Lab 2: Black-Box Testing

Per Runeson and Elizabeth Bjarnason, with contributions from
Carina Andersson, Thomas Thelin, Yeni Li Helgesson and SoberIT.

Lund University, Dept. Computer Science

January 19, 2017

1 Introduction

In black-box testing, the purpose is to the test the output from the component under test. There are different strategies to use in order to test efficiently. In this exercise you will use equivalence partitioning (EP) and boundary-value analysis (BVA) on a small code example. You will also get insight into the Junit approach by using JUnit when applying the black-box techniques.

1.1 Learning Objectives

The exercise aims at giving an understanding of black-box testing. The specific learning goal is to gain a detailed insight into two common black-box techniques, namely equivalence partitioning and boundary-value analysis.

Assessment is done by the lab supervisor(s) who will review your work, defined in the output section of each assignment. This lab is also assessed through a written lab report, see instructions in Section 5.

2 Preparation (on paper)

Assignment 1: Read chapter 9.2-9.6 in [1] and the slides from the lecture 3, ***Black-box test techniques***.

Assignment 2: If you haven't used JUnit before, read the instructions available at http://junit.org/junit4/ (for example, start by looking under the tag 'Getting started". Ensure that you know how to use JUnit before you arrive to the exercise where you are supposed to implement test cases in the tool.

Assignment 3: Read the documentation for the program Triangle available at the course homepage. (NOTE: do **not** read the source code!) Specify test cases by using the techniques equivalence partitioning (EP) and boundary-value analysis (BVA) at a unit test level (specify test cases for each class method, but exclude the main method). First define *equivalence classes*, both *valid* and *invalid* ones. Then specify test cases for EP and for BVA based on these classes. Remember to specify *test inputs*, *execution conditions* and *expected output*. Make sure that the test cases cover both *valid* and *invalid equivalence classes*.

Output:

- One set of equivalence classes covering valid and invalid cases
- Test cases for EP
- Test cases for BVA

3 Exercise (on computer)

Assignment 4: Now we would like you to thoroughly test the Triangle program using equivalence partitioning (EP) and boundary-value analysis (BVA). You should implement the test cases you prepared on paper before the exercise. Add new test cases if you discover equivalence classes you missed during the preparation. Preferably, use Eclipse and Junit.

When you have implemented the test cases, execute them.

A test report including defects should be written. Record your test results carefully in your test report. Remember to specify test case ID, what is tested, description, input, expected output and other useful information. Also note pass/fail and your reflections on the found defects. An example defect report can be found in in Appendix 1.

Output:

- Full set of test cases for EP
- Full set of test cases for BVA
- Test & defect report for EP
- Test & defect report BVA

4 Analysis and Conclusions

Assignment 5: Reflect on the outcome of your tests for each test technique (EP and BVA). Consider the following questions:

- Equivalence classes: How where the equivalence classes designed? Could they be improved?
- Test case selection: How many and which kind of test cases did you select for each method?
- Detected defects: How many and which kind of defects were found for each technique?
- Compare the test techniques:
  - For this specific case, which technique worked best (found most bugs, was most cost efficient etc) and why?
  - When each method is most applicable? Consider both this case and in general.
  - What other black-box test techniques (at least 2) could be appropriate, when and why? Consider both this case and in general.

Output: Reflections and conclusions for each of the questions in assignment 5.

5 Report

The purpose of this lab report is to compare white-box (lab 1) vs black-box testing by analysing the outcome of lab 1 and lab 2 exercises. Use the provided lab report template (see course web page). The size of the report should be 2-4 A4 pages (excluding appendix) and focus on your reflections and conclusions from the lab sessions (lab 1 and lab 2).

Describe your tests and the outcome of them. Compare and discuss the black-box vs the white-box assignments. The report must include the following and the **bolded** items must appear as sub-sections.

1. For section *Introduction* provide
   a. a brief description of the report assignment including its aim and purpose.

- b. the **Procedure for White Box Testing**, which should include a summary of *what you did* for lab 1. It should consist of a description of what you (same as for Black Box, see c). Include the flow graph and McCabe's Cyclomatic Complexity measure (Lab 1, assignment #4).
- c. the **Procedure for Black Box Testing**, which should include a summary of *what you did* for lab 2. It should consist of a description of what you did, i.e. the steps taken so that someone else could repeat your approach. Include which test techniques were used, how the techniques were applied, how the test cases were selected for each technique. Include a description of the equivalence classes (Lab 2, assignment #3).

2. For section *Results*, describe the outcome for Black Box and for White Box testing, summarised in table format with info per test method & technique and refer to this from the text for the following sub-sections:
    - a. **Number of Test Cases.** Report how many test cases you used for each test technique: EP, BVA, each of the coverage measures (statement, decision/branch and condition). Also for lab 1, report the final coverage values as given by the coverage tool.
    - *b.* **Detected Defects**. Report on how many and which type of defects were found for each method/technique (EP, BVA, each coverage type).
3. In Section *Discussions & Conclusions* you should discuss and conclude based on the Results, i.e. the outcome of your testing. Provide the following sub-sections:
    - a. **White-box techniques.** For the lab 1 case and in general, discuss:
        - i. Coverage Criterion; when different coverage criterion work best and why. Clearly connect your discussion to the results presented in the previous section.
        - ii. Coverage Tool; your experience with using the coverage tool. How did it support the testing including strengths and weaknesses?
    - b. **Black-box techniques**. For the lab 2 case and in general, discuss:
        - i. EP vs BVA, which method works best and why. Clearly connect your discussion to the results presented in the previous section.
        - ii. Which other black-box test techniques (at least 2) could be appropriate (when and why?)
    - c. **White-Box Testing vs Black-Box** . Compare white-box techniques (used in lab session 1) with black-box test techniques. Discuss advantages and disadvantages with each approach and when (situations, cases etc.) each approach is appropriate to use.
4. In *Appendix* you may provide additional and more detailed material, e.g. defect reports.
5. Attach the test cases for each lab as separate zipped files of java source code files, remember to classify them according to applied test technique and coverage measure.

6 Examination

A typical exam question on this material is [1, chapter 9.2-9.6, p. 263, Exercise 2,4,5,6,7].

References

[1] Kshirasagar N. and Tripathy P., Software Testing and Quality Assurance, John Wiley & Sons, 2008.

Appendix 1:

Test ID	Method Under Test	Description of Defect	Expected Result	Actual Result	Remarks
1.	isInsurable()	When individual is female over 85 the output is True, where actual output should be False.	False	True	Sever Defect