Analyzing the Effects of Confirmation Bias on Software Development Team Performance: A Field Study during a Hackathon

Conference or Workshop Item

Analyzing the Effects of Confirmation Bias on Software Development Team Performance: A Field Study during a Hackathon

G. Calikli[1], A. Bener[1] and F. Shirazi[2]

[1]Data Science Laboratory, Ryerson University,
Toronto, ON M5B2K3, CANADA
{gcalikli, ayse.bener}@ryerson.ca

[2]Ted Rogers School of ITM, Ryerson University,
Toronto, ON M5B2K3, CANADA
f2shiraz@ryerson.ca

ABSTRACT

Thought processes and cognitive aspects of people have a significant impact on software quality, as software is designed, implemented and tested by people. In this preliminary research, we conducted a field study during a 24 hour non-stop exploratory software development event: "hackathon". During hackathons, people collaborate intensively on software projects. The focus of this hackathon was application software development on a specific operating system (OS) that is for use on mobile phones, laptops, tablets and PCs. In this study, we analyzed the relation between confirmation bias levels of development teams and their performance. Confirmation bias is a specific type of cognitive bias and it is defined as the tendency of people to seek for evidence to verify their hypotheses rather than seeking for evidence to refute them [1]. Due to confirmation bias developers may perform the tests that make their program work rather than breaking their code. This, in turn leads to an increase in software defect density [2]. Ideally, during all levels of software testing, a systematic procedure should be followed. Therefore, based on the findings by Poletiek [3], we extend the definition of confirmation bias within the context of software development and testing to include one or both of the following: (1) tendency to verify software code and (2) failure to apply strategies (e.g., logical reasoning, hypotheses testing skills) to try to break software code.

In order to measure confirmation bias levels of the participants, we administered a written test that is based on the experiment "Wason's Selection Task" and its variations. These experiments were originally proposed by cognitive psychologists to show the existence of confirmation bias among people. Further details about the written test can be found in [2]. In the written test, there are 7 thematic and 7 abstract questions. Abstract questions require pure logical reasoning, whereas thematic questions could be answered correctly by using daily life experience. We used two measures to evaluate the written test outcomes and hence to quantify participants' confirmation biases: S_{ABS} and S_{Th}. S_{ABS} is the ratio of the correctly answered abstract questions to the total number of abstract questions in the written test. Similarly S_{Th} is the score calculated for the thematic questions. S_{ABS} and S_{Th} take continuous values in the range [0, 1] (i.e., $0 \leq S_{ABS} \leq 1$ and $0 \leq S_{Th} \leq 1$). In the ideal case, all thematic and abstract questions should be answered correctly, hence the case "$S_{ABS} = 1$ and $S_{Th} = 1$" is an indication of low confirmation bias. Moreover, we employed Reich and Ruth's categorization scheme [4] and found that each participant belonged to one of the following categories based on his/her written test outcomes: *Falsifier*, *Verifier*, *Matcher* and *None*. A participant who is categorized as a *Falsifier* has the tendency to refute a hypothesis and he/she has the required skills and strategies to do that (i.e., Participants having low confirmation biases are categorized as *Falsifiers*). *Verifiers* also exhibit some logical reasoning strategies. However, they lack the tendency to refute a hypothesis. *Matchers* answer the questions in the written test by matching patterns, while participants belonging to category *None* answer the written test questions randomly.

At the end of the event, a judge committee consisting of three people assessed the resulting Apps (i.e. performance of development teams) using a grading scale in the range [0, 100]. The overall score of each team was estimated as the average of the grades given by all three judges. Before the main part of the hackathon began, written test was administered only to development teams, who volunteered to take part in this study. Therefore, this study includes 19 development teams consisting of 38 participants in total. Among the volunteer teams, 11 of them (23 participants) came up with a working software application, while the remaining 9 teams (15 participants) did not finish the hackathon. .

Table 1. Distribution of the Falsifying, Verifying and Matching Tendencies

Tendency	Attended Demo Session	
	Yes (category #1)	**No (category #2)**
Falsifiers	34.8 % (8)	13.3 % (2)
Verifiers	17.4 % (4)	20.0 % (3)
Matchers	8.7 % (2)	33.3 % (5)
None	39.1 % (9)	33.3 % (5)

Table 2. Distribution of the Falsifying, Verifying and Matching Tendencies

	Attended Demo Session	
Measure	**Yes (category #1)**	**No (category #2)**
S_{ABS}	0.55	0.11
S_{Th}	0.36	0.22

According to our analysis results, the development teams which contained at least one member having "Falsifying" tendency were given a score of 88.5 on average by the judge committee. This value turned out to be 72.3 for the remaining teams. During the demo session, the application software developed by the teams consisting of members with only "verifying" and/or "matching" tendencies, encountered execution failures. Table 1 gives the distribution of *Falsifiers*, *Verifiers*, *Matchers* and *None* among the teams, who attended the demo session as well as for those who did not. There are more *Falsifiers* among teams who attended the demo session. Moreover, as shown in Table 2, such teams performed better in the written test in terms of S_{ABS} and S_{Th}. Moreover, we performed a regression analysis on development team performance, with the scores given by judges as dependent variables and confirmation bias metrics values of development teams as independent variables. As shown in Figure 1, the variables S_{Th} and S_{Abs} have positive impact on team performance ($\alpha = 0.01$).

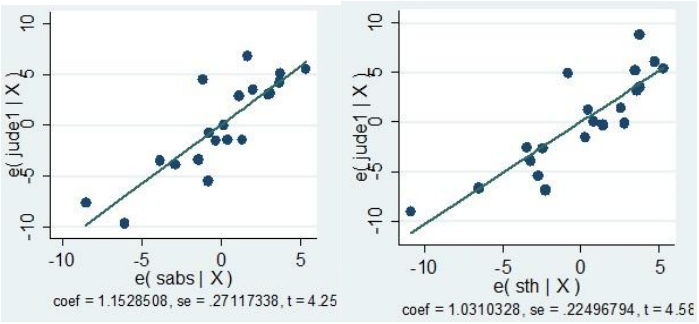

Figure 1. Correlation graphs: **(a)** S_{ABS} vs Judge Scores (average) **(b)** S_{Th} vs. Judge Scores (average)

In the long run, the outcomes of this study may shed light to understand the dynamics of project teams in software companies, especially when they have very short release periods. As future work, we plan to replicate this study in industrial settings.

References

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

[2] G. Calikli and A. Bener, "Influence of Confirmation Biases of Developers on Software Quality: An Empirical Study", (21)2:377-416, Software Quality Journal, 2013.

[3] Poletiek, F, . Hypothesis-testing behaviour. East Sussex, UK: Psychology Press, 2001.

[4] Reich, S., and Ruth, P. "Wason's selection task: Verification, falsification and matching. British Journal of Psychology", 73, 395–405, 1982.