PES UNIVERSITY

AUGUST – DECEMBER 2022 SEMESTER 5

SOFTWARE ENGINEERING LAB TASKS

Professor-in-charge: Dr. Jayashree R

Teaching Assistant: Aanchal Narendran (Sem VII)

We hope at this point in your semester you have achieved considerable progress in your Software Engineering Project. All of the tasks in this lab manual are geared towards improving and testing your project prior to your submissions.

Problem Statement – 1: Unit Testing

A unit is the smallest block of code that functions individually. The first level of testing is Unit testing and this problem statement is geared towards the same.

- Discuss with your teammates and demarcate units in your code base
    - Note: discuss why the code snippet you have chosen can be classified as a unit
- Develop test cases for both valid and invalid data
- Ideate how you could further modularize larger blocks of code into compact units with your teammates

Problem Statement – 2: Dynamic Testing

Dynamic testing involves execution of your code to analyse errors found during execution. Some common techniques are Boundary Value Analysis and Mutation Testing.

Problem Statement – 2.a: Boundary Value Analysis

When it comes to finding errors in your code base, they are often found at locations where a condition is being tested. Due to this, developers often use Boundary Value tests to reduce defect density.

- How would you define a boundary test?
    - Note: Simple relational conditions are a basic example
- Build your boundary test cases and execute them

Problem Statement – 2.b: Mutation Testing

- Using your isolated units from the first problem statement, ideate with your team mates on how to mutate the code
- Develop at least 3 mutants of the functioning code and test all 4 code bases using the test case from the first problem statement

Problem Statement – 3: Static Testing

Static testing involves validating your code without any execution. Under this problem statement, you will be expected to analyse and calculate the cyclomatic complexity of your code.

- Using the unit you selected in the first problem statement as an example, develop the control flow graph of your problem statement.
- Using the Control flow graph, calculate the cyclomatic complexity of your code.
- Using the cyclomatic complexity as an indicator, Ideate and code your unit again to reduce complexity

Problem Statement – 4: Acceptance Testing

Assume your neighbouring team is the client for your code. Give them an idea of what your product is and the software requirements for the product.

- Exchange your code base and test each others projects to see if it meets user requirements
- If you identify a bug in the project you are testing, inform the opposing team of the bug
- As a team, based in clients experience, ideate modifications to the existing project that could improve client experience

Problem Statement – 5: Maintenance Activities

Once a product is completed, it is handed off to a service based company to ensure all maintenance activities are performed without the added expenditure of skilled developers. However, a few tasks are performed by the maintenance team to gauge the product better. In this problem statement, you will be asked to experiment with your code.

- Exchange code bases with your neighboring teams and reverse engineer a block of code in order to understand it's functionality
- After understanding the code block, Re-Engineer the code
    - Ideate how to refactor the code and the portion of the code base you would have to change
    - Discuss how the new changes would impact the time and space complexity of the project during execution
- After Reverse Engineering and Re-Engineering the code, perform acceptance testing between the teams