



SE2832 Lab 9: Mutation Testing

Due: May 14, 2014 23:59

1 Objectives

- Investigate the quality of existing test suites using Mutation testing
- Analyze existing JUnit test case effectiveness using the PIT Mutation testing tool

2 Introduction

Mutation testing is an effective method that can be used to evaluate the effectiveness of JUnit test cases. The general idea is to modify the source code in small ways. These, so-called mutations, are based on well-defined mutation operators that either mimic typical user mistakes or force the creation of valuable tests. Figure 1 provides a graphical picture of the process.

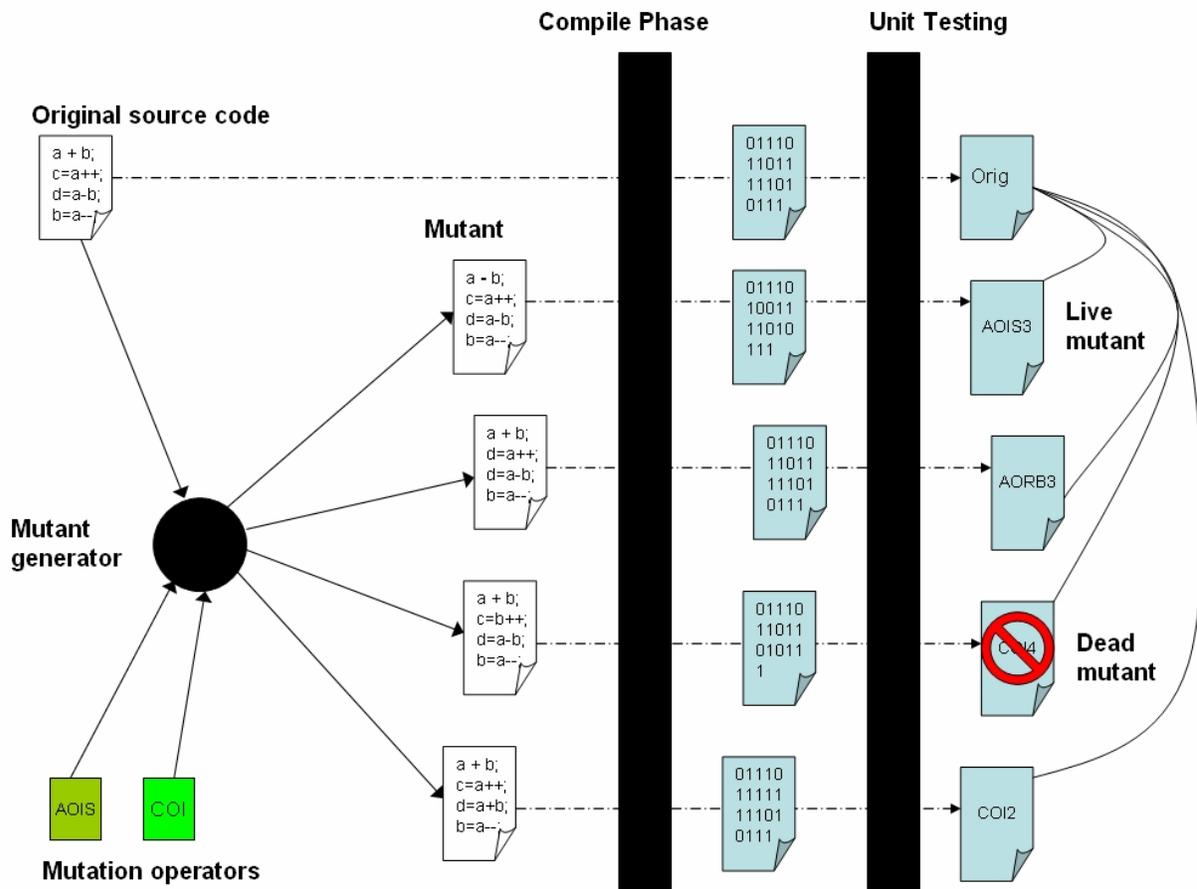


Figure 1 Mutation testing (From Sourceforge.net)



In this lab, you will use the PIT mutation tool to mutate the source code for several of your projects. In doing so, you will see how small changes to the source code can go unnoticed by test suites even if full coverage is obtained.

3 Prelab

Watch the video on how to install Pitclipse and how to use Pitclipse to do mutation testing on your software.

4 Installing Pitclipse

1. From an open Eclipse workbench, go to Help -> Install New Software
2. Add the PITclipse installation site. Click on the add button and enter the name "Pitclipse" and the url URL: <http://eclipse.pitest.org/release>
3. Install the plugin by following the dialog boxes.
4. After the installation process is complete, Eclipse will ask you if you would like to restart the workbench. Click Yes.

5 Lab Details

5.1 Step 1: Evaluating the Tax Code Tests

Working with your partner, start by running the PIT tool on the tests you created for the tax calculator code in week 3. While you used code coverage to check the tests, it is very possible that mutation testing will reveal some tests problems with your testing.

As you run the tool, take a look at each mutation which is left live by your test suite. Determine if it is an equivalent mutation or not. Calculate the initial mutation score using the formula

$$MutationScore = \frac{KilledMutants}{TotalMutations - EquivalentMutations}$$

Once you have removed all of the live mutations, calculate a mutation again. The only remaining mutations at this point should be stubborn mutants.

5.2 Step 2: Evaluating the Dental Calculator and the Stock Market Analyzer

Once you have calculated the mutation score for your tests on the tax code, run the PIT tool on your dental calculator and stock market analyzer code. (Hint: One partner can run one analysis while the other does the second program.) Determine the mutation score for each one of them.

Once you have determined the mutation score, add JUNit tests to your test suite until the mutation score is as close to 100% as you can make it. To do this, write new test methods and place them at the bottom of the file until you have exhausted the ability to create test cases. What is the mutation score now, and by how much has it improved?





6 Lab Deliverables

Each lab group will be responsible for submitting one report with the following contents:

1. Introduction
 - a. What are you trying to accomplish with this lab? This section shall be written IN YOUR OWN WORDS. DO NOT copy directly from the assignment.
2. Mutation Scores:
 - a. In a table, list each mutation from the tax collection code and whether it is killed or not. For those which are not killed, indicate if they are equivalent. From this data, calculate the mutation score including the removed equivalent mutations.
 - b. For the second portion of the lab, include the mutation scores for each of the files before test cases are added as well as after test cases are added. How much did they improve on a percent basis?
3. Equivalent and stubborn mutants
 - a. For each program, how many equivalent mutants did you find? What were they?
 - b. For each program, how many stubborn mutants did you find? What were they?
 - c. How easy was it to the analysis for a and b?
4. Added tests.
 - a. How many tests did you add to the second part to improve the mutation scores?
 - b. How many tests were added relative to the initial number of tests?
5. Things gone right / Things gone wrong
 - a. This section shall discuss the things which went correctly with this experiment as well as the things which posed problems during this lab.
6. Conclusions
 - a. What have you learned with this experience?
 - b. What improvements can be made in this experience in the future?

This material should be submitted as a single pdf file named <studentnames>_SE2832_Lab8Report.pdf.

Because you are working as a group, only one report and code submission is necessary.

If you have any questions, consult your instructor.