

RESEARCH STATEMENT

LUCIA

School of Information System, Singapore Management University

Lucia.2009@phdis.smu.edu.sg

Research Interest

Software Systems

In software development and maintenance, bug is one of the issues that could diminish the benefit of the company that employs the software, as well as the company that develops the software. US National Institute of Standards and Technology in 2002 has stated that software bugs cost US economy 59.5 billion dollars annually (Tasey,2002).

Finding the root cause of the bugs is a challenging and expensive activity. Finding the code that responsible for a bug is non-trivial especially if the code that responsible for a bug could be far away from the failure point. It is reported that testing and debugging contribute 30 to 90 percent of the labor expended for a project (Beizer, 1990). It will be very helpful if there is an automated debugging tool that could help developers to find the root cause of the bug accurately or even much better if the tool could also help developers to fix the bug. Therefore, the cost of software debugging could be reduced.

It is my interest to contribute especially in software debugging so that the technique could report the root cause of bugs with good accuracy and within inexpensive time. The accuracy of locating the root cause of the bug could be increased if we could identify the bug characteristics. However, identifying bug characteristics is a challenging task since the patterns of bugs are hard to identify. In data mining and information retrieval domain, many techniques could be potentially incorporated and extended to solve software engineering problem domain. Utilizing these domains into software engineering domain is also my research interest.

Current Research

Bug Finding

In bug finding problem, certain information is often needed to locate the root cause of the bug. Based on the information needed to locate the bug, bug finding problem could be categorized into two research problems. The first research problem works on a technique to find the code that responsible for bug, where the information about which test cases that cause program behaves different from the requirement (fail test cases), are available. Hence, the test cases and the oracle to decide whether a test case is pass or fail, are required. There have been many techniques proposed to solve the problem. Some techniques could locate bugs accurately, but they often require expensive time and domain specific knowledge about the bugs. The other techniques could locate bugs in a short time and do not require domain specific knowledge about the bugs, but they only have a good enough accuracy. Thus, how can we improve the accuracy of the technique to find the root cause of the bug within a reasonably short time? How do we improve the applicability of the technique in term type of bugs that could be covered by the technique, in term of different type of programs e.g., in term of the intensity of algorithm in the program, etc., in term of different programming languages?

The second research problem works on a technique to detect code that potentially produce bugs for a given program without having any information fail test cases. The technique usually works based on certain programming rules or bug knowledge in order to detect the bugs. Due to less information involve about which program execution that is fail, the technique could produce many false positive bug reports. It is also a challenging problem to reduce the false positive rate of bug reports produced by the technique without removing the true positive bug reports.

Bug Fixing

Besides finding the root cause of a bug, it is also an interesting research problem to propose a technique that could automatically fix a bug in a program. There have been researches on this problem. Some proposed techniques only produce suggestions of the bug fixes and leave the fixing to developers. Other techniques could both suggest and automatically replaced the buggy code with the suggestion. Generally, the proposed bug fixing techniques are usually specific to certain type of bugs. Hence, there is a need to improve the applicability of bug fixing techniques so that it can handle more bugs.

References

B. Beizer, Software Testing Techniques, 2nd ed. Boston:International Thomson Computer Press, 1990.

G.Tassey, "The economic impacts of inadequate infrastructure for software testing." National Institute of Standards and Technology. Planning Report 02-3. 2002, 2002.