should not be treated as a savior and multi tasker that he/she can play any role you want at a time.

F. No time constraint Vs Time constraint :- For any team, for any programming there must be a time constraint up to which the team has to complete their task so that the team will perform the best quality working hours otherwise they will become the lazy one and will transfer their work from one day to another. It also effect the overall cost, it's a simple logic more working hour means more wages you have to pay to your workers and more you have to wait for result. In the graph below, it is shown that

up to some extent of time the performance is increased but after that increment in time leads to a deviation in performance.

Figure 2 is showing the performance of the persons with time constraint. If there is a much more time given for completing a work then the performance may down. But if there is limited time, work gets completed more fastly.

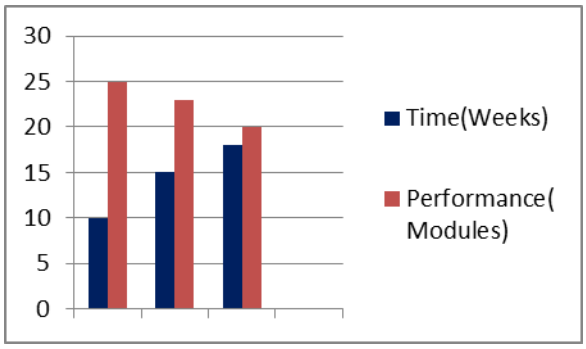

Figure2:Bar Chart for Performance Vs. Time

Table 2. Tabular representation of bar chart

Team	Time(week)	Performance(modules)
A	**10**	**25**
B	**15**	**23**
C	**18**	**20**

VI. RESULTS

After performing pair programming in its conventional manner, we provide a questionnaire based on Likert scale containing our parameters to the students. We request them to give points to our parameters on scale of 1-5.The most promising parameter get the highest points. On the basis of data filled by the students we prepare table (table :5) and calculate average for each parameter. Here, we assume that if the data is above 3,it means that parameter is liked by majority of participants and will provide a good support in pair programming.

After following this data, a pie chart prepared that shows the total percentage distribution for parameters. At the end, it is concluded that, when the performance was evaluated, the performance of the pairs selected on the basis of parameters was better than the performance of the pair selected randomly.

Table 3:Analysis of parameters based on Likert scale

Parameters	Pair1		Pair 2		Pair 3		Pair 4		Pair 5		Average
	M1	M2	M1	M2	M1	M2	M1	M2	M1	M2	
Team Capability	4	4	5	3	2	4	5	4	4	4	3.9
Switching flexibility	2	2	2	3	4	2	4	3	3	4	2.9
Level of compatibility	5	4	4	4	3	2	5	4	5	5	4.1
Knowledge about application domain	5	5	4	5	4	3	3	4	4	5	4.2
No involvement in other lifecycle activities	3	4	2	4	5	3	4	4	3	4	3.6
No time constraint Vs Time constraint	2	1	3	4	1	2	2	2	4	3	2.4

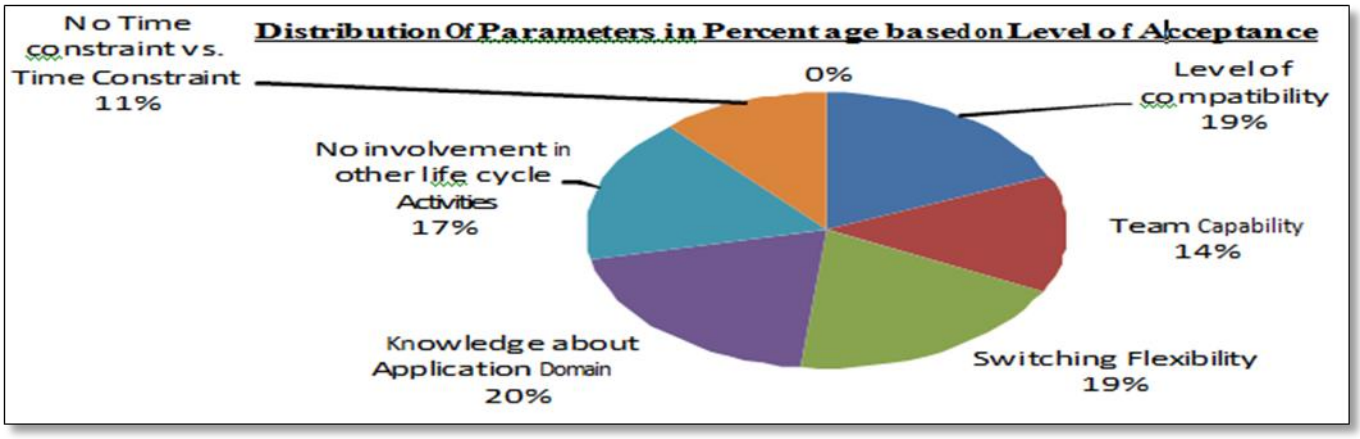

Figure3.Percentage distribution of Parameters

Table 4:Analysis of Performance of various pairs

Pairs	Lines of code	Performance	
		No. of Defects With normal Pair Programming	No. of Defects in Pair Programming with parameters
Pair 1	50	9	5
Pair 2	50	10	8
Pair 3	50	14	12
Pair 4	50	12	9
Pair 5	50	10	6

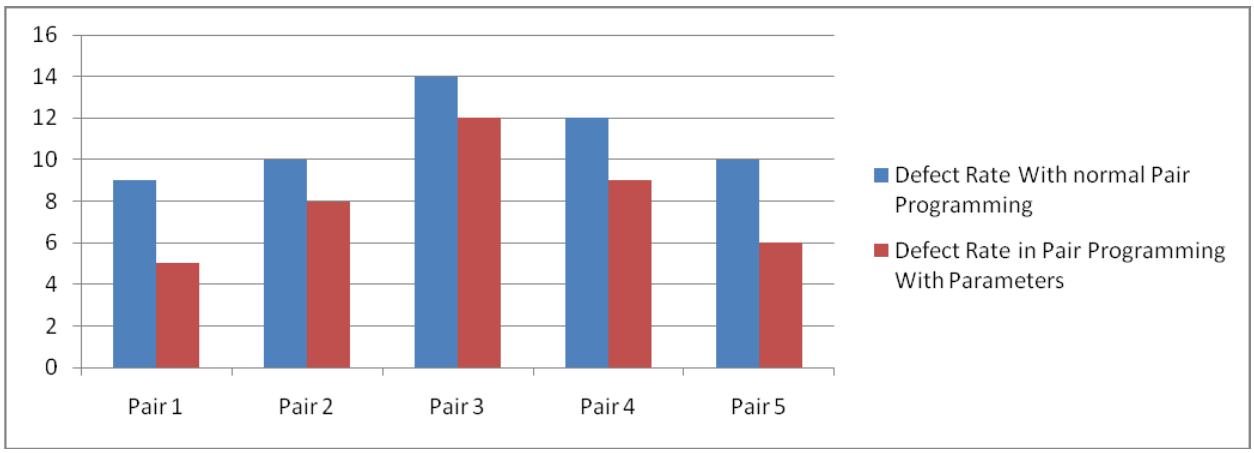

Figure4: Performance based on Defect Rate

VII. CONCLUSIONS AND FUTURE WORK

One may look back on all that has been said about agile much of its common techniques has been known for decades. This is certainly true. But people do not know much about its techniques like pair programming. In this research, we conclude that though pair programming is efficient and one of the best techniques in agile methodology but still it depends on the strength of team or pair indulges in pair programming. If the pair is strengthful and compatible, the output becomes more reliable and efficient. The parameters described here leads to a

better way of selecting pairs and so reduce the cost and time used for generating an optimum output.

Here, in research the whole experiment was conducted with students, hence for future work we recommend is to use these parameters of selecting a pair for industry people who rely on different levels so that the expertise of professionals can be utilized in a better way.

REFERENCES

[1] Laurie Williams, A Survey of Agile Development Methodologies ,2007.

[2] Lui, K.M. And Chan, K.C.C. (2006). "Pair Programming Productivity: Novice-Novice Vs. Expertexpert," International Journal of Human Computer Studies, Vol 64, Pp. 915-925.

[3] Kim Man Lui, Kyle Atikus Barnes, and Keith C.C. Chan, Pair Programming: Issues And Challenges,2010

[4] Erik Arisholm, member, IEEE, Hans Gallis, Tore Dyba , Member, IEEE Computer Society, and Dag I.K. Sjøberg, Member, IEEE , IEEE Transactions On Software Engineering, Vol. 33, No. 2, February 2007

[5] Jari Vanhanen and Casper Lassenius , Effects Of Pair-Programming At The Development [1] K. Beck, Extreme Programming Explained: An EmbraceChange, Addison-Wesley, 2000

[6] E. Arisholm, H. Gallis, T. Dyb°A, and D. I. K. Sjøberg, "Evaluating Pair Programming With Respect To System Complexityand Programmer Expertise," IEEE Transactions On Software Engineering, Vol. 33, No. 2, Pp. 65–86, 2007.

[7] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, and C. A. Visaggio, "Evaluating Performances Of Pair Designing In Industry," Journal Of Systems And Software, Vol. 80, No. 8, Pp. 1317–1327, 2007.

[8] M. M. M¨Uller, "Two Controlled Experiments Concerning The Comparison Of Pair Programming To Peer Review," Journal Of Systems and Software, Vol. 78

AUTHORS

First Author – Malay Tripathi, Research Scholar,Shri Ram Murti Smarak College of Engineering & Technology, Bareilly, malaytripathi406@gmail.com

Second Author – Ashish Agrawal. Research Scholar,Shri Ram Murti Smarak College of Engineering & Technology Bareilly,agarwal.ashish01@gmail.com

Third Author – L.S.Maurya, HOD(CS/IT) ,Shri Ram Murti Smarak College of Engineering & Technology Bareilly,lsmaurya@gmail.com

Fourth Author – Dr.Himanshu Hora ,Associate Professor, Shri Ram Murti Smarak College of Engineering & Technology Bareilly , himanshuhora@gmail.com