Assignment # 2

Name: Muhammad Fahad

ID: FA19-BSSE-0014

Course: Test Driven Development

Section: AM

Teacher: KAMRAN CS

Difference between Code Coverage and Test Coverage:

Test coverage: Includes testing the features implemented as a part of the Functional requirements specification, software requirements specification, and other required documents.	**Code coverage:** Indicates the percentage of code that is covered by the test cases through both manual testing and Selenium or any other test automation framework.
For example: if your source code has a simple if…else loop, the code coverage would be 100% if your test code would cover both the scenarios i.e. if & else.	**For example:** if you are to perform cross browser testing of your web application to ensure whether your application is rendering well from different browsers or not?
Understanding Code Coverage: Code coverage is performed by developers during unit testing to verify the code implementation in such a manner that almost all the statements of code are executed.	**Understanding Test Coverage:** Unlike code coverage which is a white-box testing methodology, test coverage is a black-box testing methodology. Since the tests are derived from these documents, there are minimal/no chances of automation.
How To Perform Code Coverage? 1. **Branch coverage** – is used to ensure that every possible branch used in a decision-making process is executed. 2. **Function coverage** – Function coverage ensures that necessary functions (especially exported functions/APIs) are tested. 3. **Statement Coverage** – in which test code has to be written in a manner that every executable statement in the source code is executed at least once. 4. **Loop Coverage** – This approach is to ensure that every loop in the source is executed at least once. There might be some loops that may execute based on results that you achieved at runtime,	**How To Perform Test Coverage?** 1. **Unit Testing –** This type of testing is performed at a unit level/module level. 2. **Functional Testing –** In functional testing, the functions/features are tested against the requirements mentioned in the (FRS). 3. **Integration Testing –** It is also called as system testing since the software is tested on a system level. 4. **Acceptance Testing –** It all depends on the result of acceptance testing whether the product would be released to the end consumer/customer.

Tools for Code Coverage:

1. **Coverage.py –** It is a code coverage tool for Python.
2. **Serenity BDD –** Supporting Java & Groovy programming languages,
3. **JaCoCo –** JaCoco is a code coverage tool for Java.
4. **JCov –** JCov is a test framework agnostic code coverage tool.
5. **PITest –** Most of the code coverage tools check the code for branch coverage, statement coverage, loop coverage, etc.

Tools for Test Coverage:

1. **JUnit –** JUnit is the unit testing framework for Java. It can also be used for UI testing. It is open-source and considered important in the development of TDD (Test Driven Development).
2. **PyUnit –** PyUnit (also called as Python Unit Testing Framework) is a widely used testing framework primarily used for unit testing.

Advantages of Code Coverage:

- Provides the effectiveness of your test code and how you can improve the coverage
- Irrespective of the type of tool being used (open-source, premium), setting up a code coverage tool should not take much time.
- Helps in improving the code quality by capturing bugs in the code.

Advantages of Test Coverage:

- A good approach to test the software features and compare the results across different specification documents (requirements, feature, product, UI/UX, etc.)
- As the tests performed as a part of the coverage are black-box in nature, executing those tests might not require much expertise.

Shortcomings of Code Coverage:

- Majority of code coverage tools are limited to unit testing.
- The methodology used by tools could be different hence; you may not be able to compare code coverage results of one tool to another.
- Searching for the best-suited tool could be a big task as you need to compare & try features from those tools before selecting the best one that suits your project requirements.

Shortcomings Of Test Coverage:

- There is no scope of automation as the tests are predominantly black-box tests. Manual comparison of the test results has to be done with the expected output since these tests are performed at a 'feature level' and not 'code level'.
- No concrete way of measuring test coverage. Hence, the coverage results largely depend on the domain competence of the tester who is performing the tests and may vary from one tester to another.

• There are very few tools that provide support for different programming languages e.g. Java, Python, C, etc. Hence, you may need to have more than one tool in case your team is using multiple programming languages (for test code development).	