Team L

HW1: Paper from 2012

i) Reference:

D. Amalfitano, A.R. Fasolino, P. Tramontana, S. De Carmine and A. M. Memon

Using GUI Ripping for Automated Testing of Android Applications

Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE)

ii) Keywords:

1) **Testing Tools**: Software that allows developers to test code more efficiently. Testing tools attempt to make testing easier, more accurate and more robust than testing manually.
2) **Testing Automation**: Wide category of tools used to test software in a way that is easily repeatable and efficient. Encompasses many forms of tests including unit test, end to end tests, and GUI tests.
3) **Android**: Extremely popular operating system mostly used for mobile devices. Apps for android devices are written in Java, but differ in structure from normal Java applications. Reliant on GUI for input and output.
4) **Code Coverage**: A metric used to describe the thoroughness of tests. Described by the number of lines executed during testing over the total number of lines of code.

iii) Paper Contents:

1) **1. Motivational Statements**: The authors of this paper are trying to create a tool that allows for better, more efficient testing of android apps. They realize that at this time (2012) the Android market is growing rapidly. Efficient testing will ensure easier development and less buggy apps.
2) **6. Statistical tests**: The paper briefly goes into how to compute code coverage. Results about code coverage is used in the tables at the end of the paper.
3) **9. Baseline Results**: Monkey (the testing framework that comes with android), encountered one bug and three crashes while testing the "WordPress" app. It also had code coverage of 25%.
4) **14. New Results**: AndroidRipper (the testing service created by the authors of the paper) performed better than Monkey. Both ran and completed testing using around same amount of time, but AndroidRipper encountered about three times more bugs and two times more crashes. AndroidRipper also had more code coverage for most of its tests.

iv) Needs Improvement

1) Could have used more than one App to test both testing suites against. This would help determine if AndroidRipper is consonantly better than Monkey.
2) Could have elaborated a bit more while discussing phrases pertaining to software testing.
3) Could have added a key describing what B1-4 and R1-3 were supposed to represent in the final tables. It is explained within the paper, but looking at the tables without having the explanation in front of you may be confusing.