An Experimental Study to Investigate Personality Traits on Pair Programming Efficiency in Extreme Programming

Ramlall Poonam
Computer Science & Engineering Dept.
University of Mauritius
Reduit, Mauritius
e-mail: poonam.ramlall1@umail.uom.ac.mu

Chuttur M. Yasser
Software and Information Systems Dept.
University of Mauritius
Reduit, Mauritius
e-mail: y.chuttur@uom.ac.mu

***Abstract*—Human Factors should not be overlooked in software development project planning. While much emphasis is often laid on technical aspects of projects, attention is rarely given to the influence of team members' personality on project success. This study investigates the effects of personality traits on the efficiency of pair programming when programmers are at the same location compared to being at different locations. The personality test model used is the Myer-Briggs Indicator Type which comprises of four traits, namely: extraversion/ introversion, sensing/intuition, thinking/feeling and judging/perceiving. A series of experiments was conducted and preliminary findings indicate that depending on location, personality traits can have an effect on programmers' efficiency. Results presented in this paper are expected to inform project managers in planning for successful software development projects, which require pair programming.**

Keywords-pair programming; project management; personality traits; human factors; software development

I. INTRODUCTION

Extreme programming (XP) is a well-known agile software development methodology created by Kent Beck [1]. As the name suggests, XP takes programming best practices to the extreme, thus assisting in the development of good quality software, which has a higher probability in satisfying customer needs within reasonable time frame [2]. XP comprises several practices of which, pair programming is considered as being very important in ensuring coding quality [3]. With the increase in the popularity of XP [4], pair programming has also attracted much attention even though it already existed in early 1995 [5].

In pair programming, two persons sit together and work collaboratively to write software codes. One of the developer acts as the "driver" and the other one as the "navigator". Normally the driver is the one who types codes and the navigator is the one observing the work of the driver and identifies any tactical defects and provides strategic planning [6]. The pair periodically switches roles, so that on overall the output is efficient [7]. A pair can either be heterogeneous or homogeneous [8]. In a heterogeneous pair, each programmer exhibits different personality traits while in a homogeneous pair each individual expresses similar traits.

Traditionally, pair programming involves two programmers working on the same project in the same location at the same time [9]. With globalization and network communication facilities, however, pair programming has evolved from being conducted in the same location to being carried out from different locations [10]. With the support of screen sharing technology and other collaborative tools, programmers can easily work in pairs regardless of locations. All communication, such as screen sharing or conversation between the pair is made synchronously on the same development project seamlessly.

In general, it is agreed that personality traits of programmers have a direct effect on project success when working in pairs [11]. However, as evidenced by the mixed results reported in previous studies, (*see* [12]-[18]), it is unclear on how and under what conditions do personality traits influence pair performance such that it is difficult to translate research findings to actual practice. More so, it is argued that pair performance when a pair works together in the same location does not necessary hold true when the same pair works from remote locations [19][20]. Consequently, further investigation into the problem is warranted.

To this end, it is essential to compare how personality traits may influence pair programming performance depending on location for due consideration in software development project planning. In this study, therefore, the effects of personality traits on efficiency of pair programming when programmers are collocated and are in foreign location will be investigated. Different hypotheses will be formulated and data will be collected through several experiments and analyzed in order to validate each hypothesis.

II. PERSONALITY TRAITS

Personality can be referred to the combined effect of physical, mental, social and emotional traits of a person as reflected in a behavioral pattern, which then becomes typical of the traits of that person [21]. Various psychologists have given various definitions for personality and it is well understood that several factors like heredity, learning, motivation, emotion, intelligence, environment, previous training, education, etc., shape the personality of an individual. In general, the personality of an individual can be measured across quantifiable parameters grouped under a set of dimensions often specified in a framework. Several such frameworks have been proposed to measure the personality

978-1-5386-5748-5/18/$31.00 ©2018 IEEE

of individuals. Examples are the: Myer-Briggs Type Indicator (MBTI) [22] [23], Eysenck Personality model [24] [25] and Five Factor Model (FFM) [26] [27]. Each personality trait measurement framework has its strengths and weaknesses and is more suitable than the other depending on the contexts and purposes.

In the present study, MBTI is perceived to be the most suitable framework for measuring personality traits of programmers. This is because MBTI is a popular framework, which has already been effectively used in pair programming studies [28]-[33]. Furthermore, it has provided reliable results when used in related areas of training and consultancy [22] and Information system and Software Engineering [34].

III. Myer-Briggs Type Indicator (MBTI)

MBTI is a framework, which has been designed to measure the personality traits of a person [22]. In this framework Myers and Myers [23] group personal behaviour in four different cognitive functions a) Extraversion versus Introversion b) Sensing versus Intuition c) Thinking versus Feeling and d) Judging versus Perceiving.

A. Extraversion versus Introversion

The 'extraversion and introversion' function describes how people are influenced from the outer world. The introverts are people who hides their own opinion whereas the extraverts are people who are widely open when dealing with others.

B. Sensing versus Intuition

The 'sensing and the intuition' function refers to the way that people collect information. Sensing pay attention to reality whereas intuition pay attention to impression or pattern of the information.

C. Thinking versus Feeling

The 'thinking and feeling' function considers how people make a decision. The thinkers focus on logical and analytical method for decision-making whereas feelers focus on personal values and understand the current situation to collect evidence.

D. Judging versus Perceiving

The 'judging and perceiving' function refers to the organising plan of a person. Judging prefers an organised and careful plan when dealing with people whereas perceiving is more open and flexible.

IV. Research Questions and Hypotheses

Too often, emphasis is placed on techniques, processes, and methods involved in developing software at the expense of human factors. Although several studies have looked into the effect of personality traits on pair programming efficiency, mixed results reported prevent the translation of research findings to actual practice. Added to this dilemma, is the evolving nature of pair programming from being collocated to being conducted from different locations. Hence, the following research questions (RQ) are revisited/formulated.

RQ1: Is there a relationship between personality traits and pair programming efficiency?

RQ2: Is there a relationship between pair programming efficiency and the location (same/different) of pair programmers?

To address the above research questions, the following hypotheses are assessed.

H_{A0}: Personality traits have no effect on efficiency when programming in pairs in collocated situation.

H_{A1}: Personality traits have an effect on efficiency when programming in pairs in collocated situation.

H_{B0}: Personality traits have no effect on efficiency when programming in pairs when at foreign location.

H_{B1}: Personality traits have an effect on efficiency when programming in pairs when at foreign location

V. Methodology

As illustrated in Table I, a two-group experimental design was adopted to investigate the research questions presented earlier. Subjects were divided into four groups: same personality, same location; same personality, different locations; different personality, same location; and different personality, different locations. Location and Personality were defined as independent variables (IV) whereas Efficiency was defined as the dependent variable (DV). Efficiency was measured in terms of speed and quality. Speed was recorded as the time taken to produce the required codes whereas quality was the number of errors detected per code size.

TABLE I. Two Group Experimental set up

		Independent variable: Location	
		Local	**Remote**
Independent Variable: Personality	**Same personality**	Group A (n=20)	Group B (n=20)
	Different personality	Group C (n=20)	Group D (n=20)

A. Subject Selection and Recruitment

The University of Mauritius offered a suitable pool of subjects for this study. Level 2 and level 3 undergraduate students in computer science or related programmes of study, who have knowledge in Java programming were requested to fill out a questionnaire to assess their programming proficiency and academic level. The questionnaire was distributed by email or during face-to-face presentation of the project in classes. A total of 147 students filled and returned the questionnaires. Thirty-six applicants were not retained either due to missing information in the questionnaire submitted or for not being suitable for the study. The remaining 111 were contacted by

phone and email and only 87 responded positively. Seven students were selected for pretesting the experiment and the remaining 80 students were assigned to each of the four experimental groups for the experiment. Comments received during the pretesting phase were taken on board to refine the final experiment.

B. Assessing Personality Traits of subjects

Prior to the experiment, the selected 80 students were requested to take an online test, which evaluated personality traits. The online test had several multiple-choice questions, which was based on evaluating the cognitive functions of the MBTI. Results obtained from the test were used to group the subjects into either having same personality traits or having different personality traits. It must be highlighted that the results of the online test presented the personality traits of subjects using four dimensions as per the four cognitive functions of MBTI (i.e., Extraversion versus Introversion, Sensing versus Intuition, Thinking versus Feeling and Judging versus Perceiving) such that four combinations of personality traits were obtained for a particular subject. Based on personality traits expressed by all four dimensions of the online test, it was not possible to distribute the 80 subjects into two groups of 40: one group of participants with same personality and the other group with different personality. This is because there were fewer subjects (< 40) who had similar personality traits on all four dimensions compared to the number of subjects who exhibited different personality. To address this limitation and to obtain a balanced number of subjects in each experimental group, only two dimensions of the online personality traits test were considered. Thus, the subjects could be equally shared into two groups of either similar traits or different traits.

C. Conducting the Experiment

Pair programming requires a programming problem with two programmers trying to solve the same problem through coding. Selected subjects were considered to all as having similar programming level (based on selection criteria) and were initially separated into two groups – one with similar personality, the other with different personality. Then, groups of 20 subjects satisfying criteria laid down on Table1 were formed through random assignment. A total of four groups were obtained for the experiment and IDs were used to keep all participants anonymous. To cater for the location variable, two computer labs at the University of Mauritius were used. Subjects falling under the same location group were asked to work in pair on the same computer in the same computer lab whereas subjects falling under remote locations were asked to work individually in each lab. For remote locations, groups working in pairs used an online collaborative tool to work on the same programming problem.

The same programming problem was given to all participants. The problem given constituted a situation, which required knowledge of arrays in Java. Array problems were selected for two reasons: 1) it requires a certain level of understanding for the Java programming language 2) the complexity level of array problem (as demonstrated during the pretesting phase) helped trigger collaboration between the pairs. At the start of the experiment, subjects were briefed about the study and given time to settle. All participants were asked to pay careful attention to the quality of the program paying particular attention to indentation, commas, comments and best practices in programming. Those instructions ensured that participants were aware of expectations for the study. Participants were also requested to notify the research coordinator once all the tasks were completed for the experiment.

D. Data Collection

A spreadsheet was used to record the efficiency of each pair for each group. Efficiency was calculated as the time taken by a pair to solve the task given for the experiment multiplied by the quality score of the programmers (measured in terms of errors detected in the final code produced by each pair). Time was recorded when a pair started addressing the problem given until the pair completed the task. Once subjects were given permission to leave the lab, the produced codes were saved in an editor for later analysis.

VI. Data Analysis and Results

The experiment was carried out using four groups: A, B, C and D (see Table I). Independent variables were *location* and *personality traits* whereas the dependent variable was *efficiency* (to be interpreted inversely, i.e., a higher efficiency score implies lower performance). Two independent coders calculated quality score. The researchers resolved any discrepancy before conducting any data analysis. SPSS software was used for two-way ANOVA calculations between groups. A plot of mean efficiency of the four groups is shown in Fig. 1.

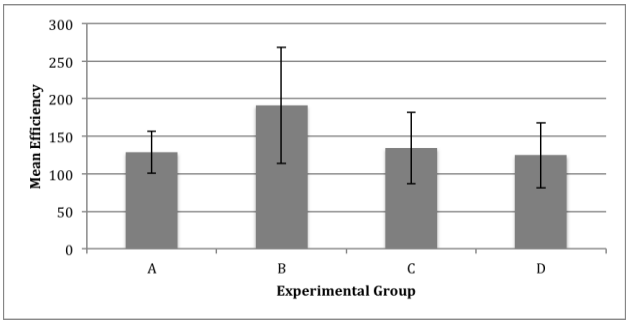

Figure 1. Mean efficiency plot with standard deviation for groups A-D.

TABLE II. ANOVA Results for Collocated Groups (Same vs. Different Traits)

Efficiency Score

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	156.800	1	156.800	.103	.752

Two-way analysis showed that there was a statistically significant interaction between the effects of location and

personality traits on pair programming efficiency, $p < 0.05$. Simple main effects analysis was further conducted for each level of the independent variables and results obtained are displayed in Tables II to V.

Table II shows a p value > 0.05. This indicates that there is not sufficient evidence to reject the null hypothesis H_{A0}. In other words, when pairs work on the same problem and are in the same location, it seems likely that personality will have no effect on pair programming performance.

TABLE III. ANOVA RESULTS FOR REMOTE GROUPS (SAME VS. DIFFERENT TRAITS)

Efficiency Score

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	22177.800	1	22177.800	5.64	.029

Table III, in contrast, shows a p value < 0.05. This indicates that there is sufficient evidence to reject the null hypothesis H_{A0} and accept H_{A1}. In other words, when pairs work on the same problem with individual programmers working from different locations, it seems likely that personality traits will have an effect on pair programming performance.

TABLE IV. ANOVA RESULTS FOR SIMILAR PERSONALITY GROUPS (LOCAL VS. REMOTE LOCATION)

Efficiency Score

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	19531.250	1	19531.250	5.760	.027

The p value is <0.05 in Table IV. Thus, there is not sufficient evidence to reject H_{B0}. Pairs with similar personality traits perform equally regardless of location (same and remote).

TABLE V. ANOVA RESULTS FOR DIFFERENT PERSONALITY GROUPS (LOCAL VS. REMOTE LOCATION)

Efficiency Score

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	470.450	1	470.450	.228	.639

Table V displays a p value > 0.05. There is, therefore, sufficient evidence to reject the null hypothesis H_{B0} and accept H_{B1}. In other words, pairs with different personalities will more likely perform differently depending on location (local and remote).

VII. DISCUSSIONS

Table VI shows the significant effects observed between the four groups analysed. In general, there is no significant difference in pair programming performance observed between all groups except for the case when programmers had different personality traits and were working remotely.

TABLE VI. SIGNIFICANT EFFECTS OBSERVED BETWEEN GROUPS

	Local	**Remote**
Same personality	No significant effect	No significant effect
Different personality	No significant effect	Significant effect

This result partly corroborates with some of the previous findings described in [30], [31] and [32] whereby personality traits was not seen to have any significant effects on pair programming. Those studies were carried out with pairs working at same location but did not consider cases where pairs worked from remote locations. Here, it is observed that depending on location, personality traits will likely affect pair performance. It is interesting to see that location has no effect on efficiency when pairs of similar traits work together. Such finding supports the argument by [34] who argues that communication and collaboration issues may compromise the success of pair programming. Individuals of similar traits usually communicate easily and understand each other regardless of location such that performance is optimum.

There are possible reasons to explain why personality traits at same location do not influence the efficiency of pair programmers even when personality traits are different. In face-to-face situations, developers have a tendency to adapt their behaviour to match the other pair member expectation and achieve the same goal set for the pair. Furthermore, with no communication barrier, it may be easier for pairs to communicate and collaborate.

Behaviour protocols are not same when in same location and at remote locations. Gaps of experiences and capabilities between two partners can lead to different coordination in the pair. Pairs working at foreign location have to communicate properly for the partner to understand the other. Depending on personality traits, it may be difficult for individuals to sometimes communicate, which may eventually influence pair efficiency. This is true especially when pairs with different personality traits work from remote location. As noted in Table VI, individuals with different personality traits will more likely perform differently than individuals with similar personality traits. Limitations imposed by communication facilities between remote team may be a further factor to affect pair performance.

There are some limitations to this study, which are worth mentioning. The same results presented here may differ depending on project types, duration of project, collaborative tool used, type of personality dimensions used to form pairs, expert level of participants (novice/experts), and pairs gender. Results presented in this study, however, remain useful in that project managers will want to consider the personality traits of individuals when setting up teams especially when collaboration is to be conducted from remote locations.

VIII. CONCLUSIONS

The influence of personality traits of programmers on pair performance is an important issue that cannot be ignored.

This study considered pair performance when working from same and remote locations. It was found that there is a relationship between personality traits of programmers, location of programmers and pair performance. In general, the effect of personality traits is only significant when collaboration takes place from remote locations. Due to several limitations uncovered during the study, further investigation is required to obtain a holistic understanding for the effect of human factors on pair performance.

REFERENCES

[1] BECK, K., 2000. *Extreme Programming Explained: Embrace Change.* Addison Wesley: Longman, Inc., Reading, Massachusetts.

[2] JURIC, R., 2000. Extreme programming and its development practices. *In: Proceedings o f the 22nd International Conference on Information Technology Interfaces*, 16 June 2000, Pula, Croatia. IEEE, 97-104.

[3] MULLER, M.M., PADBERG, F., 2003. On the economic evaluation of XP projects. *In Proceedings o f the 2003 European Conference on Software Engineering (ESEC 2003), September 2003*. Helsinki, Finland, 168-177.

[4] LIVERMORE, J.A., 2006. What elements of XP are being adopted by industry practitioners?. *Proceedings of the IEEE SoutheastCon,* p.149-152.

[5] CONSTANTINE, L.L., 1995. *Constantine on Peopleware*. Yourdon Press, Englewood Cliffs NJ

[6] LUI, K.M., CHAN, K., 2004. A Cognitive Model for Solo Programming and Pair Programming. *Proceedings of the Third IEEE International Conference on Cognitive Informatics,* 16-17 August 2004 Hong Kong. IEEE, p.94-102.

[7] WILLIAMS, L., KESSLER, R.R., 2002. *Pair Programming Iluminated*. Boston, MA, USA. Addision-Wesley Longman Publishing Co., Inc., p.67.

[8] MOHAMMED, S., ANGELL, L.C., 2003. Personality heterogeneity in teams: Which differences make a difference for team performance? Small Group Research, 34(6), 651 - 677

[9] LOWELL, L and RON, J., 2004. Extreme programming and agile software development methodologies. *Information Systems Management*, 41-52.

[10] WILLIAMS, L., 2012. What agile teams think of agile principles. *Communications of the ACM*, 55(4), 71-76.

[11] SFETSOS, P., ANGELIS, L., STAMELOS, I., 2006. Investigating the extreme programming system - an empirical study. *Empirical Software Engineering*, 11(2), 269-301.

[12] NOSEK, J.T., 1998. The case for collaborative programming. *Communication of ACM*, 3 March, p.105-108.

[13] COCKBURN, A., WILLIAMS,L., 2000. The Costs and Benefits of Pair Programming. *Humans and Technology Technical Report, 2000, 01 January 2000,* Sardinia, Italy.

[14] WILLIAMS, L., KESSLER, R.R., CUNNINGHAM, W., JEFFRIES, R., 2000. Strengthening the case for pair programming. *IEEE Software*, p.19-25.

[15] COCKBURN, A., 2000. Selecting a Project's Methodology. *IEEE Software*, 17(4), 64-71.

[16] LUI, K.M., CHAN, K., 2003. When does a pair outperform two individuals?. Proceedings o f the Fourth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2003), May 2003, Genova, Italy, p.225-233 102.

[17] THOMAS, L., RATCLIFFE, M., ROBERTSON, A., 2004. Code warriors and code-a-phobes: A study in attitude and pair programming. Proceedings o f the ACM Special Interest Group on Computer Science Education (S1GCSE 2003), August 2004, p. 94-102.

[18] JOHNSTON, A., JOHNSON, C.S, 2003. Extreme programming: A more musical approach to software development? Proceedings o f the Fourth International Conference on Extreme Programming and Agile Processes in Software Engineering (XP 2003), May 2003, P. 325-327.

[19] HERBSLEB, J., MOCKUS, A., FINHOLT, T., GRINTER, R., 2000. Distance, dependencies, and delay in a global collaboration. *In Proceedings of the 2000 ACM conference on Computer supported cooperative work*, December 2000, p.319-328.

[20] BAHETI, P., GEHRINGER, E., STOTTS, D., 2002. Exploring the efficacy of distributed pair programming. *Extreme programming and agile methods XP/Agile Universe*, 2002, p. 387-410.

[21] SHARMA, A., and DALIP M. 2007. *Personality and social norms*. New Delhi: Concept Pub. Co.

[22] FURNHAM, A., 1996. The big five versus the big four: The relationship between Myers-Briggs Type Indicator (MBTI) and NEO-PI five factor model of personality. *Personality and Individual Differences*, 21(2), 303- 307.

[23] MYERS, I.B., MYERS, P.B., 1995. Gifts differing: Understanding Personality Type. MountainView, CA: Davies-Black Publishing

[24] BURGER, J.M., 2010 . *Personality*. 8th Edition. Cengage Learning.

[25] EYSENCK, H. J., & EYSENCK, S. B. G. (1975) Manual of the Eysenck Personality Questionnaire. London: Hodder & Stoughton.

[26] MCCRAE, R.R., JOHN, O.P., 1992. An introduction to the five-factor model and its application. *Journal of Personality*, 60(2), 175-215.

[27] JOHN, O.P., SRIVASTAVA, S., 1999. The big five trait taxonomy: History, measurement and theoretical perspectives. *L. A. Pervin & O. P. John*. New York/London: The Guilford Press.

[28] SALLEH, N., MENDES, E., GRUNDY, J., 2011. Empirical studies of pair programming for CS/SE teaching in higher education: A systematic literature review. *IEEE Transactions*, 37(4), 509-525

[29] CHOI, K.S., 2004. *A discovery and analysis of influencing factors of pair programming*. Thesis (PhD), Faculty of New Jersey Institute of Technology.

[30] KATIRA, N., WILLIAMS, L., OSBORNE, J., 2005. Towards increasing the compatibility of student pair programmers. *27th International Conference on Software Engineering (ICSE'05)*, 15-21 May 2005. IEEE, 625-626.

[31] CHOI, K.S., DEEK, F.P., IM, I., 2008. Exploring the underlying aspects of pair programming: The impact of personality. *Information and Software Technology,* 50(11), 1114-1126

[32] LAYMAN, L., 2006. Changing students' perceptions: an analysis of the supplementary benefits of collaborative software development. *Proceedings of the 19th Conference on Software Engineering Education & Training (CSEET'06)*, 19-21 Aril 2006, p.159 - 166

[33] WILLIAMS, L., LAYMAN, L., OSBORNE, J., KATIRA, N., 2006. Examining the compatibility of student pair programmers. *Proceedings of the Conference on AGILE 2006 (AGILE'06),* 23-28 July 2003. IEEE Computer Society, 411-420.

[34] GORLA, N., LAM, Y.W., 2004. Who should work with whom? : Building effective software project teams. *Communications of the ACM*, 47(6), 79-82.