



Swinburne University of Technology
Department of Computing Technologies

Software Testing and Reliability
SWE30009
Semester 2, 2024

Project Report

This is an individual assignment worth 60% of the total unit score.

Due date: 23:00 Monday 14 October 2024

DESCRIPTION

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

A. TASKS

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

Task 1: Present your understanding of testing concepts

- **Subtask 1.1. Present your understanding of effectiveness metrics and how they can be applied to evaluate random testing and partition testing.** The description must cover P-measure, E-measure and F-measure. You are required to discuss the effectiveness of random testing and partition testing in testing a program with an input domain consisting of 3 partitions of sizes 100, 200, and 250. Assume there are 3, 5, and 2 failure-causing inputs, respectively in these partitions. **Demonstrate how the proportional sampling strategy works** on the abovementioned example and discuss its outcome in terms of effectiveness.
- **Subtask 1.2. Present your understanding of metamorphic testing.** Your description must cover **at least:** test oracle, untestable systems, the motivation and intuition, metamorphic relations (MRs), the process of metamorphic testing and some applications. You are required to provide your own examples with **at least three metamorphic relations** in your description.
- **Subtask 1.3. Present your understanding of the mutation testing.** Your description must cover **at least:** the mutants, mutation operators, and how mutants are killed. You are also required to provide examples for a set of at least 5 mutation operators of your choice which can be used to generate a set of 20 mutants in total.

Task 2: Test a program of your choice.

In this task, you are required to test a real-world program of your choice using **metamorphic testing** and evaluate your testing using **mutation testing**. You are required to propose and describe **at least two metamorphic relations (and at least five metamorphic test groups for each metamorphic relation)**. These metamorphic relations must be different to those given by you in Subtask 1.2. You can freely choose any program, but make sure that (i) the selected program is implemented by somebody else other than yourself, it satisfies all metamorphic relations and metamorphic test groups defined by you, and it

must be available from a public repository in GitHub; (ii) the selected program must be written in either Python, Rust, Java, JavaScript, Ruby, C/C++, C#, Bash, R, Go or MATLAB; and (iii) the selected program is neither too large and complex nor too simple so that you can generate **at least 30 non-equivalent mutants** from the original program (either manually or using a mutant generation tool) for testing in this task. You are required to **provide screenshot(s) capturing your program being executed and discuss the findings with supporting data** showing the effectiveness of MRs 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 must prepare **a self-contained, complete, comprehensible and typed Project Report on Tasks 1 and 2**. Your Project Report must use 12-point font size on A4 papers. It must contain **no more than 12 pages and be smaller than 10 MB**. Coverage page is not required but your report must contain your full name and student number on the first line of its first page.
- You must prepare **a single compressed ZIP file for Task 2** which consists of (i) the complete program selected for testing in Task 2, its data and a readme file indicating where you get the program, putting them all in a folder named SUT; (ii) the list of mutants constructed in Task 2 in a folder named MUTANTS; and (iii) the list of metamorphic test groups used in Task 2 and, if applicable, test scripts, in a folder named TEST. The single ZIP file must have a **size smaller than 10 MB**. Choose your program and prepare the testing wisely.
- **You must submit these two files:** (i) the Project Report as a single PDF file with the filename specified in this format as “ProjectReport-YourStudentID-YourFullName.pdf”, for example, ProjectReport-12345678-JohnNash.pdf; and (ii) the ZIP file with the file name in the format “ProjectReport-YourStudentID-YourFullName.zip”, for instance, ProjectReport-12345678-JohnNash.zip **before the due date** using the Submission System in Canvas available to you.

MARKING CRITERIA

- **A maximum of 30 marks** for the demonstration, technical correctness, comprehensibility, cohesion and quality of Task 1. They include 15 marks for Subtask 1.1, and 10 marks for the Subtask 1.2 and 5 marks for Subtask 1.3.
- **A maximum of 60 marks** for the demonstration, technical correctness, comprehensibility, cohesion and quality of Task 2. They include 20 marks for metamorphic relations (including novelty, technical description, correctness), their metamorphic test groups and 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 quality and efficiency of code and testing; 10 marks for the quality of results, discussion and insights; and 10 marks comprehensibility, technical correctness and cohesion of description.
- **A maximum of 10 marks** for the presentation, formatting, structure of your report, as well as the compliance with the submission requirements.
- **Penalty** will be applied for late submission and plagiarism. Refer to the Unit Outline for the policy on late submission and plagiarism.
- You are **NOT allowed to use of any artificial intelligence tool** such as ChatGPT, Bard to complete any part of your assignment submission (including report contents, code, mutants and test script). It will be treated as plagiarism and will be penalized.