

# Paper Review CS846, Feb. 06

## *Gothas from mining bug reports*

### *and It's not a bug, it's a feature: How misclassification impacts bug prediction*

Wenhan Zhu (Cosmos)  
w65zhu@uwaterloo.ca  
University of Waterloo  
Waterloo, Canada

#### ABSTRACT

This week's chapter from the book being reviewed here is **Gothas from mining bug reports** by *Sascha Just and Kim Herzig* and **It's not a bug, it's a feature: How misclassification impacts bug prediction** by *Kim Herzig, Sascha Just and Andreas Zeller*

*Just et al.*'s chapter talks about bug reports in general, how they work, their problems and gave some examples of bug reports.

*Herzig et al.* talked about their discovery of bug reports are often misclassified and how the misclassification can lead to undesired results of studies on bug reports.

#### KEYWORDS

paper review

#### 1 SUMMARY

*Just et al.* starts the chapter by talking about an example of how real world bug fixes does not always reflect the three fundamental assumptions empirical software engineering research assumes. The assumptions are the location of the defect is the location of code change, issues correctly report bugs, bug fixing commits are atomic. In the real world, sometime the way to fix a bug is to change code somewhere else that will prevent the bug happening which make assumption 1 not true all times. The paper's result showed that assumption 2 is not true all the time. And the last assumption is not often true due to developers sometimes do multiple things in one commit and it's a challenge to determine which portion of the changed code correlate with the things done in one commit.

*Herzig et al.*'s paper invested more than 7000 real world issues reports to see if bug reports are accurate. Their findings indicate that every third bug report is misclassified and on average around 39% of files marked as defective didn't contain a bug. Two of the authors manually inspected the issue reports separately and compared their results to merge conflicts. They decided to classify issue reports in 6 categories, fix request (bug), feature request, improvement request, documentation request, refactoring request and other when things are not quite clear. Detailed investigations on these separate categories revealed different mean percentage and deviations of misclassification. They average around 40% with feature request being the one with most deviation around mean. A significant finding in this study is that a third of the bug reports are not actually bugs.

#### 2 THOUGHTS

The theme of this combo is issue reports in real world and how they don't behave what we expect they do.

The chapter gave a grand overview of the current state of evaluating issue reports in empirical software engineering. The 3 assumptions used previously is not necessarily true and empirical evaluations of the problems gave more insight on the status of the problem. I think the introduction of issue report system really helped developers to keep track of issues in software, however, the results showed that we still have a long way to go to have better management of the problems. I think the reason for the inconsistencies between the intent of the system and the current state we are facing is the lack of knowledge of the software world for the users of many programs. Although the chapter did not mention where the examples are taken from but I think that the issue for different software will be dramatically different due to the usage of the software. This is shown in the paper this week, the misclassification for different software is extremely different in some cases. One way I could think of how to address this problem is approach the problem from the issue submitters perspective. When we collect the information we want the people that gave the information the best convenience so that we won't miss anything. Then we can classify the information for developers and gave them what they want. The problem I see in this process is that there could be information lost in the process. But these outliers can always be manually identified if needed.

The paper gave a real evaluation of the problems in issue report systems. I consider it a "Myst buster" type of paper. The manually evaluation ensures that there are no automatic classification errors but the size of evaluation is quite small compared to large MSR studies. I find it quite interesting that in the paper they mentioned that auto classifying the categories have around the same success rate and the correct rate of classifications in issue systems. I wonder if newer improvement of classification have used more accurate data and improved. I think the trackers the author looked at is not representative enough since they didn't consider the differences between user's of the software. I would imagine the bug reports of a much developer specific software would have less "noise" compared to the one tested. But it remains unknown unless there are more studies on the topic.

### 3 RATINGS

I would rate *Just et al.*'s chapter 4.5/5. It gave the status of our knowledge on bug report systems and how they might not be true in more details.

I would rate *Herzig et al.*'s work a 3.5/5. It is a decent paper and gave insights on some common believes that turns out to be not quite true. However, I feel like there are way more to explore on the topic and some of the decision is not quite convincing to me.