Manual steps for Mutation Testing

Team Info

Name: Sim_Hackers

Members: Ramya Panambayil Rajan
Sarath Sasidharan Nair
Pai Radhesh Udyavara
Vignesh Sekar

Purpose

This document outlines the manual steps for Mutation testing, which is used to evaluate the test quality of unit test by introducing small code changes(mutations) to the original source code

Pre-requisites

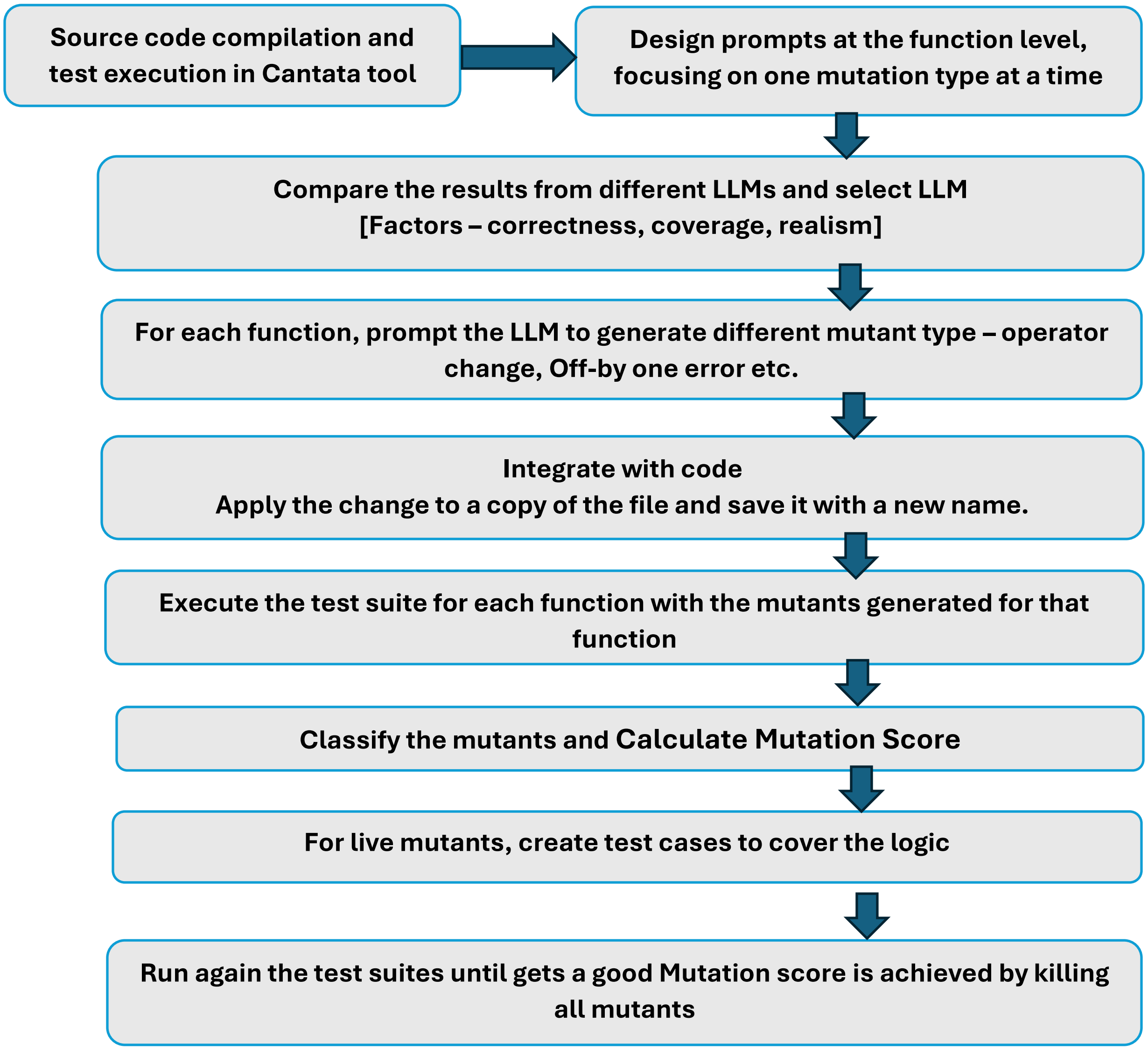
- VS code
- LLM Farm
- Cantata

Manual Steps:

1. Basic understanding of the source code, test suites and other env files shared by the hackathon team
   - In our case, its Net_MonitoringClasses.c file and test suite in Cantata
2. Make sure code can be compiled and test suites can be executed in Cantata
3. In LLM Farm, compare different LLM (GPT-4o mini and Gemini 1.5 Flash) by providing same prompts and code to different LLMs
4. Select the right prompting to get the more realistic mutants
   ****The prompt Selection of LLMs by comparing are added in design document**
5. Generate mutants for each function present in code considering different mutant type. Each mutant contains only one change
6. Create mutant code by duplicating the existing code and replace the line changed
7. Create new workspace with this mutant and import into Cantata and execute the existing test suite
8. Check the report summary
9. Identify the mutants – Live mutants, killed mutants, Equivalent mutants and Error mutants
   a. Live Mutants are the one in which test case is passed
   b. Killed Mutants are mutants in which test case is failed
   c. Equivalent mutants -- mutants that behave identically to the original program for all possible inputs
   d. Error Mutants in which compilation fails

We should focus only on first 2 mutants

10. Calculate Mutation Score= (Number of Killed Mutants/ Total Number of Mutants)×100
    A higher score indicates better test coverage and fault detection capability
11. For Live mutants, identify what is the difference between original code and mutant
12. Identify the test case for the function for which mutant is survived
13. Then add test case to cover that logic
14. Re-run testing until all mutants are killed and calculate the Mutation score



Source code compilation and test execution in Cantata tool

Design prompts at the function level, focusing on one mutation type at a time

Compare the results from different LLMs and select LLM [Factors – correctness, coverage, realism]

For each function, prompt the LLM to generate different mutant type – operator change, Off-by one error etc.

Integrate with code
Apply the change to a copy of the file and save it with a new name.

Execute the test suite for each function with the mutants generated for that function

Classify the mutants and Calculate Mutation Score

For live mutants, create test cases to cover the logic

Run again the test suites until gets a good Mutation score is achieved by killing all mutants