Software Engineering Testing (CMPE3008)

Semester 1, 2022

Due: Tuesday 17 May, 23:59 GMT+8

Weight: 25% of the unit mark

1 Introduction

In this assignment, you will be investigating real-world code and its associated tests in a team of roughly 3 people. The goal is to apply concepts taught in Software Engineering Testing to Analyse the code, evaluate the testing suit in place, and propose improvements.

2 The Tasks

As part of this assignment, you are expected to complete a series of tasks. You are first tasks with finding source code according to specifications detailed below. Using that source code, you are to then investigate and apply coverage techniques.

2.1 Source Code.

Part of this assignment will be to find suitable source code for use with this assignment. There are many public repositories such as GitHub and Google Code that may assist you in finding examples. For the purposes of the assignment, you are able to choose code from any exception-capable language you wish (so long as it is not an esoteric language such as Whitespace.)

2.1.1 Requirements for Source Code

You are to find source code with at least **two (2) methods** and at least **two (2) related test methods** for a total of four methods. Additionally, your choices should adhere to the following requirements.

- *All* chosen methods:
  - Must not be empty
  - Must come from a publically available source
- *All* Non-Test methods:
  - Must have some form of control structure, such as if statements and/or loops
  - Must contain at least five lines, not to include method declarations or lines containing only brackets
  - Must have at least one variable
  - Must have at least one input (explicit or implicit)
  - Must have one of the following:
    - Multiple returns
    - An exception is thrown
    - A loop is present

2.1.2 Coverage & Testing

Once you have chosen the methods to investigate, you are to perform the following tasks using concepts taught in the lectures and tutorials:

- Convert your non-test methods into graphs
- Perform Prime Path Coverage on your graphs
- Design a set of tests using Base Choice analysis
    - Each method should have at least two characteristics
    - You may omit actual values for this analysis and focus only on blocks

Compare your results to the test methods you have chosen and make a short analysis of those test methods. Note that some projects may require some complex knowledge of their context. For this reason, you are **not** expected to determine how thorough or complete the coverage of the tests are. You should, however, make an estimated judgement regarding how many test paths are performed and provide an opinion on whether or not the testing is sufficient.

2.2 Testing Tool Investigation & Presentation

Your team will research the tools and frameworks that are available for testing software projects, write a summary, and give a presentation providing an evaluation on a particular tool. The tool or framework you choose and report on will be at your team's discretion, but it should be related to testing software.

You may not use JUnit or Python's `unittest` library for this assignment.

Your team should identify, at a minimum:

- The purpose and functionality of the given tool or framework
- How the tool or framework should be used
    - General descriptions and examples should both be provided
- Any other relevant considerations (such as standards, popularity, etc.)
- Whether or not your team would recommend it for actual use

Although you may choose any relevant tool, a useful place to start can be found here: http://softwareqatest.com/qatweb1.html

2.2.1 Presentation & Report

Your team will create a **5-minute presentation** on your chosen testing tool/framework. This presentation will be held during the normal lecture day and times on October 13th and 20th over Blackboard Collaborate.

A **short** written report detailing your findings should be included. The report should not exceed two pages except to include graphics. The report should cover the points outlined above.

3 Deliverables

The assignment should be submitted as a .pdf document or series of documents. They should:

(a) Include the details of your chosen code project and where it can be located

(b) Your chosen methods and their associated graphs

(c) The coverage for those graphs

(d) Your analysis for each test

(e) Your written summary for the research tool

4 Marking

The allocation of marks for this assessment are as follows:

- Source Code
  - Code is accessible via public repository **[3 marks]**
  - Code conforms to criteria specified
    - No specific marks, however, penalties apply to graph-related coverage for missing criteria
- Coverage & Testing
  - Prime Path Coverage **[10 marks]**
  - Base Choice Coverage **[10 marks]**
  - Test Analysis **[5 marks]**
- Test Tool/Framework
  - Written Report **[10 marks]**
  - Presentation **[7 marks]**
- Submission Quality
  - Submitted materials are professionally laid out, readable

    [5 marks]

Academic Integrity

Please see the *Coding and Academic Integrity Guidelines* on Blackboard.

In summary, this is an assessable task. If you use someone else's work or assistance to help complete part of the assignment, where it's intended that you complete it yourself, you will have compromised the assessment. You will not receive marks for any parts of your submission that are not your own original work. Further, if you do not *reference* any external sources that you use, you are committing plagiarism and/or collusion, and penalties for academic misconduct may apply.

Curtin also provides general advice on academic integrity at academicintegrity.curtin.edu.au.

The unit coordinator may require you to provide an oral justification of, or to answer questions about, any piece of written work submitted in this unit. Your response(s) may be referred to as evidence in an academic misconduct inquiry.