Swinburne University of Technology
Department of Computing Technologies

Software Testing and Reliability
SWE30009
Semester 2, 2023

Project Report

This is an individual assignment worth 60% of the total unit score.
Due date: 11:00pm, Monday, 16 Oct 2023

DESCRIPTION

In this assignment, you are required to do the followings:

A. TASKS

This assignment is on metamorphic testing, random testing and mutation testing. You are required to complete the following tasks.

Task 1: Random testing

- ***Subtask 1.1. Present your understanding of the random testing methodology.*** The description must cover at least: the intuition of random testing, distribution profiles for random testing, the process of random testing and some applications. You are also required to provide examples and/or illustrations whenever appropriate to improve the comprehensibility of your description.
- ***Subtask 1.2.*** You are required to ***apply random testing methodology to generate concreate test cases to test a program*** which is designed to sort a non-empty list of integer numbers which may contain duplicated numbers. You are required to provide at least 2 test cases being generated by a random generator to demonstrate how the methodology works. Note that you don't need to test any actual program in this task.

Task 2: Metamorphic testing

- ***Subtask 2.1. Present your understanding of the metamorphic testing methodology.*** The description must cover at least: test oracle, untestable systems, the motivation and intuition of metamorphic testing, metamorphic relations, the process of metamorphic testing and some applications. You are required to provide examples and/or illustrations whenever appropriate to improve the comprehensibility of your description.
- ***Subtask 2.2.*** You are required to apply metamorphic testing to ***propose at least two metamorphic relations (MRs) to test a program*** which is designed to sort a non-empty list of integer numbers which may contain duplicated numbers. You are required to provide *at least one concrete metamorphic group for each MR*. Note that you don't need to test any actual program in this task. You are required to provide description, explanation on the intuition of each MR and how it works.
- ***Subtask 2.3.*** You are also required to ***compare the advantage and disadvantage of random testing and metamorphic testing*** which should cover, at least, oracle and test case.

Task 3: Test a program of your choice.

In this task, you are required to test a real-world program of your choice using the metamorphic testing technique and evaluate it using the mutation analysis. You are required to propose and describe at least *two metamorphic relations* for testing in this task. You can freely choose any program, but make sure that (i) the original program under test is implemented correctly and must be obtained from GitHub; (ii) the program must be written in either Python, Java, JavaScript, Ruby, C/C++, C#, Swift, Visual Basic, Fortran, R, Go, Perl, PHP or MATLAB; and (iii) it is neither too large and complex nor too simple so that you can generate at least 20 non-equivalent mutants from the original program (either manually or using a mutant generation tool). You are required to have *screenshot(s) capturing your program being executed* and *discuss the finding* with supporting tables and/or figures showing the effectiveness of metamorphic relations in this task. You are also required to provide a self-contained and complete description of the testing process and discussion of results in the report.

B. SUBMISSION REQUIREMENTS

- You need to prepare a self-contained, complete, comprehensible and typed report and (ii) a complete source codes compressed in a ZIP file. Your report must use 12-point font size on A4 papers. Your report must contain no more than 10 pages and be smaller than 10 MB. Coverage page is not required, but your submission must contain your full name and student number on the first line of its first page.

- You must include the program under test (for Task 3) and other relevant data. The program is required to be written in one of the following languages: Python, Java, JavaScript, Ruby, C/C++, C#, Swift, Visual Basic, Fortran, R, Go, Perl, PHP or MATLAB. It must consist of mutants, test cases and, if applicable, test scripts. All of them must be compressed in a single ZIP file, which must have a size smaller than 10 MB. Choose your program and prepare the testing wisely.

- You must submit both files: (i) the Project Report as a single PDF file (which must have the filename specified in this format as "ProjectReport-YourStudentID-YourSurname.pdf", for example, ProjectReport-12345678-Smith.pdf); and (ii) the ZIP file containing source codes for the program under test, mutants, test cases and if applicable, test scripts (with the file name in the format "ProjectReport-YourStudentID-YourSurname.zip", for instance, ProjectReport-12345678-Smith.zip) before the due date using the Submission System in Canvas available to you.

MARKING CRITERIA

- o **A maximum of 15 marks** for the demonstration, technical correctness, comprehensibility, cohesion and quality of Task 1. They include 10 marks for Subtask 1.1, and 5 marks for the Subtask 1.2.
- o **A maximum of 25 marks** for the demonstration, technical correctness, comprehensibility, cohesion and quality of Task 2. They include 10 marks for Subtask 2.1, 10 marks for the Subtask 2.2; and 5 marks for Subtask 2.3.
- o **A maximum of 50 marks** for the demonstration, technical correctness, comprehensibility, cohesion and quality of Task 3. They include 20 marks for metamorphic relations, their test cases and outputs and the effectiveness of metamorphic relations; 10 marks for the selection of program under test, the diversity of its generated mutants and the source codes (containing the program, the mutants, the test cases, and if applicable, the test scripts); 10 marks for the results and discussion; and 10 marks for comprehensibility, cohesion of description, and technical correctness.
- o **A maximum of 10 marks** for the presentation, formatting, structure, comprehensibility and cohesion of your report, as well as the compliance with the submission requirements.
- o **Penalty** will be applied for late submission and plagiarism. Refer to the Unit Outline for the policy on late submission and plagiarism.