

Goal Document

Master's Thesis - Fault localization for automated testing

Stakeholders

Axel Hildingson, Student
a.hildingson@gmail.com, 073-5810289

Jonas Svalin, Student
jonas.svalin@gmail.com, 073-6458132

Lars Andersson, Supervisor, Qlik
lars.andersson@qlik.com

Emilie Engström, Supervisor, LTH
emilie.engstrom@cs.lth.se

Per Runeson, Examiner, LTH
per.runeson@cs.lth.se

Background

Today, Qlik have a working automated test framework for their system, courtesy of a previous Master's Thesis project performed at the company. While the automated tests work well in identifying and clustering related issues with the system, they do not provide any assistance in locating the source of the issue. Qlik have seen that many of the test failures that occur can be traced back to the same commit, but locating the commit responsible is currently done manually and is thus a slow and tedious process. What Qlik is requesting assistance with is investigating the possibility of building an extension of the current testing system that can identify potential fault suspects.

Project Description

The purpose of this Master's Thesis project is to investigate whether it is possible to extract the code run during a test (by means of code coverage) and the code from a series of commits to Git in a manner where they can be compared. The comparison would then be used to try to locate potential causes of the test failures, by both identifying specific deliveries as well as the lines of code affected by that delivery. Hopes are that this will increase efficiency when investigating issues, which is also the criteria of success that will be used when evaluating the project.

Method

This project will follow the pattern of problem solving action research, meaning that a problem has already been identified and we intend to investigate whether it can be solved by the processes outlined in the following section. Background for the project consists of related work on automated testing, fault-localization and debugging that will be analyzed during the literature review. The results of the project will include an evaluation of the prototype, based on increased efficiency and user feedback when investigating test failures.

Timeline

The project is planned to run for 20 study weeks, spread out from the 19th of January to the 21st of June. At the end, the intent is to finalize a report that summarizes the project and

presents the results, hopefully accompanied by a working prototype. During these weeks the project will be divided into three essential parts where different tasks are planned. During the first phase a literature review will be performed where we acquire deeper knowledge of related work in the field. During the second phase, focus is on implementing a prototype and finally in the third phase the results will be summarized and presented.

Planned Results

The finalized report will present related work in the field and the results from our project. Fault localization is currently one of the most time consuming processes in software development and an automated tool for fault localization that would cut down on effort and time expenditures would be of great value. Hopefully the report will be accompanied by a working prototype that fulfills these goals and expectations.

Related work

James A. Jones, Mary Jean Harrold, and John Stasko. 2002. Visualization of test information to assist fault localization. In *Proceedings of the 24th International Conference on Software Engineering* (ICSE '02). ACM, New York, NY, USA, 467-477. DOI=10.1145/581339.581397 <http://doi.acm.org/10.1145/581339.581397>

James A. Jones and Mary Jean Harrold. 2005. Empirical evaluation of the tarantula automatic fault-localization technique. In *Proceedings of the 20th IEEE/ACM international Conference on Automated software engineering* (ASE '05). ACM, New York, NY, USA, 273-282. DOI=10.1145/1101908.1101949 <http://doi.acm.org/10.1145/1101908.1101949>

Artzi, S. IBM Software Group, Littleton, MA, USA Dolby, J. ; Tip, F. ; Pistoia, M. 2011. Fault Localization for Dynamic Web Applications. In *IEEE Transactions on Software Engineering* (vol 38 , issue 2, page 314-335). DOI=10.1109/TSE.2011.76 <http://ieeexplore.ieee.org.ludwig.lub.lu.se/xpl/articleDetails.jsp?arnumber=5975173>