Exploring the Underlying Influencing Factors of Extreme Programming

Kyungsub Steve Choi

Dept. of Computer Information Systems, Manhattan College
Manhattan College Parkway, Riverdale, NY 10471, USA
Tel: +1-718-862-7309, Fax: +1-718-862-8032, E-mail: Kyungsub.choi@manhattan.edu

Abstract

A study on the pair programming productivity influencing variables is undertaken. Pair programming can be briefly summarized as an unorthodox programming paradigm in which two programmers share one keyboard and monitor during real-time programming sessions. This programming paradigm creates a high peer pressure programming session, where the two programmers are constantly pushed to their limits. Moreover, there are a number of unsuspecting underlying variables that significantly impact the outcome. One such variable is the psychosocial link between the two programmers. In this study, we have evaluated the personality factor. An empirical experiment is administered to a group of 68 computing majoring undergraduates and 60 graduate students. The results revealed that the group of two programmers whose personality is diversely matched has outperformed the group of two programmers whose personalities are either similarly matched or oppositely matched. The specific categories, or dependent variables, are code quantity, code quality, the level of communication, satisfaction, confidence and compatibility, and gender.

Keywords:

Pair Programming, Team programming, personality

Introduction

With the increasing presence of internet applications and their impacts on the business world, the software engineering community is well challenged and supported to develop and introduce new software process models and methods that comply with the current business ecosystem. The agile software process paradigm is one of the newer approaches that fit the requirements. There are many paradigms in agile software process paradigms, and the most prominent and popular one is extreme programming (XP) (Beck, 2000). XP boasts twelve core practices, and one of the twelve core practices is pair programming. Briefly, pair programming can be described as a programming activity where two individuals sit next to each other sharing one keyboard and one display. Each programmer takes turns in “driving” the keyboard while the other programmer is “navigating” the activity (Williams and Kessler, 2002). Aside those explicit characteristics, because this programming activity requires a high level of pair collaboration, communication, and team work, we suspect some unknown underlying variables that would eventually influence the pair programming experience and output.

Among a list of unknown underlying variable candidates, one we have chosen to experiment is personality variable. This is because, similar to one’s handwritten signature, programming is a distinct personal activity. When two programmers with different personalities are paired for one programming goal, each must compromise and collaborate to achieve that goal.

Experiment

The basic experiment flow is 1) setup a number of subject groups according to their personality types, 2) each group then goes through a set of programming problems, and 3) finally their work is judged and evaluated. In devising the experiment, the first task was allocating a suitable personality assessment instrument. Among many, one that is most widely used and familiar to the public is Myers-Briggs Type Indicator (MBTI) (Bayne, 1995; Keirsey, 1998; Myers & Myers, 1995) as it’s used in 84 of the Fortune 100 companies and more than 50 different countries according to CCP Inc., the official MBTI material distributor.

MBTI measures a person’s preferences using four basic scales with opposite poles. The four scales are: (1) extraversion/introversion: EXTRAVERSION – people who prefer Extraversion tend to focus on the outer world of people and things, INTROVERSION - people who prefer introversion tend to focus on the inner world of ideas and impressions; (2) sensate/intuitive: SENSING - people who prefer sensing tend to focus on the present and on concrete information gained from their senses, INTUITION - people who prefer intuition tend to focus on the future, with a view toward patterns and possibilities; (3) thinking/feeling: THINKING - people who prefer thinking tend to base their decisions primarily on logic and on objective analysis of cause and effect, FEELING - people who prefer feeling tend to base their decisions primarily on values and on subjective evaluation of person-centered concerns; and (4) judging/perceiving: JUDGING - people who prefer judging tend to like a planned and organized approach to life and prefer to have things settled, PERCEIVING - people who
prefer perceiving tend to like a flexible and spontaneous approach to life and prefer to keep their options open.

The various combinations of these preferences result in 16 total personality types and are typically denoted by four letters—for example, INTJ (Introversion, Intuition with Thinking and Judging)—to represent one’s tendencies on the four scales.

The type theory (Bayne, 1995) generally asserts that one of the four preferences - Sensing, Intuition, Thinking or Feeling - usually dominates the others. For example, a person uses the dominant type the most and feels most comfortable when using it; it is an essential part of the person at his or her best. A person uses the dominant type in his or her daily life. Complementing the dominant type is the auxiliary or secondary type. The auxiliary type is thought of as a second dominant type in that a person uses it more than any type except for the dominant type. As with the dominant type, the auxiliary type is readily used and a person will frequently unconsciously shift back and forth between the dominant and auxiliary. This leads to the following possible combinations: ST, SF, NF, and NT, where the first letter of each type represents the dominant type and the second letter represents the auxiliary type. For example, ST can be found at ESTJ, ISTJ, ESTP, and ISTP.

In our experiment, the type theory is utilized to group the subjects. We grouped the subjects into a ‘diverse’ group, an ‘opposite’ group, and an ‘alike’ group:

‘Diverse’ group: These are pairs of subjects who are alike in EITHER their dominant OR auxiliary preferences but not both (ST-SF, NT-NF, ST-NT, and SF-SF). Labeled as [divrs] in this experiment.

‘Alike’ group: These are pairs of subjects who are alike in BOTH their dominant AND auxiliary preferences (ST-ST, NF-NF, NT-NT, and SF-SF). Labeled as [alike] in this experiment.

‘Opposite’ group: These are pairs of subjects who are completely OPPOSITE in BOTH their dominant and auxiliary preferences (ST-NF and NT-SF). Labeled as [opp] in this experiment.

For the subjects, we used a group total of 128 computing majoring students (management information systems, information systems, information technology): 68 undergraduate students (40 undergradmen and 28 upperclassmen) and 60 graduate students pursuing masters degrees. In pairing the students, besides the MBTI factor, many factors were considered. We had considered each student’s grade from his or her current programming course, and each student’s any prior programming experience. In addition to the code evaluation, the pair programming experience is also evaluated in terms of subjects’ satisfaction, confidence, compatibility, communication, and gender.

For the experiment task, we have devised a set of four similar programming problems. A group of three professional programmers from outside is invited to evaluate the set for its appropriateness and difficulty level. Upon the review, all four problems were deemed to be similar in their level of programming difficulty.

The flow of the experiment was carried out so that each pair would receive two programming sessions where each session has a different date and different programming problem. At the end of each session, 45 minutes long, the work is collected and then given to a set of two independent judges, expert programmers, for scoring. The scoring or evaluation is done on a scale of 0 to 10, 10 being the excellent work score. The categories are code quality and code productivity.

The Analysis of Results

For the code evaluation, table 1 shows that the [divrs] pairs had higher scores than both [opp] and [alike] pairs in the categories of code efficiency and code productivity. The mean score is referring to the judges’ scores on code evaluation. For example, [divrs] pairs received a mean value of 6.46 on code productivity, and a mean value of 6.49 on code efficiency from the judges.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Code prod.</td>
<td>[opp]</td>
<td>40</td>
<td>5.31</td>
<td>3.19</td>
</tr>
<tr>
<td></td>
<td>[divrs]</td>
<td>40</td>
<td>6.46</td>
<td>2.43</td>
</tr>
<tr>
<td></td>
<td>[alike]</td>
<td>43</td>
<td>4.64</td>
<td>3.03</td>
</tr>
<tr>
<td>Code Eff.</td>
<td>[opp]</td>
<td>40</td>
<td>5.24</td>
<td>3.05</td>
</tr>
<tr>
<td></td>
<td>[divrs]</td>
<td>40</td>
<td>6.49</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td>[alike]</td>
<td>43</td>
<td>4.62</td>
<td>3.05</td>
</tr>
</tbody>
</table>

Additionally, the groups are significantly different among each other in both code productivity and code efficiency. The Kruskal-Wallis test result show both code productivity and code efficiency value p >0.05 (code productivity p = 0.029, code efficiency p = 0.021).
Table 2 Test Statistics of [pairtype]

<table>
<thead>
<tr>
<th></th>
<th>Code productivity</th>
<th>Code efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chi-Square</td>
<td>7.067</td>
<td>7.710</td>
</tr>
<tr>
<td>df</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Asymp. Sig.</td>
<td>.029</td>
<td>.021</td>
</tr>
</tbody>
</table>

Kruskal Wallis Test
Grouping Variable: [pairtype]

A 37-item Likert-scale question is designed to capture those experiences. They were checked for validity and reliability by performing factor analysis and Cronbach reliability measurement (Rosenthal and Rosnow, 1991; Straub, 1989). The factor analysis was used by means of principal components analysis with varimax rotation. The parameters are 1) eigenvalues greater than one and 2) maximum iteration for convergence was set to be 25. Typically, factor analysis is repeated until all item values are acceptable. For example, after a factor analysis, any item’s value with less than 0.500 is discarded (Straub, 1989) as is any construct with less than three items (minimum) as well. Only after all item values are 0.500 or higher and all constructs possess a minimum of three questions is the factor analysis complete. The Cronbach reliability measurement indicates how well a set of items measures a single latent construct. Typically, 0.700 is viewed as the acceptable minimal value and the higher the value, the more reliable the set is (Cohen and Swerdlik, 2002). After a few rounds of factor analysis we have the Cronbach result:

Table 3 Cronbach’s Reliability Measurement

<table>
<thead>
<tr>
<th>Constructs</th>
<th>Reliability (Alpha)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP comm.</td>
<td>0.8794</td>
</tr>
<tr>
<td>PP satisf.</td>
<td>0.8909</td>
</tr>
<tr>
<td>PP conf.</td>
<td>0.8314</td>
</tr>
<tr>
<td>PP compat.</td>
<td>0.7084</td>
</tr>
</tbody>
</table>

Based on these results, we have found that except for the communication, satisfaction, confidence, and compatibility are not found to be significantly different among the pairs; [divrs], [alike], and [opp]. On the gender variable, the female-female pairs exhibited the highest levels of satisfaction and compatibility. However, the male-male or male-female pairs did not exhibit similar results.

The following is a summary of experimental data and conclusions based on results from this study: 1) the diversely MBTI type matched pairs outperformed both oppositely and similarly MBTI type matched pairs, 2) there are no significant differences between all three differently MBTI type matched groups in the levels of communication, satisfaction, confidence and compatibility, 3) the pairs that are diversely MBTI type paired and also male-male paired showed to be the best performing teams.

Conclusion

A list of theoretical and practical implications can be derived from the results of the experiment. Theoretical implications include the notion that the MBTI personality type has a significant impact on PP. Different combinations of the MBTI personality type clearly illustrate different levels of influences on PP as is evidenced in the areas of code productivity and code efficiency. When examining the gender factor, same gender pairs show a higher level of communication, satisfaction, and compatibility while engaged in PP, with female and female pairs showing the strongest response. Other implications are that confidence on a job completed is not correlated to communication skill level, satisfaction, compatibility, MBTI personality type, or gender. Lastly, given the fact that measuring one’s communication skill level is a difficult task, the instrument that was used to measure this attribute proved to be an effective and valid one. Many practical implications from this experiment can also be seen. One notable suggestion is for programming shop managers to pair the programmers according to one’s personality profile, or MBTI personality type. Also a higher level of communication, satisfaction, and compatibility between the programmers can be achieved through same gender pairing. Additionally, it can be inferred that an appropriate level of managerial intervention and support in the pairing process is needed to alleviate the very fast and demanding pace of PP and also minimize any non-work related motives such as personal favoritism or office politics.

References