2016 42th Euromicro Conference on Software Engineering and Advanced Applications

A Replication Study: Where and When Should Defects be Re-Assigned

Tamer Abdou
Suez Canal University
Ryerson University
Email: tamer.abdou@ryerson.ca

Ayse Bener
Data Science Lab
Ryerson University
Email: ayse.bener@ryerson.ca

***Abstract*—Software development organizations often need to balance productivity and sustainable profitability. To keep a balance, attention needs to be given as to where and when defects are re-assigned. This paper reports a replicated experiment comparing the prediction differences of re-assigned defects between proprietary and open-source software (OSS) projects. We pursue a quantitative approach based on logistic regression that relies on two OSS defect tracking datasets: namely, Mozilla and Eclipse. Here, we focus our attention to the reputation of software developers and where it affects the re-assigned defects. We explore the correlation between re-assigned defects and other related factors such as the reputation of the software developer. The replication of the original quantitative approach aims to verify the results of the original experiment on OSS defect datasets, rather than on proprietary ones.**

***Keywords*-Open Source Software; Predictive models; Software measurement; Data analysis; Software Maintenance.**

I. Introduction

The aim of this paper is to describe the replication of empirical research applying the guidelines highlighted for reporting experimental replications, proposed by Carver [1]. This paper reports the replication of the original empirical research [2] [3] developed to investigate the causes of reopening bugs. It is important to determine these factors in order to acquire recommendations to enhance the software maintenance process. As a follow-up from the original empirical study, we designed and conducted a controlled empirical study to evaluate the validity of some of their conclusions. The original study had a main goal of identifying how and when defects should be re-opened.

II. Information About the Original Study

The research questions from the original study are summarized in I. The experimental procedure in the original study was divided into three parts [2]: In the first part, the responses to most of the survey questions were studied in the context of the two versions of the Microsoft Windows operating system project. Response length varied from one phrase (e.g. "bug cause was not initially understood") to long paragraphs. All 358 responses were printed out on index cards, in order to sort them out. The process of sorting cards was conducted independently by two authors and then they merged their results into a single taxonomy.

In the second part, a random sampling technique was used and the sample size was 20 Windows Vista bug reports with reopens. In the third part, descriptive statistics along with correlation and regression was used and analysis was done with the use of the R statistical package.

Table I
Research Questions in the Original Study

Parts	Research Questions	Statistical Techniques	Dataset Sources
Part 1	RQ_1	Qualitative Survey	A Free Response Survey
Part 2	RQ_2	Manual Inspection Random Sampling Sorting Technique	Bug reports with reopens (Windows Vista)
Part 3	RQ_3	Quantitative Analysis Grouping Relative Ratios	Entire Windows bug database (Windows Vista & Windows 7
	RQ_4	Success Rates	
	RQ_5	Quantitative Analysis Relative Ratios	
	RQ_6	Logistic Regression Analysis of deviance R statistics Packages	

The values of the context variables were all taken from the Windows bug database and the Microsoft employee personnel database. The database is based on two main data sets: (i) The pre- and post-release bug reports for Windows Vista in July 2009. This means that data have been mined 2.5 years after Vista's release date. (ii) Bug reports for Windows 7, representing the development period (3 years).

III. Information about the Replication

The results of the original study revealed some significant relationships with factors impacting the process of reopening bugs. This allows for defining a strategy to confirm if these results are significant in OSS projects (Mozilla and Eclipse). To do this, we replicated the empirical design in the quantitative analysis part, see Table IV, and Table V. The purpose of replicating this part of the original study is to confirm the results of the original study and obtain robust conclusions on their findings. Like the original quantitative study in RQ_6, the replication was divided into three sub research categories: (i) the probability that a bug will be reopened, (ii) the probability that a bug will be fixed after the bug has been reopened, and (iii) the probability that a bug will be fixed.

A. Changes to the Original Experiment

The analysis demonstrated in this replication was based on data from the defect tracking database [4], which aims to

2376-9505/16 $31.00 © 2016 IEEE
DOI 10.1109/SEAA.2016.57

collect data from two open-source software (OSS) projects, namely Mozilla and Eclipse.

In general, the replication is quite similar to the original study. The research questions, factors, and design procedures are all kept the same, except the following:

- We have run the replication with a dataset that has different characteristics than the dataset of the original experiment.
- The first two and the fifth research questions have been omitted. These research questions are the ones related to the qualitative study, manual inspection, and geographical distance. However, the responses of the survey have been considered through replicating the quantitative part, specifically during the selection of the factors.
- The datasets [4] in this replication study do not include enough information about the sources of reopened bugs. We substituted these categories with two categories in which reopened bugs frequently occurred.
- The factors of relevance to organizational and geographic distance are omitted, as there is inadequate information considering the geographic locations of project members.
- Two additional factors have been included in our model during the quantitative part. Since the change in priority level was one of the reasons for re-opening bugs, see Section II, "Level of priority" and "Priority changed?" factors have been included.

IV. Comparison of Replication Results to Original Results

Our goal is to determine whether the environment and factors of the original study that were changed in the replication are likely to be the reasons of having differences in the results [1]. The following sub-sections illustrate the results of the replication after applying the statistical techniques similar to the ones in the original study. Then, we compared theses results with the outcomes of the original study for each of the research questions.

A. Comparing the results of RQ_3

RQ_3 Does the source of a bug influence the likelihood of bug reopens?

Table II reports the results of the analysis of where a bug was found, using two different categories, (i) High-rate bug-likelihood components, and (ii) Low-rate bug-likelihood components.

1) Consistent Results: Similar to the original study, we report the reopen ratios relative to the baseline percentages P, Q and R, which are the reopen rates for all bugs in Windows XP, Linux, and MAC OS X respectively. The original study reported the ratios relative to the reopen rates for all bugs in Windows Vista and Windows 7.

Table II
The influence of the bug source on bug reopens Replication Study

	Bug-likelihood Components	Windows XP	Linux	Mac OS X
	Reopen rate for all bugs	P	Q	R
Mozilla	High-rate	0.58.P	0.49.Q	0.57.R
	Low-rate	0.64.P	0.71.Q	0.57.R
Eclipse	High-rate	0.56.P	0.45.Q	0.47.R
	Low-rate	0.63.P	0.77.P	0.76.R

Table III
The influence of the bug source on bug reopens Original Study

Bug Source	Windows Vista	Windows 7
Reopen rate for all bugs	P	Q
Code analysis tools	0.52.P	0.73.Q
Human review	0.85.P	0.66.Q
Ad-hoc testing	0.87.P	0.99.Q
Internal user	1.12.P	0.97.Q
Component testing	1.13.P	0.81.Q
System testing	1.21.P	1.46.Q
Customer	1.33.P	1.12.Q

In the replication study, as shown in Table II, it has been observed that bugs found in High-rate bug-likelihood components in the Mozilla project are less likely to be reopened ($0.58 \sim 0.49 \sim 0.57$ times for High-rate bug-likelihood components and $0.64 \sim 0.71 \sim 0.57$ times for Low-rate bug-likelihood components). Similar results have been observed in the Eclipse dataset as well. Possible reasons might be that some bugs found in less bug prone components are easy to fix due to the absence of bug dependencies. Another possible reason might be that module owners tend to re-investigate testing techniques in less bug-prone areas in order to enable them in high bug-prone areas.

2) Differences in Results: Instead of defining bug source as "How was this bug found?" e.g., by a customer, an internal Microsoft user, or a system test, in this study, we define it as "Where was this bug found?" e.g. in High-rate bug-likelihood components, or in Low-rate bug-likelihood components.

Our definition is based on defect densities of components, with median as a statistical threshold. This is due to inadequate information on how bugs were found in our datasets. Table III reports the results of the analysis of how bugs were found in the original study, using seven different categories.

B. Comparing the results of RQ_4

RQ_4 : Does the reputation of the opener and first assignee Influence the likelihood of bug reopens?

1) Consistent Results: In the replication study, we observed that bugs opened by users who have been highly successful or unsuccessful in getting their bugs fixed (achieved high or low reputation respectively [5]) are less likely to be reopened, as shown in Figure 1. Additionally, bugs opened by first-time users with a lower reputation are less likely to

be reopened, as shown in Figure 3. This observation matches the findings in the original study, as shown in Figure 2, and Figure 4.

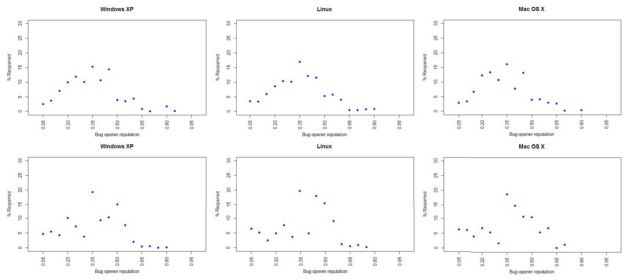

Figure 1. Percent of reopened Windows XP (left), Linux (middle), and Mac OS X bugs (right) vs. bug opener reputation (rounded up to nearest 0.05). (Replication Study)

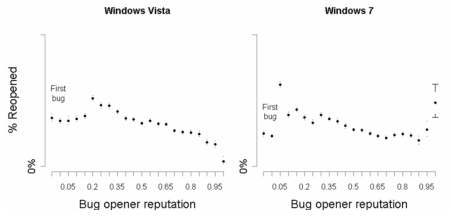

Figure 2. Percent of reopened Windows Vista (left) and Windows 7 bugs (right) vs. bug opener reputation (rounded up to nearest 0.05). (Original Study)

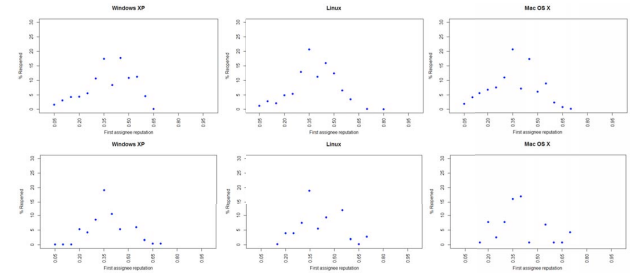

Figure 3. Percent of reopened Windows XP (left), Linux (middle), and Mac OS X bugs (right) vs. first assignee reputation (rounded up to nearest 0.05). (Replication Study)

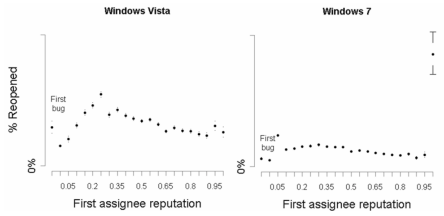

Figure 4. Percent of reopened Windows Vista (left) and Windows 7 bugs (right) vs. first assignee reputation (rounded up to nearest 0.05). (Original Study)

2) Differences in Results: By looking at the bug reports in the Mozilla dataset, Figure 1, we will notice a consistent increase in bug-reopening likelihood as the opener reputation increases, and then a consistent decrease till forming the bell shape. However, looking at the bug reports in Windows Vista from the original study, Figure 2, we notice a consistent decrease in bug-reopening likelihood with the increase of the opener reputation. A possible reason might be that, in an OSS project, most new users do not have enough information about the OSS project and the process of reporting a bug due to inadequate documentation in the OSS development process.

C. Comparing the results of RQ_6

RQ_6 : What are the influences on bug reopens?

Table IV and Table V report the results of the analysis of the influences on reopened, reopened-after-fixed, and fixed bugs on the Mozilla and Eclipse datasets respectively. Although, the original study referred to R as the statistical method used in analyzing the data, there was inadequate information on the details of packages used in the quantitative analysis. In this replication study, the data analysis was conducted using `R` v.3.2.3 [6]. The logistic regression models were built using the package GLMULTI [7].

The coefficients of the selected best models are listed in Table IV and Table V. The original study tested whether each coefficient of the independent factors was statistically significant at $p < 0.001$. We use the same significance rate in our replication study, and we marked coefficients which are statistically significant at $p < 0.01$ with "$*$", Table IV, and Table V. There were insufficient details in the original study on measuring the performance of the regression models. In our replication study, the quality of separation represented by the area under the curve (AUC), and the extent of recall and precision balance represented by F.measure are measured for each best-fitted model.

1) Consistent Results:

Interpreting the Descriptive Mode "Reopen": The descriptive mode has been interpreted It has been observed that there is a positive relationship with reopened bugs, as indicated by the corresponding coefficients, which are positive: (i) whether the re-opener of the bug was a temporary employee (Mac OS X), which matches the outcome of the original study for Windows Vista, (ii) the initial severity of the reopened bug (Windows XP), which matches the results for both versions of Windows in the original study, and (iii) whether severity was upgraded (Mac OS X), which matches the results for Windows Vista, as shown in Table VI. Similarly, the descriptive mode ""Fixed-when-Reopened", and the descriptive Mode "Fixed" can be interpreted.

2) Differences in Results:

Interpreting the Descriptive Mode "Reopen": Our reported results showed that there is a positive correlation between the reputation of the opener and the reopened

bugs for all datasets, and for all operating systems. These results are inconsistent with the outcomes of the original study which shows a negative correlation between this factor and reopened bugs. Moreover, the reputation of the first assignee in the replication study showed a positive relationship with reopened bugs, as opposed to the results in the original study, which showed a negative correlation. The replication study showed that the number of times a bug got reopened was positively correlated with the probability that a bug get reopened. Similarly, the descriptive mode ""Fixed-when-Reopened", and the descriptive Mode "Fixed" can be interpreted.

V. Conclusions Across Studies

We have conducted an empirical validation of factors in order to characterize where and when a bug can be reassigned. In this paper, our objectives are to (i) apply the guidelines for reporting a replication experiment, and (ii) analyze our findings for drawing cross study conclusions for both the original and replication studies. On the basis of findings obtained from both studies, we can make a number of general recommendations for improving the process of reassigning bugs. We analyze and highlight conclusions about the relationships between different factors in the maintenance process and re-assigned bugs. Additionally, the original and replication studies each used a different dataset. Therefore, combining results will be useful to generalize evidences and draw conclusions. Although this study is a replication and our findings provide guidance for future research to understand causes of reassigning bugs, further validations are required with different techniques.

Acknowledgment

This study is supported in part by NDG 402003-740 2012.

References

[1] J. C. Carver, "Towards Reporting Guidelines for Experimental Replications: A Proposal," in *1st International Workshop on Replication in Empirical Software Engineering*, Cape Town, South Africa, 2010.

[2] T. Zimmermann, N. Nagappan, P. J. Guo, and B. Murphy, "Characterizing and Predicting Which Bugs Get Reopened," in *Proceedings of the 34th International Conference on Software Engineering*, ser. ICSE '12. Piscataway, NJ, USA: IEEE Press, 2012, pp. 1074–1083.

[3] P. J. Guo, T. Zimmermann, N. Nagappan, and B. Murphy, "Characterizing and Predicting Which Bugs Get Fixed: An Empirical Study of Microsoft Windows," in *Proceedings of the 32th International Conference on Software Engineering (ICSE)*. Association for Computing Machinery, Inc., may 2010.

[4] A. Lamkanfi, J. Pérez, and S. Demeyer, "The Eclipse and Mozilla Defect Tracking Dataset: A Genuine Dataset for Mining Bug Information," in *Proceedings of the 10th Working Conference on Mining Software Repositories*, ser. MSR '13. Piscataway, NJ, USA: IEEE Press, 2013, pp. 203–206.

Table IV
Descriptive logistic regression models for (i) bug reopen rate, (ii) bug-fix probability for re-opened bugs, and (iii) bug-fix probability for all bugs (Replication Study on Mozilla Project)

Independent Factors		Mozilla Project								
		Coefficients (Windows XP)			Coefficients (Linux)			Coefficients (MAC OS X)		
Global	Specific	Reopen	Fixed When Reopened	Fixed	Reopen	Fixed When Reopened	Fixed	Reopen	Fixed When Reopened	Fixed
Bug Source	High bug-likelihood	♣	♣	♣	♣	♣	♣	♣	♣	♣
	Low bug-likelihood	n.s.	n.s.	0.038*	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
Reputation	Bug opener	0.174	0.158	0.052	0.170	0.151	0.0452	0.133	0.104	n.s.
	First assignee	n.s.	0.011	0.193	0.004*	0.014	0.134	n.s.	0.006*	0.163
	Temp. employee?	n.s	n.s.	-0.024	n.s	n.s	-0.040	n.s	n.s	-0.017*
Severity	Initial severity	0.003*	n.s.	0.057	n.s	0.007*	0.041	n.s	n.s.	0.063
	Severity changed?	n.s	-0.006*	-0.049	n.s	n.s	n.s	0.006	n.s	-0.051
Priority	Initial priority	n.s	n.s.	n.s.	n.s	n.s	-0.027	n.s	n.s	n.s
	Priority changed?	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
Number of	Components	n.s.	n.s.	0.018*	0.007	n.s.	-0.046	-0.005*	n.s	-0.030
	Reopenings	0.188	0.146	n.s.	0.158	0.107	0.020*	0.227	0.208	0.075
AUC		0.99	0.99	0.77	0.99	0.98	0.69	0.99	0.98	0.77
F.measure		0.20	0.19	0.75	0.26	0.28	0.74	0.19	0.20	0.80

Table V
Descriptive logistic regression models for (i) bug reopen rate, (ii) bug-fix probability for re-opened bugs, and (iii) bug-fix probability for all bugs (Replication Study on Eclipse Project)

Independent Factors		Eclipse Project								
		Coefficients (Windows XP)			Coefficients (Linux)			Coefficients (MAC OS X)		
Global	Specific	Reopen	Fixed When Reopened	Fixed	Reopen	Fixed When Reopened	Fixed	Reopen	Fixed When Reopened	Fixed
Bug Source	High bug-likelihood	♣	♣	♣	♣	♣	♣	♣	♣	♣
	Low bug-likelihood	-0.025	0.015*	0.095	n.s.	n.s.	n.s.	-0.039*	n.s.	n.s.
Reputation	Bug opener	0.125	0.108	0.041	1.081	1.276	0.721	0.136	0.147	n.s.
	First assignee	0.010	n.s.	-0.021	n.s.	n.s.	n.s.	n.s.	n.s.	-0.075*
	Temp. employee?	n.s.	0.016	0.058	n.s.	0.11	0.301	0.025	n.s.	0.067*
Severity	Initial severity	n.s.	0.023	0.125	n.s.	n.s.	0.039*	n.s.	n.s.	0.108
	Severity changed?	n.s.	n.s.	n.s.	n.s.	-0.023	n.s.	n.s.	n.s.	n.s.
Priority	Initial priority	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
	Priority changed?	n/a	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
Number of	Components	-0.005*	0.008	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.	n.s.
	Reopenings	0.276	0.117	n.s.	0.485	0.136	n.s	0.283	0.097	n.s.
AUC		0.99	0.96	0.67	0.99	0.97	0.70	0.99	0.97	0.70
F.measure		0.34	0.22	0.68	0.35	0.23	0.63	0.34	0.24	0.74

Table VI
Descriptive logistic regression models for (i) bug reopen rate, (ii) bug-fix probability for re-opened bugs, and (iii) bug-fix probability for all bugs (Original Study on Microsoft Project)

Independent Factors		Coefficients (Windows Vista)			CoefficientsWindows 7		
Global	Specific	Reopen	Fixed When Reopened	Fixed	Reopen	Fixed When Reopened	Fixed
Bug Source	Human review	n.s.	0.377	0.511	-0.343	0.529	0.770
	Code analysis tool	-0.503	n.s.	0.357	-0.291	0.884	0.349
	Component testing	0.238	-0.160	0.065	-0.116	0.406	0.488
	Ad-hoc testing	♣	♣	♣	♣	♣	♣
	System testing	0.204	n.s.	-0.129	0.182	-0.342	-0.040
	Customer	0.239	-0.498	-0.347	-0.466	-0.511	-0.427
	Internal user	n.s.	-0.465	-0.454	-0.611	-0.398	-0.723
Reputation	Bug opener	-0.266	1.632	2.193	-0.948	1.601	2.480
	First assignee	n.s.	1.651	2.463	-0.697	1.589	2.407
	Temp. employee?	0.178	-0.144	-0.125	n.s.	-0.403	-0.260
Severity	Initial severity	0.127	n.s.	0.033	0.081	0.383	0.202
	Severity changed?	0.331	n.s.	0.256	n.s.	0.463	0.300
Geographical	Same manager?	0.721	n.s.	0.676	0.149	n.s.	n.s.
	Same building?	0.468	n.s.	0.270	0.376	n.s.	0.493
Number of	Editors	0.236	0.127	0.240	0.236	0.125	0.289
	Assignee buildings	0.090	-0.213	-0.257	0.101	-0.111	-0.145
	Component path changes	-0.160	-0.162	-0.232	-0.053	-0.135	-0.214
	Reopens	n/a	n/a	-0.135	n/a	n/a	0.024

[5] P. Hooimeijer and W. Weimer, "Modeling Bug Report Quality," in *Proceedings of the Twenty-second IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '07. New York, NY, USA: ACM, 2007, pp. 34–43.

[6] R Core Team, "R: A Language and Environment for Statistical Computing," Vienna, Austria, 2013.

[7] V. Calcagno and C. D. Mazancourt, "glmulti : An R Package for Easy Automated Model Selection with (Generalized) Linear Models," *Journal of statistical software*, vol. 34, no. 12, pp. 1–29, 2010.