Sindh Univ. Res. Jour. (Sci. Ser.) Vol.44 (3) 507-510 (2012)

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Quality Programming Analysis In Pair Programming In Context Of Hetrogenous And Homogenous Pairs

M.R MAREE, F. N. MEMON, A.W.SHAIKH, Z.U.A. KHUHRO, I. A. MEMON*

Institute of Mathematics and Computer Science, University of Sindh, Jamshoro.

Received 12th May 2012 and Revised 21st July 2012

Abstract: Reliable communication is necessary for accomplishment of any task in a better way. For many years programmers in software industry have experienced that by working collaboratively high quality software can be developed in shorter time. Extreme Programming (XP), a lightweight agile software development methodology propagates its twelve interesting practices and stresses to work collaboratively. Pair programming is one of the popular practices of XP, containing the idea of working two programmers collaboratively side by side on design, coding and testing of piece of software. The question arises how the teams of programmers be formalized? Would heterogeneous pairs work better than the homogenous pairs? Or would homogenous pairs work better than the heterogeneous? The existence of these factors can vary in homogenous and heterogeneous pairs. For these reasons and more this paper presents the empirical analysis based upon the evaluation between the homogenous and heterogeneous pairs in terms of efficiency, accuracy, quality, quantity of bugs, brainstorming performance and creativity potential during programming activity.

This paper presents the empirical analysis based upon the evaluation between the homogenous and heterogeneous pairs in terms of efficiency, accuracy, quality, quantity of bugs, brainstorming performance and creativity potential during programming activity.

Keywords: *Extreme programming, pair programming, brainstorming comparison, software engineering.*

1. INTRODUCTION

Introduction of new software development methods have changed the conventional trends. XP is known as the best approach methodology to software development that emphasizes customer satisfaction and teamwork. It is also low risk, flexible, predictable, scientific, and fun way to develop software (Kent, 1999). It has attracted our attention due to its simplicity and is considered today as a sound approach (Radmila 2000). Pair programming is one of its practices in which two programmers with individual personalities (Hans Arisholm, and Dyba, 2003), work collaboratively at one computer on the same design, algorithm, code, or test (Laurie *et al.,* 2003). In this technique of programming one controls the keyboard/mouse and actively implements the program while other, continuously observes the work to identify tactical defects and to provide strategic planning (Kim and Keith 2004). Literature (Hiroshi *et al.,* 2003) (Frank *et al.,* 2003) reported that higher quality software can be developed through this technique in half of the time required for a single programmer. The collaborative working of two brains in pair programming form is beneficial and advantageous over the single programmer. However creativity (Eric *et al.* 2000), brainstorming quality (Jillian *et al.,* 2002) and speed of programming development task can vary in heterogeneous and homogenous pairs (Robert *et al.,* 1997) Brain storming performance and creativity are interrelated, for purpose of empirical analysis we have made it separate entity.

In this paper empirical analysis is presented. The analysis is based upon homogeneous and heterogeneous pairs consisting of three male pairs and three female pairs (homogenous) and three pairs each consisting of one male and one female (heterogeneous), working on the same algorithm and design. The comparative findings in terms of efficiency, accuracy and quality, quantity of bugs, brainstorming performance and creativity potential during programming activity among these pairs.

2. MATERIAL AND METHODS

Efficiency, accuracy and quality

In most of the cases within organizations, tasks are carried out by groups. The reason behind it is that, the tasks may be complex enough to combat. Simultaneously time to market is also critical for software organizations (Hans Arisholm, and Dyba, 2003), Pair programming is advantageous over solo programming as it has a higher development speed (Frank *et al.,* (2003). While the efficiency, accuracy and quality of programming are also important factors which effects generally. The reflection of these factors

++Corresponding Author M.R MAREE mrmaree@yahoo.co.uk, cell. .no. + 92-3332601823

* Institute of Physics, University of Sindh, Jamshoro

may be variable due to gender difference in pairs. For the observation of these factors and making comparison between homogenous and heterogonous pairs, we assigned three types of remarks for the evaluation and analysis excellent, good and fair. We comparatively evaluated efficiency, in terms of getting down to solutions quickly, accuracy in terms of early elimination of defects and quality in terms of higher development speed among the constituted pairs during programming task. Remarks assigned to performance (80-99) excellent, (70- 79) good and (60-69) fair respectively. Empirical assessment results during analysis of each category in average are shown in **(Table 1).**

Table 1. Efficiency, accuracy and quality

	Homogenous (Male)	Homogenous (Female)	Heterogeneous
Quality	70	69	79
Accuracy	65	75	79
Efficienc	70	75	79

Quantity of bugs

In pair programming, programmers perform the programming task continuously, documentation lives in the heads of the programmers (Laurie 2001) .That incorporates the basic concept of design and review which is essential in the development process. There is very limited chance of bugs in pair programming as compared to solo, if bugs occur then the quantity of bugs can vary during programming activity in homogenous and heterogeneous pairs due to personality or gender difference. In the comparative evaluation of quantity of bugs, we have counted the bugs occurrence per one hundred programming statements. The calculated average bugs per each pair category in per hundred statements are shown in **(Table 2).**

Table 2. Quantity of bugs

	Homogenous (Male)	Homogenous (Female)	Heterogeneous
Quantity of bugs per 100 statements	10	8	7

Brainstorming performance

Brainstorming is counted as effective method of problem solving. The quality of ideas and thoughts team creates constitutes an upper limit on the quality of problem solving process (Radmila 2000) Brainstorming mostly depends on chance to achieve the perspective shift (Jillian *et al.,* 2002). In pair programming it is necessary for pairs to communicate about possible approaches in order to discuss expertise during program development. There are two core principles of brainstorming, deferred judgment and quantity breeds quality (Jillian *et al.,* 2002). The quality and quantity of idea generation in terms of heterogeneous and homogenous pairs can be variable due to personality and gender difference, as well as existence of logic in programming activity. Pair brainstorming can be considered as a task of divergent thinking and the ideas generated in such situation can be counted and scored. We considered idea generation in terms of quality and quantity. Appropriateness in terms of approach to programming development as essential factors to compare among the pairs during brainstorming activity. Observed variations in terms of these factors are shown in **(Table 3)**.

Table 3. Brainstorming performance

	Homogenous (Male)	Homogenous (Female)	Heterogeneous
Appropriate-ness	70	70	70
Quality of idea	80	70	80
Quantity of idea	70	80	80

Creativity potential

Creativity is gaining ground in today's fast changing world. Idea formation as well as generation activity always produces new creative ideas. There is widespread evidence available in the knowledge management literature for the importance of creativity and innovation for organizational success in the

changing environment (Aybuke, *et al.,* 2001). Some claim that creative ideas are always new combinations of old ones (Eric *et al.,* 2002). Presently research evolves to utilize the creativity and brainstorming potential with technology support for achieving optimum gains. For these reasons the use of group support systems, computerized group decision support systems and existence of collaborative virtual environments are clear signs to support creativity and potential of geographically dispersed participants is clear sign towards the extraction of creativity and brainstorming potential for optimum gains. In case of pair programming, pairs have not only the speed advantage but they are capable to utilize instant creativity potential jointly for production of ideas in problem solving. The creativity potential may vary in between the homogenous and heterogeneous pairs because each individual pair has own creative potential. In Pair programming pairs exchange their ideas during problem solving, the discussion with each other is obvious to exchange the individual created ideas. The combination of such created ideas again trigger the new idea generating process such cycle continues till the pair reaches to the solution. Due to these reasons and more the comparative empirical analysis in relation to creativity potential in terms of analytical faculty (new ideas / innovations), quality of idea (degree of relieving the symptoms), problem solving (coping with the complexity), logic (reasoning/common sense) the teams generated new ideas with the combination of old, is shown in table 4.

Table 4. Creativity potential

	Homogenous (Male)	Homogenous (Female)	Heterogeneous
Analytical	70	70	70
Quality of Idea	70	70	70
Problem	70	65	70
Logic	65	80	80

3. RESULTS AND DISCUSSION

Homogeneous and heterogeneous pairs were formed for quality analysis. Three male pairs and three female pairs (homogenous) and three pairs each consisting of one male and one female (heterogeneous), total 18 programmers and 5 judges took part in the process. For analysis and observations team of five judges was formalized having expertise in the field. Pairs have been provided separate personal computers to work on same algorithm and design. Remarks assigned to performance excellent (80-99), good (70-79) and (60-69) fair respectively. Observations made based upon *efficiency* (in terms of how teams got down to solutions quickly), *accuracy* (in terms of how teams eliminated defects early) and *quality* (in terms of higher development speed). Where the *quantity of bugs* was calculated average bugs per each pair category in per hundred statements of programme. *Brainstorming* was observed with respect to the idea generation in terms of quality and quantity while appropriateness in terms of approach to programming development was observed. *Creativity* potential was observed in terms of analytical faculty where generation of new ideas/innovations, quality of ideas, problem solving (coping with the complexity) were also considered, logic (reasoning/common sense) were also considered as key elements among teams.

6. CONCLUSIONS

In pair programming, pairs consist of two different personalities. The variation in performance is possible if the pairs are homogenous or heterogeneous. We presented the empirical analysis to find out these variations in terms of efficiency, accuracy and quality, quantity of bugs, brainstorming performance and creativity potential among these pairs during pair programming activity. As per comparative analysis we found homogenous female pairs stood second in overall performance. Heterogeneous pairs proved excellent in occurrence of bugs, took lead in brainstorming performance as well as in creativity potential. Comparative empirical analysis showed that the heterogeneous pair can perform better than homogenous.

REFERENCES:

Aybuke A., J. Cross, M. Handzic, C. V. Toorn, (2001) "Software Support for Creative Problem Solving". *IEEE Int. Conf. On Advanced Learning Technologies, ICALT'01*, 160-163, Madison, Wisconsin, USA© IEEE 2001.

Eric L., S. Robert O. Briggs, Gert-Jan deVreede, (2002) "Toward an Understanding of Creative Solution Generation", Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS-35.02), © 2002 IEEE 234-238.

Eric L. S. Robert, O. Briggs, Gert-Jan de Vreede, (2000) "The Cognitive Network Model of Creativity: a New Causal Model of Creativity and a New Brainstorming Technique". Proceedings of the 33rd Hawaii International Conference on System Sciences–2000, ©2000 IEEE 111-114.

Fabian Di F., P. Vandoren and F. V. Reeth, (2004) "Multimodal Interaction in a Collaborative Virtual Brainstorming Environment". LNCS 3190, 47-60. Springer-Verlag Berlin Heidelberg (ISSN: 0302-9743) ©2004 Springer

Frank P., M. Matthias M¨uller, (2003) "Analyzing the Cost and Benefit of Pair Programming". *Proceedings of the Ninth International Software Metrics Symposium (METRICS'03). © 2003 IEEE 22-30.*

Frank P., M. Matthias M uller, (2000) "Modeling the Impact of a Learning Phase on the Business Value of a Pair Programming Project". Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04) © IEEE 1031-1036.

Hans G., E. Arisholm, and T. Dyba, (2003) "An Initial Framework for Research on Pair Programming". Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03) © 2003 IEEE

Hiroshi N., J. Favela, A.L. Morán, D. Decouchant, and A.M. Martinez-Enriquez, (2003) "Distributed Pair Programming on the Web**".** Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03) © 2003 IEEE

Jennifer B., L. Werner, C. McDowell, (2002) **"**Guidelines for the Use of Pair Programming in a Freshman Programming Class". Proceedings of the 15th Conference on Software Engineering Education and Training (CSEET.02) © 2002 IEEE 53-58.

Jillian M.. H. Thomas and L. Rodgers, (2002) "Improving Group Creativity: Brainstorming Versus Non-brainstorming Techniques in a GSS Environment". Proceedings of the 34th Hawaii International Conference on System Sciences – 2001 © 2002 IEEE

Kent B., (1999) "*Extreme Programming Explained, Embrace Change*". *Addison-Wesley, Reading, mass* .

Kim M. L and C.Keith (2004) "A Cognitive Model for Solo Programming and Pair Programming". Proceedings of the Third IEEE International Conference on Cognitive Informatics (ICCI'04) © 2004 IEEE

Laurie W., Charlie McDowell, N. Nagappan, J. Fernald, and L. Werner, (2003) "Building Pair Programming Knowledge through a Family of Experiments"**.** Proceedings of the 2003 International Symposium on Empirical Software Engineering (ISESE'03) © 2003 IEEE.2033.

Laurie W. (2001) "Integrating Pair Programming into a Software Development Process". Proceedings 14th Conference on Software Engineering Education and Training February 19-21, 2001 Charlotte, North Carolina© 2001

Martin L., P. Becker-Pechau, H. Breitling, J. Koch, A. Kornstädt, S. Roock, A. Schmolitzky, H. Wolf, and H. Züllighoven, (2003) "Developing Complex Projects Using XP with Extensions". *IEEE Computer Society. © 2003 IEEE*

Radmila J. (2000) **"**Extreme Programming and its Development Practices". 22nd Int. Conf. lnformation Technology lnterfaces / TI 2000, June 13-16, 2000, Pula, Croatia ©IEEE 23-26.