Sanket Mehta

TechDoc#7: Software Testing and JUnit

What is Software Testing?
Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.
Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing.
In simple terms, Software Testing means Verification of Application Under Test (AUT).

Types of Software Testings

1. Functional Testing
    a. Unit Testing
    b. Integration Testing
    c. User Acceptance Testing (UAT)
2. Non-Functional Testing or Performance Testing
    a. Performance Testing
    b. Scalability Testing
    c. Load Testing
3. Maintenance (Regression and Maintenance)
    a. Regression Testing

What is Unit Testing?
Unit testing is used to verify a small chunk of code by creating a path, function or a method. The term "unit" exists earlier than the object-oriented era. It is basically a natural abstraction of an object oriented system i.e. a Java class or object (its instantiated form).
Unit testing and its importance can be understood by below-mentioned points:

1. Unit testing is used to identify defects early in software development cycle.
2. Unit testing will compel to read our own code. i.e. a developer starts spending more time in reading than writing.
3. Defects in the design of code affect the development system. A successful code breeds the confidence of developer

What is Integration Testing?
In a software product there are multiple modules that converse with each other through an interface. Integrating these individual software modules and testing them as one is known as **Integration**

Testing. The objective of integration testing, which is an extension of unit testing, is to take smaller units of unit testing and test their behavior as a whole.
Although each software module is unit tested, defects still exist for various reasons like

1. A Module, in general, is designed by an individual software developer whose understanding and programming logic may differ from other programmers. Integration Testing becomes necessary to verify the software modules work in unity.
2. At the time of module development, there are wide chances of change in requirements by the clients. These new requirements may not be unit tested and hence system integration Testing becomes necessary.
3. Interfaces of the software modules with the database could be erroneous.
4. External Hardware interfaces, if any, could be erroneous.

Advantages of Integration Testing

1. Integration testing validates if all the unit tested modules are compatible with each other.
2. Integration testing or incremental testing is necessary to verify whether the software modules work in unity.
3. Integration testing can catch bugs across the modules which unit testing doesn't have scope.

What is JUnit ?

JUnit is an open source framework designed by Kent Beck, Erich Gamma for the purpose of writing and running test cases for java programs. In the case of web applications JUnit is used to test the application without server. This framework builds a relationship between development and testing process. Developers who follow test-driven methodology must write and execute unit tests first before writing any code. Once you are done with code, you should execute all tests, and it should pass. Every time any code is added, you need to re-execute all test cases and makes sure that nothing is broken. Junit is used to write unit as well as integration tests.

Resources:

1. JUnit Testing Basics
2. Junit5
3. Mockito Tutorial
4. EasyMock Tutorial
5. Junit5 + Mockito + Spring Example