Wei Le <weile.cs@gmail.com>

ICSM 2012 author response (paper 136)

1 message

ICSM 2012 <icsm2012@easychair.org> Wed, May 30, 2012 at 10:36 AM
To: Wei Le <wei.le@rit.edu>

Dear Wei Le and Daniel Krutz,

Thank you for submitting the paper

How to Group Crashes Effectively: Comparing Manually and Automatically Grouped Crash Dumps

to ICSM 2012. The ICSM 2012 review response period will be between May 30, 2012 and June 4, 2012, 23.59.59 Howland Island Time http://www.worldtimezone.com/time/wtzresult.php?CiID=42242

During this time, you will have access to the current state of your reviews and have the opportunity to submit a response of up to 500 words. Please keep in mind the following during this process:

* The response must focus on any factual errors in the reviews and any questions posed by the reviewers. It must not provide new research results or reformulate the presentation. Try to be as concise and to the point as possible.

* The review response period is an opportunity to react to the reviews, but not a requirement to do so. Thus, if you feel the reviews are accurate and the reviewers have not asked any questions, then you should not respond.

* The reviews are as submitted by the PC members, without any coordination between them. Thus, there may be inconsistencies. Furthermore, these are not the final versions of the reviews. The reviews will be updated to take into account the online discussion happening after the rebuttal phase. Also, we may find it necessary to solicit other outside reviews after the review response period if we realize that extra expertise is needed on some papers.

* The program committee will read your responses carefully and take this information into account during the discussions. On the other hand, the program committee will not directly respond to your responses, either before the program committee meeting or in the final versions of the reviews.

* Your response will be seen by all PC members who have access to the discussion of your paper, so please try to be polite and constructive.

The reviews on your paper are attached to this letter. To submit your response you should log on the EasyChair Web site for ICSM 2012 and select your submission on the menu.

Best wishes,

Massimiliano Di Penta and Jonathan I. Maletic
ICSM 2012 Program co-chairs

---------- REVIEW 1 ----------
PAPER: 136
TITLE: How to Group Crashes Effectively: Comparing Manually and Automatically Grouped Crash Dumps
AUTHORS: (anonymous)

REVIEWER'S COMMENTS:

This paper describes an empirical study of crash reports, with particular attention to comparing manual and automatic groupings of crash reports. The goal is to learn how to inform the development of better tools for grouping crash reports by studying criteria used to group them, information used for grouping, imprecision of current approaches, and characteristics of the call stacks in different groupings done manually versus automatically. Bugzilla reports were analyzed manually (about 450 of them) while Mozilla crash reporter information was analyzed for the automatic system. A set of conclusions is drawn about grouping crash reports -- we need to design grouping critera that enables better uses of group information for diagnosing failures, to correlate multiple sourcse of inforamtion and connect related apps, and establish precise relations between symptoms and code.

The study overall is an interesting study of automatic and manual grouping of crash reports.
I have concerns about the methodology used for each of the 4 aspects studied.
For grouping criteria, the authors define a set of keywords and count the frequency of those keywords in bugzilla entries. How did you develop those keywords? How do you justify that frequency of using those words is a good estimate? It seems some kind of manual analysis and accuracy computation should be done to justify this as a good heuristic before using it as the measure in the study and depending on it being accurate.

For determining imprecision, a list of several properties such as a corrupted signature or call stack,... is used to determine imprecision. Again, how do you justify these intuitively and how accurate did you find them. The heuristic needs to be judged to be a good estimator before using it in the study and depending on it being a good estimator.

For comparing call stack characteristics, what is the use of this information and why choose the information that you did? There is no intuition given for looking at these characteristics and what you want to learn from them.

There are a lot of results provided, but without more explanation and justification of the measures used, the results are without solid backing.

As I was reading, the paper seemed to have figures and definitions out of order, being presented after the terminology had already been used and the information already given in text form. This is particularly the case in the first few pages where the paper needs less repetition and more clear organization where terms are defined before used, and overview figures are given earlier as the overview is first presented instead of later and repeated.

The authors go into minute detail about the results. The bullets at the start of each result section just after the research question is presented seemed out of order. I could not figure out where these claims or conclusions were coming from. The data should be presented first and then big picture stories developed, or be sure to introduce the big stories with a lead in sentence to make it clear what the bulleted items are meant to be and what data led to them.

The graphs should have units on the axes to make them standalone without having to read the text.

Reference 4 is missing part of it.

Overall, an interesting study but readers need more sense of the accuracy and justification for the measures before being able to appreciate the results.

QUESTIONS FOR THE AUTHORS:

---------- REVIEW 2 ----------
PAPER: 136
TITLE: How to Group Crashes Effectively: Comparing Manually and Automatically Grouped Crash Dumps
AUTHORS: (anonymous)

Nowadays, many companies rely on automatic crash reporting tools to collect crash reports directly from users' environment. The crash reports are later grouped based on the similarity of the call stacks they contain. This paper investigates the accuracy of automatic grouping approaches. Through a qualitative analysis of call stacks contained in bug reports, the paper proposes a list of criteria to

improve the grouping of crash reports. A total number of 1.550 groups are investigated. The groups are collected from 5 Mozilla applications; Firefox 14.0a1, Thunderbird 10.0, Fennec 2.1.2, Fennec Android 13.0a1, SeaMonkey 2.7.2. Each group is mapped to a list of dependent bugs and the call stacks contained in the bug reports are compared using four similarity metrics.

+The topic of the paper is interesting. Improving the accuracy of crash reports grouping approaches is very important to provide developers with relevant information to fix the bugs. Indeed, previous studies have observed that if the crash reports caused by multiple bugs are grouped together it takes longer time to fix the bugs. However, this paper does not provide any concrete method to improve automatic grouping approaches.

-When I first read about the analysis of crash dumps extracted from bug reports, I was expecting some concrete rules that can be used to improve the automatic grouping of crash reports. Instead, the authors only speculated on the potential motivations behind developers selections of the crash dumps.

-The data collection process should be elaborate further. It is very unclear how the authors extracted information from Bugzilla. How were the key words selected and used? How accurate was this process? The authors mention a manual validation of automatic groups, but no details is provided on the number of subjects involved in the validation and the number of crash reports inspected.

-Also, many opinions in the paper are attributed to developers but no details are provided on how the opinions were collected. The difference between facts and authors' opinion is not clear. For example, the authors' claim repeatedly that "developers group crash dumps" while developers actually do not group crash dumps but instead select crash dumps to fix bugs. "Manual groups" of crash dumps are actually created by the author's; therefore attributing the groups to developers is wrong.

-Information contained in Table II is puzzling. The authors are able to relate 20 groups of crash reports to 110 bugs in Bugzilla, this high number of opened bug reports is very surprising.

-The authors should explain how the "Time" reported in Table IV was computed. It is not clear how the time period between the introduction of a mistake in Bugzilla and the identification of the mistake was tracked.

-The developers discussion logs mentioned in the results section are not mentioned in the data collection.

-It would have been nice if data were provided in the paper to backup the many claims made by the authors throughout the paper… For each of the criteria reported in Table III, the authors should also report the proportion of crash dumps concerned by the criteria.

-The authors should discuss the following related work:

@inproceedings{Glerum:2009:DLT:1629575.1629586,
author = {Glerum, Kirk and Kinshumann, Kinshuman and Greenberg, Steve and Aul, Gabriel and Orgovan, Vince and

Nichols, Greg and Grant, David and Loihle, Gretchen and Hunt, Galen},
title = {Debugging in the (very) large: ten years of implementation and experience},
booktitle = {Proceedings of the ACM SIGOPS 22nd symposium on Operating systems principles},
series = {SOSP '09},
year = {2009},
isbn = {978-1-60558-752-3},
location = {Big Sky, Montana, USA},
pages = {103--116},
numpages = {14},
url = {http://doi.acm.org/10.1145/1629575.1629586},
doi = {http://doi.acm.org/10.1145/1629575.1629586},
acmid = {1629586},
publisher = {ACM},
address = {New York, NY, USA},
keywords = {blue screen of death, bucketing, classifying, error reports, labeling, minidump, statistics-based debugging.},

-The paper should be proof read to remove grammatical errors.

e.g. “RQ1: Why do we group crash dumps?” ->How can we group crash dumps?

---------- REVIEW 3 ----------
PAPER: 136
TITLE: How to Group Crashes Effectively: Comparing Manually and Automatically Grouped Crash Dumps
AUTHORS: (anonymous)

REVIEWER'S COMMENTS:

The authors conducted a study on investigating how to group crashes effectively comparing manual and automatically grouped crashes.

Overall the problem targeted by the authors is a worthwhile and interesting one. Much of previous related work focuses on providing tool support for clustering/grouping crash dumps rather than providing in-depth comparative analysis of practices of manual and automatic approaches, which is the focus of this paper.

The study findings are valuable for developers/researchers of future tools for grouping crash dumps.

The title is suggested to change to reflect the research work more faithfully. First, the title needs to indicate that the work is an empirical study. Therefore, "How to Group Crashes Effectively" may be left out because the study findings don't have direct answers to this question. Second, "Comparing Manually and Automatically grouped Crash Dumps" may not be accurate: the study compares different ways of grouping crash dumps rather than comparing crash dumps.

The authors should include a project web URL to include more detailed information on how the (textual) information from different sources is interpreted by the authors. So far the authors simply list what they found/interpreted without showing the detailed process/procedure to allow others to reproduce what they found.

A lot of the interpretations of developers' behaviors or decisions

for m-grouping are based on the authors' interpretations of bug reports, emails, ... What if the info documented in these information sources is not complete? Confirmation of developers on the findings and interpretations would be desirable to improve the trustworthy of the findings.

Claim 4 (small application dump analysis is more effective manually than automatically) seems to be lacking sufficient evidence. The current justification for the claim is that the smallest application, SeaMonkey, was most effectively analyzed manually. The correlation is shown. But causation is not shown. In other words, the results may not be generalizable to other small-sized applications.

It would make the findings more interesting and more valuable if the authors can discuss the possibility of applying their findings to build up a human-assisting tool that can incorporate more knowledge of developers to help a-grouping. The reason is that a-grouping is the direction since manual grouping is not scalable.

The paper's writing could be further improved. For example "Since a same bug" should be "Since the same bug"; "Since same symptoms are" should be "Since the same symptoms are".

The following two recent papers are related for the authors to look into:

Shi Han, Yingnong Dang, Song Ge, Dongmei Zhang, and Tao Xie, Performance Debugging in the Large via Mining Millions of Stack Traces, in Proceedings of the 34th International Conference on Software Engineering (ICSE 2012), Zurich, Switzerland, June 2012.

Yingnong Dang, Rongxin Wu, Hongyu Zhang, Dongmei Zhang and Peter Nobel, ReBucket C A Method for Clustering Duplicate Crash Reports based on Call Stack Similarity, in Proceedings of the 34th International Conference on Software Engineering (ICSE 2012), Software Engineering in Practice, Zurich, Switzerland, June 2012.

QUESTIONS FOR THE AUTHORS: