Experiment 03 - Version Control and Testing Lab

December 15, 2022

Aim

Prepare software test plan for the software (Unit Testing & Functional Testing)

Theory

- **Unit Testing** : It is a way of testing a unit - the smallest piece of code that can be logically isolated in a system.

- **Functional Testing** : Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered. The unit testing is a *subset* of the functional testing and performing multiple such test simulations the basis of this practical.

- **Software Plan** : The software plan is a detailed document which describes software testing areas and activities. It outlines the test strategy, objectives, test schedule, required resources (human resources, software, and hardware), test estimation and test deliverable.

- **Comparison** : Unit testing helps isolate specific code and determine if the unit works as intended. The goal is to facilitate early detection of code flaws that may be difficult to find and fix in later testing stages. Functional testing helps verify that each application feature works as intended.

- **Corner Cases** : A corner case testing is testing multiple parameters for extreme levels, as the user sticks to the corner of the configuration space.

- **Edge Cases** : An edge case is a problem or situation that occurs only at an extreme operating parameter.
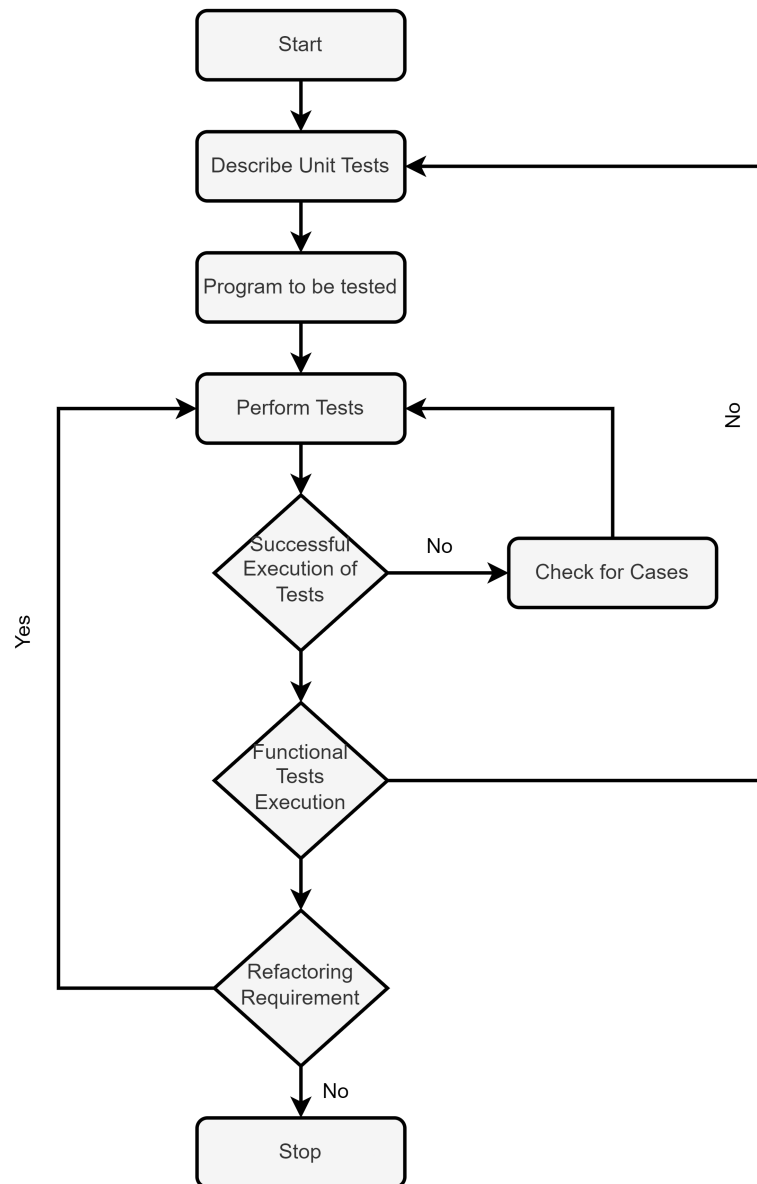
Results

Figure 1: Flowchart for Software Plan inclined towards Unit & Functional Testing

```
                    ┌─────────────┐
                    │    Start    │
                    └─────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │ Unit Tests for   │
                  │ Prime Number     │
                  │ Program          │
                  └──────────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │ Prime Number     │
                  │ Implmentation    │
                  └──────────────────┘
                           │
                           ▼
                  ┌──────────────────┐
                  │  Execute Test    │
                  └──────────────────┘
```

Start

Unit Tests for Prime Number Program

Prime Number Implmentation

Execute Test

Check if number is less than or equal to 2

No

Check for Input

No

Return type is Boolean addressable

No

Passed all test cases as a functional test
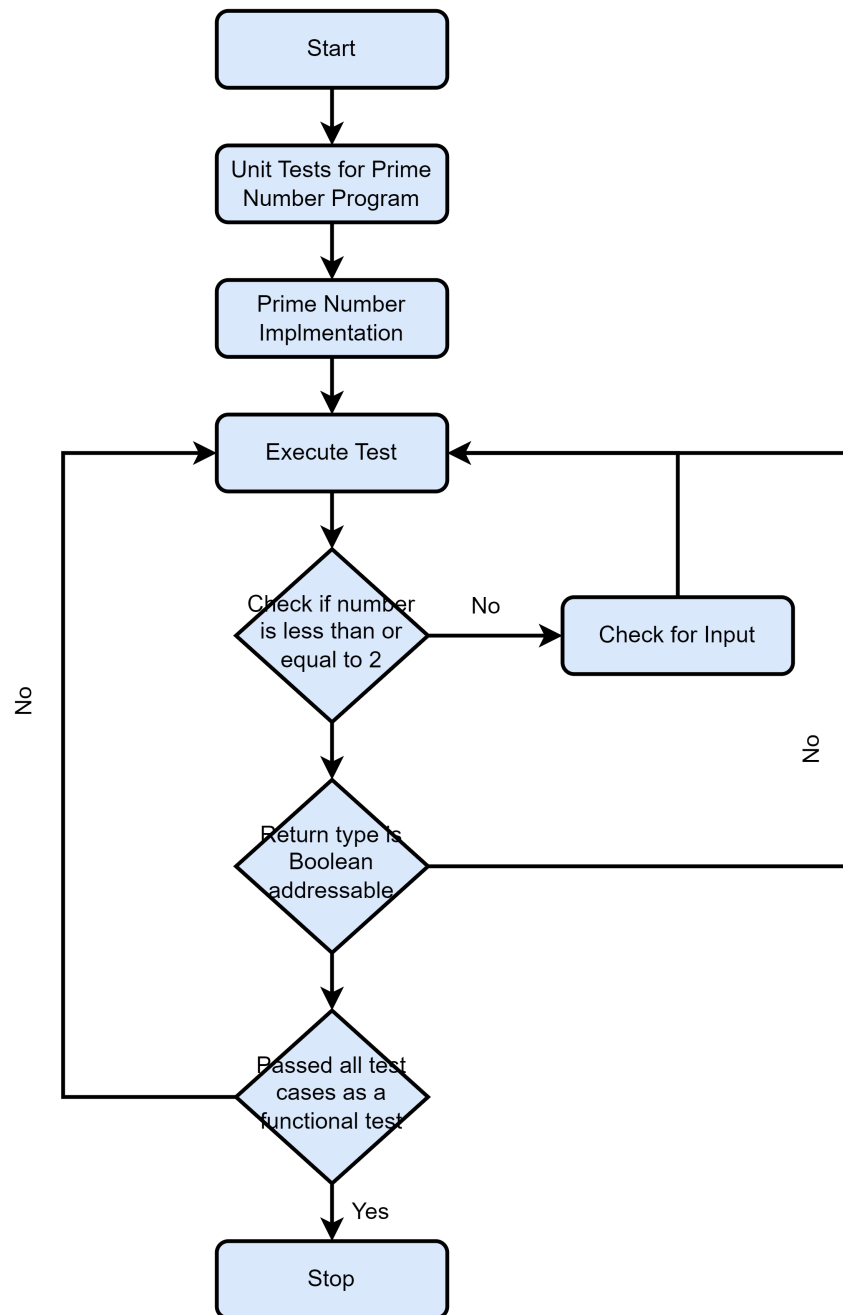
Yes

Stop

Figure 2: Software Plan for a prime number program that covers Unit as well as Functional Tests

Conclusion

In this experiment, the structured layout of advisable software plan for Unit and Functional testing is done effectively. This experiment is the premise of the next experiment which will be highly implementation oriented. The figure where prime number test example is taken, will be implemented in the latter experiment keeping this experiment pretty much as a blueprint of the deliverables to be executed.