An Analysis of the Effects of Company Culture, Education and Experience on Confirmation Bias Levels of Software Developers and Testers

Gul Calikli
Software Research Laboratory
Department of Computer Engineering, Bogazici University, Istanbul, Turkey.
0090 3595400-7227
gul.calikli@boun.edu.tr

Ayse Bener
Software Research Laboratory
Department of Computer Engineering, Bogazici University, Istanbul, Turkey.
0090 3595400-7226
bener@boun.edu.tr

Berna Arslan
Software Research Laboratory
Department of Computer Engineering, Bogazici University, Istanbul, Turkey.
0090 3595400-7227
ba.arslan@boun.edu.tr

ABSTRACT
In this paper, we present a preliminary analysis of factors such as company culture, education and experience, on confirmation bias levels of software developers and testers. Confirmation bias is defined as the tendency of people to verify their hypotheses rather than refuting them and thus it has an effect on all software testing.

Categories and Subject Descriptors
H.1.2 [User/Machine Systems]: Human Factors, Software Psychology

General Terms
Measurement, Experimentation, Human Factors.

Keywords
Cognitive biases, confirmation bias, software engineering, software testing.

1. INTRODUCTION
Besides technical issues, difficulty in the study of software engineering arises due to human aspects. Among these human aspects are the cognitive biases, which are defined as the deviation of human mind from the laws of logic and accuracy [1]. The notion of cognitive biases was first introduced by Tversky and Kahneman [2, 3]. There are various cognitive bias types and confirmation bias is one of these biases which is likely to affect software development process. The tendency of people to seek for evidence that could verify their theories rather than seeking for evidence that could falsify them is called confirmation bias [1]. The term confirmation bias was first used by Peter Wason in his rule discovery experiment [4]. In his work, Wason challenged subjects to identify a rule applying to triples of numbers, starting from the information that the triple (2,4,6) fits the rule. For each triple generated by the subject, the experimenter gave feedback by telling whether or not each triple conformed to the rule. One can succeed only by employing eliminative strategy rather than enumerative strategy [4, 7]. In other words, the subject must try refute the hypothesis in his/her mind in order to discover the exact rule, rather than trying triples that conform to his/her hypothesis one by one.

Wason also explained the results of his selection task experiment using facts based on confirmation bias [5]. In this task, Wason gave subjects partial information about a set of objects, and asked them to specify what further information they would need to tell whether or not a conditional rule (“If A, then B”) applies. It has been found repeatedly that people perform badly on various forms of this test, in most cases ignoring information that could potentially refute the rule.

According to an empirical evidence, testers are more likely to choose positive tests rather than negative tests [6]. However, during all levels of software testing the attempt should be to fail the code to reduce software defect density. In order to discover more defects during all levels of testing, confirmation bias levels of testers and developers need to be low. In this empirical study, we perform a preliminary analysis and investigate whether confirmation bias can be circumvented by some factors such as company culture, education and software development/testing experience.

The rest of the paper is organized as follows: Proposed approach, which is making tests to measure/quantify confirmation bias levels of individuals, is explained in Section II. Section III mentions the metrics extracted from these tests. Information about the dataset used in this empirical analysis is given in Section IV. Section V consists of the preliminary results. Finally, the impact of the results and possible future directions are discussed in Section VI.

2. PROPOSED APPROACH
In order to perform an empirical analysis, we needed a methodology to measure/quantify confirmation bias levels of individuals. For this purpose, we prepared two types of tests that are interactive test and written test respectively.

2.1 Interactive Test
What we call interactive test is Wason’s rule discovery task [5]. As in the original task, subjects are given a record sheet and they are asked to discover the rule by writing down triples together with the reasons for their choice. Tester gives feedback to the subject by telling whether the triple written by the subject...
conforms to the rule or not. The subject can announce the rule only when he/she is highly confident. The experiment ends if the rule is discovered. If the subject cannot discover the rule, he/she can continue giving instances together with reasons for his/her choice. This procedure may continue iteratively until either the subject discovers the rule or he/she wishes to give up. There is no time limit, however if subject cannot discover the rule in 45 minutes, the experimenter aborts the test.

2.2 Written Test

Written test is based on Wason’s selection task [9]. In the original task, the subject is given four cards, where each card has a letter on one side and a number on its other side. These four cards are placed on a table showing respectively D, K, 4, 7. For example, given the rule: Every card that has a D on one side has a 3 on the other side, the subject is asked which card(s) must be turned over to find out whether the rule is true or false. Besides the original experiment that is abstract, various replications with thematic content have been performed [7, 8].

2.2.1 Abstract Questions

To answer all kinds of abstract questions correctly requires pure logical reasoning ,while some can also be answered correctly by matching.

In our test there are 8 abstract questions. One of these questions is abstract-thematic [7], and hence it requires logical reasoning despite its thematic content. The remaining questions are purely abstract.

2.2.2 Thematic Questions

Subject can answer thematic questions correctly using the cues produced by memory. This phenomenon where the stage of logical reasoning is bypassed is called memory cueing [8]. In our test there are 6 thematic questions which can be solved correctly through everyday life experience.

2.2.3 Questions with Software Development/Testing Theme

This type of questions is also thematic questions where pure logical reasoning can be bypassed by experience in software development and testing. Yet, the answers to these questions can be correct. Our test contains 8 questions of this type.

3. METRICS

In order to make an empirical analysis, some metrics have been defined by the authors, except for the eliminative/ enumerative index been inherited from Wason’s rule discovery task [6].

3.1 Interactive Test Metrics

3.1.1 Eliminative/Enumerative Index (IndElim/Enum)

Eliminative/ enumerative index was constructed by Wason [6] in order to determine the proportion of the total number of instances that are incompatible with reasons to those that are compatible. It is desirable that eliminative/ enumerative index is greater than one. The higher the eliminative/ enumerative index, the more tendency the subject has for refuting his own hypotheses.

3.1.2 Interactive Test Duration (T_D)

For each subject, total time in minutes it took to discover the correct rule is recorded during tests.

3.1.3 Immediate Rule Announcement Frequency (F_R)

This metric is the total number of times the subject makes a rule announcement without giving any instances after his/her previous incorrect rule announcement.

3.1.4 Average Length of the Series of Immediate Rule Announcement (avg_L_IR)

It was observed that some subjects announces a series of rules without giving any instances in between. We called the total number of immediate rule announcements in such a series, as the “length” of that series. During interactive tests, subjects who make one or more announcement series can be observed. We calculated the average length of these announcement series.

3.1.5 Average Frequencies of Reason Repetition/Reformulation (avg_F_RR)

During experiments some of the subjects wrote down the same reason(s) more than once with or without reformulation. Total number of times of repetitions are calculated for each reason. The results obtained are used to calculate the average of these values.

3.1.6 Total Number of Rule Discovery Attempts (N_AD)

This metric defines the total number of rule announcements made to discover the correct rule.

3.2 Written Test Metrics

The written test is analyzed in three separate parts, which are abstract questions, thematic questions and questions with software development/testing theme, respectively. Table 1 shows written test metric and their corresponding abbreviations. For each part, scores is the ratio of the total number of correct answers to the total number of questions in that part.

<table>
<thead>
<tr>
<th>Abbr.</th>
<th>Metric Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>S_ABS</td>
<td>Score in abstract questions</td>
</tr>
<tr>
<td>S_TH</td>
<td>Score in thematic questions</td>
</tr>
<tr>
<td>T_TH-ABS</td>
<td>Duration it took to solve abstract and thematic questions (minutes)</td>
</tr>
<tr>
<td>S_SW</td>
<td>Score in questions with software development/testing theme</td>
</tr>
<tr>
<td>T_SW</td>
<td>Duration it took to solve questions with software development/testing theme (minutes)</td>
</tr>
</tbody>
</table>

4. DATA

We performed both interactive and written tests in four main groups. First project group (Group 1) is the software developers/testers working in a large scale telecommunication in Europe. Group 1 consists of 34 software engineers. Members of the second group (Group 2) are employees of a large software

1 Abbr. stands for "Abbreviation".
development company in North America. Group 2 consists of 32 developers. Finally we have a control group (Group 3) that consists of 29 people who are graduate students at the Computer Engineering Department of Bogazici University. Both interactive and written tests have been conducted with these subjects and the metrics mentioned in the previous section are extracted from the results of each test.

5. RESULTS

As shown in Table 2, comparison of Group 1 and Group 2 resulted in a significant difference between interactive test performances. Although average eliminative/enumerative indices of both groups are greater than one, according to the results of the Kolmogorov-Smirnov tests, average eliminative/enumerative index of Group 2 is significantly higher than that of Group 1. This implies that members of Group 2 are more inclined to refute their own hypotheses to find the correct rule during the interactive test. In other words, the problem solving strategy they employ is more eliminative. Moreover, average time and total number of rule announcements it took for members of Group 2 to find the correct rule is significantly lower than those of Group 1.

According to the test protocol, at the beginning of the test subjects are told that after an incorrect rule announcement, they can either continue giving triple instances together with their reasons for choice, or give up the test if they wish. Despite this fact, while performing the interactive test to members of Group 1, repetition of the same reasons for choice several times have been observed more, as well as immediate rule announcements. These results are highly suggestive about the fact that members of Group 1 exhibit a more organized problem solving strategy which results in a better performance in the test.

As shown in Table 2, comparison of Group 1 and Group 2 resulted in a significant difference between interactive test performances. Although average eliminative/enumerative indices of both groups are greater than one, according to the results of the Kolmogorov-Smirnov tests, average eliminative/enumerative index of Group 2 is significantly higher than that of Group 1. This implies that members of Group 2 are more inclined to refute their own hypotheses to find the correct rule during the interactive test. In other words, the problem solving strategy they employ is more eliminative. Moreover, average time and total number of rule announcements it took for members of Group 2 to find the correct rule is significantly lower than those of Group 1.

According to the test protocol, at the beginning of the test, subjects are told that after an incorrect rule announcement, they can either continue giving triple instances together with their reasons for choice, or give up the test if they wish. Despite this fact, while performing the interactive test to members of Group 1, repetition of the same reasons for choice several times have been observed more, as well as immediate rule announcements. These results are highly suggestive about the fact that members of Group 1 exhibit a more organized problem solving strategy which results in a better performance in the test.

As shown in Table 3, although there is not a significant difference in written test solving durations, Group 3 members have higher scores on average. This is probably due to the fact that member of this group are dealing with mathematics and logic in their research studies. This makes them employ logical reasoning especially while solving abstract questions.

Due to the results obtained by comparing Group 1 and Group 3, we also explored the possible effects of experience in software development/testing on confirmation bias level. For this purpose we made two different analyses which are within Group 1 and within Group 2 respectively. For each analyses we divided each group into two subgroups as being group of experienced and less
experienced members. We defined the members of the experienced subgroup as individuals who have experience in software development and testing that is more than the average value. Hence members with experience that is equal to or less than the average value are labeled as “less experienced”. The average experience in software development and testing in Group 1 is 5.78 years. 17 of the Group 1 members have experience above 5.78 years, while those of remaining 17 are below the average. In Group 2, average years of experience is 12.22 years and 13 of the members have experience above 12.22 years whereas remaining 19 has years of experience less than this value. years in Group 2. As shown in Table 4 and Table 5, there is not a significant difference in terms of performance in both interactive and written tests.

Table 4. Results of the Kolmogorov-Smirnov test among the experienced and less experienced members of Group 1

<table>
<thead>
<tr>
<th></th>
<th>(Group 1)EXP</th>
<th>(Group1)NEXP</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ind_{Ellin/Enum}</td>
<td>1.11</td>
<td>1.12</td>
<td>0.6899</td>
</tr>
<tr>
<td>T_j</td>
<td>18.06</td>
<td>16.59</td>
<td>0.3874</td>
</tr>
<tr>
<td>F^{IR}</td>
<td>1.00</td>
<td>0.67</td>
<td>1.0000</td>
</tr>
<tr>
<td>avg_L^{IR}</td>
<td>0.55</td>
<td>0.53</td>
<td>1.0000</td>
</tr>
<tr>
<td>avg_F^{RR}</td>
<td>1.17</td>
<td>0.80</td>
<td>0.8644</td>
</tr>
<tr>
<td>N_5</td>
<td>3.61</td>
<td>2.18</td>
<td>0.1170</td>
</tr>
<tr>
<td>S_{ABS}</td>
<td>0.19</td>
<td>0.13</td>
<td>0.3874</td>
</tr>
<tr>
<td>S_{Th}</td>
<td>0.72</td>
<td>0.71</td>
<td>0.9313</td>
</tr>
<tr>
<td>T_{Th+ABS}</td>
<td>18.12</td>
<td>14.5</td>
<td>0.2336</td>
</tr>
<tr>
<td>S_{SW}</td>
<td>0.46</td>
<td>0.53</td>
<td>0.9303</td>
</tr>
<tr>
<td>T_{SW}</td>
<td>17.59</td>
<td>14.41</td>
<td>0.3874</td>
</tr>
</tbody>
</table>

6. CONCLUSIONS AND POSSIBLE FUTURE DIRECTIONS

During all levels of software testing the attempt should be to fail the code to reduce software defect density. In an early work, it has been empirically proven that people have more tendency to make positive tests rather than negative tests due to confirmation bias [6]. It is highly probable that confirmation bias can be circumvented. In order to find the effective methods for this purpose, we should be able to identify factors that have effect on confirmation bias. In this preliminary work, we analysed the possible effect of factors such as company culture, education and experience on confirmation bias.

As future work, we intend to perform a nonparametric factor analysis to further investigate possible factors affecting confirmation bias. Moreover, we intend to formulate the relation between defect density and tester performance with confirmation bias levels. Finally, we also plan to perform a Test Driven Development (TDD) versus non-TDD analysis. In TDD, developers write tests to fail the code [9]. Hence, we plan to perform an empirical analysis to investigate whether TDD circumvents side effects of confirmation bias or not.

Table 5. Results of the Kolmogorov-Smirnov test among the experienced and less experienced members of Group 3

<table>
<thead>
<tr>
<th></th>
<th>(Group 3)EXP</th>
<th>(Group3)NEXP</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ind_{Ellin/Enum}</td>
<td>1.75</td>
<td>1.48</td>
<td>0.2584</td>
</tr>
<tr>
<td>T_j</td>
<td>12.29</td>
<td>12.18</td>
<td>0.0937</td>
</tr>
<tr>
<td>F^{IR}</td>
<td>0.23</td>
<td>0.22</td>
<td>1.0000</td>
</tr>
<tr>
<td>avg_L^{IR}</td>
<td>0.13</td>
<td>0.13</td>
<td>1.0000</td>
</tr>
<tr>
<td>avg_F^{RR}</td>
<td>1.54</td>
<td>1.66</td>
<td>0.8217</td>
</tr>
<tr>
<td>N_5</td>
<td>1.96</td>
<td>1.85</td>
<td>1.0000</td>
</tr>
<tr>
<td>S_{ABS}</td>
<td>0.46</td>
<td>0.46</td>
<td>1.0000</td>
</tr>
<tr>
<td>S_{Th}</td>
<td>0.63</td>
<td>0.83</td>
<td>0.3873</td>
</tr>
<tr>
<td>T_{Th+ABS}</td>
<td>18.08</td>
<td>15.67</td>
<td>0.2421</td>
</tr>
<tr>
<td>S_{SW}</td>
<td>0.77</td>
<td>0.82</td>
<td>0.9048</td>
</tr>
<tr>
<td>T_{SW}</td>
<td>9.50</td>
<td>11.00</td>
<td>0.7437</td>
</tr>
</tbody>
</table>

7. ACKNOWLEDGMENTS

This research is supported in part by Turkish Scientific Research Council, TUBITAK, under grant number EEEAG108E014.

8. REFERENCES


