

CSc-179 - Software Test+Quality Assurance

Assignment 2: Black-Box Testing

Fall 2017

(Due Sept 22, 2017)

1 Introduction

In black-box testing, the purpose is to test the output from the component under test. In this assignment, you will use equivalence partitioning (EP) and boundary-value analysis (BVA) on a small code example. The assignment 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.

2 Preparation:

Step 1: Read chapter 9.2-9.6 in [1] and the course slides from the lecture week 3 in SacCT, *Black-box testing*.

Step 2: Read the specification for the program Triangle available attached in the assignment 2. 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. Please exclude the main method). First define *equivalence classes*, both *valid* and *invalid* ones. Then design test cases for EP and for BVA based on these classes. Remember to specify test scenario, test case, *execution conditions (Pre-condition)*, *Test step (when applicable)*, *Test Data*, 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 Implementation (on computer)

Step 3: 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. Please use the provided attached jar file for your work.

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 Appendix 1.

Output:

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

4 Analysis and Conclusions

Step 4: Reflect on the outcome of your tests for each test technique (EP and BVA). Consider the following questions:

- ☐ Equivalence classes: How were 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 step 4.

References

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

Appendix 1: (Sample only – Not all fields are applied)

Test Case ID	Date Tested	Tester	Use Case Module	Use Case being tested	Pass/Fail	Precondition(s)	Input(s)	Observed Output(s)	Expected Output(s)	Post Condition(s)	Comments
TC_Session_01	02/22/2016 02/29/2016	Sushma Gajendra	Session	System reads a customer's ATM card	Pass	System is on and IDLE (not servicing a customer)	User presses on "click to insert card" button. User enters ATM card number : 1	System is requesting user to enter the pin. It displays the following message: "Please enter your PIN Then press ENTER"	Card is Accepted: System asks for entry of PIN	System changes state from IDLE and is now servicing a customer	Test case passes. No issues.

Credits: CSc-243 students (Spring 2016)