

SE463: Software Testing and Quality Assurance

ASSIGNMENT: Specification-based Testing Techniques

The purpose of this assignment is for you to practice with creating and executing test cases using various specification-based (aka functional or black box) testing techniques described in Chapter 5 of our textbook such as Boundary Value Testing (BVT) or (BVA for Boundary Value Analysis), Special Value Testing or Random Testing. You will also be able to compare the effort needed, efficiency and effectiveness of such test-case design techniques. To this end, we will learn and use appropriate effective tools such as Excel workbooks with spreadsheets and the embedded Visual Basic Editor to automate the execution of such test cases.

What to do: More specifically, you are required to do the following:

- 1) Find and download two Excel workbooks posted under this module on Moodle.
- 2) Review carefully the complete set of test cases included in these workbooks for the Adder program. These test cases have been created using the BVT and Random Testing approaches described in our textbook.
- 3) Test drive the Adder program by simply clicking on the button to execute the various test cases included in each spreadsheet. In addition, you will find a Clear button to erase the previously generated test cases.
- 4) Using the VB Editor, open and review the implemented code behind the buttons. You might need to activate the “Developer” tab in the workbook by following the instructions provided on the spreadsheets (next to the buttons).
- 5) Next, you will create your own separate Excel workbook and include a complete set of test cases for the Triangle program. You will need to create the necessary buttons using the VB editor and write the VB code needed for these buttons. You may add any additional buttons as you see necessary e.g. count/remove redundancies, automatically fill-in the expected results (aka “oracle”) etc.
- 6) While generating the necessary test cases, you will use all the testing techniques described in Chapter 5 as shown below:
 - Boundary Value Testing (BVT) or BVA-Boundary Value Analysis (4 flavors):
 - i. Normal (single fault, valid input data)
 - ii. Robust (single fault, valid/invalid input data)
 - iii. Worst-case (multi-fault, valid input data)
 - iv. Robust worst-case (multi-fault, valid/invalid input data)
 - Special Value (see an example in the PP slides)
 - Random Testing (you may use your own random-generation formula or the one mentioned in chapter 5 of your textbook)
- 7) For clarity, you should use separate spreadsheets (i.e. tabs) for each testing technique (same as the Adder program). Also, keep track of the time you needed to generate the test cases for each of the techniques (you will need it for item 8 below).
- 8) Execute all the test cases you have generated with a stable implementation of the Triangle program (you may use your own implementation or the one in your textbook). If you

discover any defects within your implementation here, please make a note of it and mention it in your report!

- 9) Next, you will inject five different faults in your code for the Triangle program. Make the faults to be very different in nature. Number each fault with a brief comment in the code explaining what the fault is. Run all the test cases again for each technique. Since you know the actual faults injected, you can check if each of the above methods revealed all (or some) of the faults that you injected.
- 10) Based on the above results you will compare and comment on the relative effectiveness (i.e. fault-finding power) of each of the above testing techniques used to generate your test cases. In other words, indicate the percentage of faults found (e.g. 3 out of 5 is 60%) by each technique. Include this information on a separate column in your spreadsheet.
- 11) In addition, for each technique used, you will assess its efficiency. This will include the amount of redundancy of test cases generated. You may choose to do it manually or automate the redundancy by creating/programming an extra button on the spreadsheet, which when clicked will count and identify all the redundant test cases. Include this information on a separate column in your spreadsheet (again as a percentage).
- 12) Finally, comment on the relative effort (i.e. time spent) required from your part, in order to create the test cases for each of the above techniques. More specifically, you will need to keep track of the amount of time spent to create test cases for each method and include this information in your spreadsheet.

What to Submit: On the due date, you will upload a single Excel workbook to include all of the following:

- 1) Six spread sheets with the complete set of test cases you created for each testing method mentioned above
- 2) The VB code for both the stable and faulty versions of your Triangle program
- 3) A separate spread sheet (tab) with a table, which will summarize the following metrics:
 - a. Total number of test cases generated by each technique
 - b. Percentage of defects discovered by each technique
 - c. Percentage of redundancy of test cases generated by each technique
 - d. Total time spent to generate test cases for each testing technique
- 4) A brief discussion of your comparison and evaluation of all testing methods used (you can include this in another separate spreadsheet)

Grading Rubric: This assignment will be graded out of 100% as follows:

- Correct and complete generation of test cases for all testing methods used: 50%
- Analysis of effectiveness (i.e. fault-finding power) of each method used: 15%
- Analysis of efficiency (i.e. redundancy) of each method used: 15%
- Your effort estimates for each testing method (i.e. time spent): 10%
- Overall quality of your assessment, conclusion and report: 10%