Types of Testing

Unit Testing

UNIT TESTING is a level of software testing where individual modules/units/components of a software are tested. It requires detailed knowledge of the internal program design and code.

This type of testing should cover at least 80% of the code as validated by a code coverage tool such as SonarQube or Codacy.

Who is going to do Unit Testing?

It is normally done by developers themselves or their peers.

When is it performed?

Unit Testing is the first level of software testing and is performed prior to Integration Testing.

Unit Testing Tools

- JUnit
- NUnit
- PHPUnit
- CUnit
- CPPUnit

Unit Testing Best Practices

a. Unit Test cases should be independent. In case of any enhancements or change in requirements, unit test cases should not be affected.
b. Test only one code at a time.
c. Follow clear and consistent naming conventions for your unit tests
d. In case of a change in code in any module, ensure there is a corresponding unit Test Case for the module, and the module passes the tests before changing the implementation
e. Bugs identified during unit testing must be fixed before proceeding to the next phase in SDLC
f. Adopt a "test as your code" approach. The more code you write without testing, the more paths you have to check for errors.

Integration Testing

INTEGRATION TESTING is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

Who is going to do Integration Testing? Testers perform Integration Testing.

When is Integration Testing performed?

Integration Testing is the second level of testing performed after Unit Testing and before System Testing.

System testing

The entire system is tested as per the requirements. It is based on overall requirements specifications and covers all combined parts of a system.

Regression Testing

Regression testing is a type of software testing that verifies software previously developed and tested still functions and performs correctly after it was modified. Such modifications may include software enhancements, bug fixes, patches and configuration changes.

Functional Test Automation

Test automation is the use of special software to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place or add additional testing that would be difficult to perform manually.

Test automation is critical for continuous delivery and continuous testing.

There are various methods of functional test automation like using Selenium framework.