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 to *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 level 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

ICSE '10, May 2-8 2010, Cape Town, South Africa

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 ($Ind_{Elin/Enum}$)

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 that 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_I)

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^{IR})

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_A)

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 1. Written test metrics with their abbreviations

Abbr.[1]	Metric Explanation
S_{ABS}	Score in abstract questions
S_{Th}	Score in thematic questions
T_{Th+ABS}	Duration it took to solve abstract and thematic questions (minutes)
S_{SW}	Score in questions with software development/testing theme
T_{SW}	Duration it took to solve questions with software development/testing theme (minutes)

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.

Table 2. Results of the Kolmogorov-Smirnov test for Group 1 and Group 2 (average scores)

	Group 1	Group 2	p-value
$Ind_{Elin/Enum}$	1.04	1.59	0.0039
T_I	18.14	11.56	0.0351
F^{IR}	0.85	0.19	0.0535
avg_L^{IR}	0.54	0.11	0.0535
avg_F^{RR}	1.01	1.61	0.5001
N_A	3.07	1.78	0.0221
S_{ABS}	0.27	0.43	0.4285
S_{Th}	0.73	0.71	1.0000
T_{Th+ABS}	16.36	16.54	0.6826
S_{SW}	0.51	0.79	0.0008
T_{SW}	15.97	11.54	0.0319

Performance of Group 1 and Group 2 in abstract and thematic parts of the written test does not exhibit a significant difference. However, average performance of Group 2 is better in the written test with software development/testing theme at a significant level. Since performance in abstract test depends solely on pure logical reasoning compared to other written test parts, we can state that Group 2 members can master software development/testing domain well.

The significant difference in performances of Group 1 and 2 can be explained by various factors. In order to find out whether education is the main factor affecting test performances, we formed a control group (Group 3). The members of Group 3 are the graduate students at the Computer Engineering Department of Bogazici University, which is one of the most prominent university in Computer Engineering in Turkey. In order to make a comparison, we removed data that belong to members who are graduates of the most prominent universities in Turkey from Group 1. Our results are shown in Table 3. Kolmogorov-Smirnov tests showed that there is not a significant difference between the average eliminative/enumerative indices of Group 1 and Group 3. However, average time it takes to find the correct rule is significantly lower among Group 3 members. This might be due to the fact that there is always a tight schedule to rush the next release ready for the market in case of Group 1 members. On the other hand, Group 3 members are not subjected frequently to such a pressure and they have a more relaxed environment in terms of tight schedules.

Table 3. Results of the Kolmogorov-Smirnov test for Group 1 and Group 3 (average scores)

	Group 1	Group 3	p-value
$Ind_{Elin/Enum}$	1.19	1.04	0.39800
T_I	18.14	7.40	0.00430
F^{IR}	0.88	0.29	0.22220
avg_L^{IR}	0.65	0.25	0.22220
avg_F^{RR}	1.11	1.04	1.00000
N_A	3.07	2.12	0.59800
S_{ABS}	0.13	0.61	0.00006
S_{Th}	0.53	0.88	0.00670
T_{Th+ABS}	16.41	13.65	0.7259
S_{SW}	0.40	0.80	0.00003
T_{SW}	15.41	12.10	0.2749

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 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

	$(\text{Group 1})_{EXP}$	$(\text{Group1})_{NEXP}$	p-value
$Ind_{Elin/Enum}$	1.11	1.12	0.6899
T_I	18.06	16.59	0.3874
F^{IR}	1.00	0.67	1.0000
avg_L^{IR}	0.55	0.53	1.0000
avg_F^{RR}	1.17	0.80	0.8644
N_A	3.61	2.18	0.1170
S_{ABS}	0.19	0.13	0.3874
S_{Th}	0.72	0.71	0.9313
T_{Th+ABS}	18.12	14.5	0.2336
S_{SW}	0.46	0.53	0.9303
T_{SW}	17.59	14.41	0.3874

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

	$(\text{Group 3})_{EXP}$	$(\text{Group3})_{NEXP}$	p-value
$Ind_{Elin/Enum}$	1.75	1.48	0.2584
T_I	12.29	12.18	0.0937
F^{IR}	0.23	0.22	1.0000
avg_L^{IR}	0.13	0.13	1.0000
avg_F^{RR}	1.54	1.66	0.8217
N_A	1.96	1.85	1.0000
S_{ABS}	0.46	0.46	1.0000
S_{Th}	0.63	0.83	0.3873
T_{Th+ABS}	18.08	15.67	0.2421
S_{SW}	0.77	0.82	0.9048
T_{SW}	9.50	11.00	0.7437

7. ACKNOWLEDGMENTS

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

8. REFERENCES

[1] Stacy, W. and MacMillan, J., 1993. Cognitive bias in software engineering. Communication of the ACM. 38, 6 (June 1995), 57-63. DOI= http://doi.acm.org/10.1145/203241.203256

[2] Kahneman D., Slovic P., and Tversky, A. (Eds.) 1982 *Judgment Under Uncertainty: Heuristics and Biases*. New York: Cambridge University Press ISBN 978-0521284141

[3] Tversky, A. and Kahneman, D. 1971. Belief in the law of small numbers. Psychological Bulletin, 76, 105-110.

[4] Wason, P. C. 1960. On the failure to eliminate hypotheses in a conceptual task. Quarterly Journal of Experimental Psychology (Psychology Press), 12. 129–140.

[5] Wason, P. C. 1968. Reasoning about a rule. Quarterly Journal of Experimental Psychology (Psychology Press) 20: 273–28.

[6] Teasley, B., Leventhal, L. M., and Rohlman, S. Positive test bias in software engineering professionals: What is right and what's wrong. In *Empirical Studies of Programmers: Fifth Workshop,* C.R. Cook, J.C. Scholtz, and J. C. Spohrer, Eds.1993.

[7] Poletiek, F. 2001 Hypothesis Testing Behavior (Essays in Cognitive Psychology). Psychology Press Ltd.

[8] Wason, P. C. and Shapiro, D. 1971, Natural and contrived experience in a reasoning problem, Quarterly Journal of Experimental Psychology (Psychology Press) 23: 63–71.

[9] Erdogmus, H., Morisio, M., and Torchiano, M. 2005. On the effectiveness of test-first approach to programming. IEEE Transactions on software Engineering, 31, 1, 226-237.