Preliminary analysis of the effects of confirmation bias on software defect density

Preliminary Analysis of the Effects of Confirmation Bias on Software Defect Density

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

Ayse Bener
Ted Rogers School of Information Technology Management,
Ryerson University, Canada
0001 4169795297
ayse.bener@ryerson.ca

ABSTRACT
In cognitive psychology, confirmation bias is defined as the tendency of people to verify hypotheses rather than refuting them. During unit testing software developers should aim to fail their code. However, due to confirmation bias, most defects might be overlooked leading to an increase in software defect density. In this research, we empirically analyze the effect of confirmation bias of software developers on software defect density.

Categories and Subject Descriptors
H.1.2 [**User Machine Systems**]: Software psychology, Human factors. D.4.8 [**Software Engineering**]: Performance – measurements, modeling and prediction.

General Terms
Measurement, Human Factors

Keywords
Confirmation bias, software development, software defect density

1. INTRODUCTION
The term confirmation bias was first used by Peter Wason in his rule discovery experiment and later in his selection task experiment [1]. During all levels of software testing the attempt should be to *fail the code.* Therefore, during unit testing, developers must try to fail their code to reduce software defect density. While testing code, mere tendency to refute code does not help to detect defects. Hence, within the context of software development and testing, we extend the definition of confirmation bias to include one or both of the following: 1) *The tendency to verify software code, 2) The incompetency to apply strategies to try to fail software code.* In this research, we empirically analyzed the relation between confirmation bias of software developers and software defect density.

The rest of the paper is structured as follows: In Section II details of our case study design are mentioned. Results are given in Section III and Section IV concludes our work.

2. CASE STUDY DESIGN
In our case study, we analyzed a software development team in a large scale telecommunication company in Europe which is responsible from the development of a customer services software package. Based on the file commit history, we discovered that most of the files were created and/or updated by a group of one or more members of the software development team. As a result of churn data analysis, we found 124 such developer groups. We designed and prepared written and interactive tests, which are based on Wason's Selection Task and Wason's Rule Discovery Task respectively. From the outcomes of these tests, we extracted the values for the confirmation bias metrics. Except for Wason's eliminative enumerative index, the rest of the metrics have been defined by us. Detailed explanation about these metrics can be found in [2].

3. RESULTS
We defined *defect density* for each developer group as the ratio of the total number of defected files created/updated by that group to the total number of files that group created/updated. In order to visualize the effect of confirmation bias on software *defect density*, we constructed a linear regression model with confirmation bias metrics as the predictor (independent) variables and defect density as the response variable. The results indicate that 42.43% of variability in defect density can be explained by our linear regression model ($R^2_{adj} = 0.4243$). Moreover, we can expect the constructed model to explain only about 32% of the variability in predicting new observations ($R^2_{prediction} = 0.3264$).

4. CONCLUSIONS & FUTURE WORK
Results show that confirmation bias metric values of software developers are not direct indicators of software defect density. However, software defect density is also affected by process, product and many other human related attributes. Hence, the results obtained are quite significant. As future work, confirmation bias metrics will be used in addition to product and process metrics in learning based defect prediction systems. We believe that this will improve defect prediction performance of such systems.

5. REFERENCES
[1] Evans, J. St. B. T, Newstead, S. E. and Byrne, R. M. Human reasoning: The psychology of deduction (1993).

[2] Calikli, G., Bener, A., and Arslan, B. An analysis of the effects of company culture, education and experience on confirmation bias levels of software developers and testers, to appear in the Proc. of 32nd Int. Conference on Software Eng. (ICSE 2010).

ESEM'10, September 16–17, 2010, Bolzano-Bozen, Italy.