Software Testing Levels

- Basically to identify missing areas and prevent overlap and repetition between the development life cycle phases
- Requirement gathering and analysis, design, coding or implementation, testing and deployment; each phase goes through the testing. Hence there are various levels of testing.

1. Unit testing: It is basically done by the developers to make sure that their code is working fine and meet the user specifications. They test their piece of code which they have written like classes, functions, interfaces and procedures.
2. Component testing: It is also called as module testing. The basic difference between the unit testing and component testing is in unit testing the developers test their piece of code but in component testing the whole component is tested.
3. Integration testing: Integration testing is done when two modules are integrated, in order to test the behavior and functionality of both the modules after integration. Below are few types of integration testing:
   - Big bang integration testing
   - Top down
   - Bottom up
   - Functional incremental
4. System testing: In system testing the testers basically test the compatibility of the application with the system.
5. Acceptance testing: Acceptance testing are basically done to ensure that the requirements of the specification are met.
6. Alpha testing: Alpha testing is done at the developers site. It is done at the end of the development process
7. Beta testing: Beta testing is done at the customers site. It is done just before the launch of the product.

Unit testing

- A unit is the smallest testable part of an application like functions, classes, procedures, interfaces.
- Unit testing is a method by which individual units of source code are tested to determine if they are fit for use.
- Unit tests are basically written and executed by software developers to make sure that code meets its design and requirements and behaves as expected.
- The goal of unit testing is to segregate each part of the program and test that the individual parts are working correctly.
- White Box Testing method is used
- should be done before Integration testing.
- should be done by the developers.

Pros:

- Issues are found at early stage
- helps in reducing the cost of bug fixes

- Unit testing helps in maintaining and changing the code

Component testing ~ **module testing**

- After unit testing is executed, component testing comes into the picture.
- method where testing of each component in an application is done separately. Suppose, in an application there are 5 components. Testing of each 5 components separately and efficiently is called as component testing.
- For example, in a student record application there are two modules one which will save the records of the students and other module is to upload the results of the students. Both the modules are developed separately and when they are tested one by one then we call this as a component or module testing.
- finds the defects in the module and verifies the functioning of software.
- done by the tester.
- may be done in isolation from rest of the system depending on the development life cycle model chosen for that particular application.
- In such case the missing software is replaced by **Stubs** and **Drivers (dummy data)** and simulate the interface between the software components in a simple manner.
- Component testing plays a very important role in finding the bugs. Before we start with the integration testing it's always preferable to do the component testing in order to ensure that each component of an application is working effectively.

Integration testing

- Integration testing tests integration or interfaces between components, interactions to different parts of the system such as an operating system, file system and hardware or interfaces between systems.
- after integrating two different components together we do the integration testing.
- done by a specific integration tester or test team.
- Systematic integration strategies may be based on Incremental approach, top down approach, bottom up approach, big bang approach
  - Component integration testing: when both the modules or components are integrated then the testing done is called as Component integration testing. This testing is basically done to ensure that the code should not break after integrating the two modules.
    - Done after component testing
    - Tests the interactions between software components
  - System integration testing: System integration testing (SIT) is a testing where testers basically test that in the same environment and all the related systems should maintain data integrity and can operate in coordination with other systems.

        - May be done after system testing
        - Interactions between h/w and s/w
- The greater the scope of integration, the more difficult it becomes to isolate failures to a specific component, which may lead to increased risk and additional time for troubleshooting.
- While integrating 2 modules, testers are interested in testing the communication between the modules, not the functionality of the individual module (which was done in component testing)

System testing

- concerned with the behavior of a whole system/product
- the behavior of whole system/product is tested as defined by the scope of the development project or product
- may include tests based on risks and/or requirement specifications, business process, use cases, or other high level descriptions of system behavior, interactions with the operating systems, and system resources.
- most often the final test to verify that the system to be delivered meets the specification and its purpose.
- the test environment should correspond to the final target or production environment as much as possible in order to minimize the risk of environment-specific failures not being found in testing.
- carried out by specialists testers or independent testers.
- should investigate both functional and non-functional requirements of the testing.
- Testers also need to deal with incomplete or undocumented requirements.

Acceptance testing

- After the system test has corrected all or most defects, the system will be delivered to the user or customer for acceptance testing.
- Acceptance testing is basically done by the user or customer although other stakeholders may be involved as well. ~ responsibility of the customers
- The goal of acceptance testing is to establish confidence in the system.
- Acceptance testing may assess the system's readiness for deployment and use
- Acceptance testing is most often focused on a validation type testing.
- Acceptance testing may occur at various times in the life cycle, for example:
  o A (Commercial Off The Shelf) COTS software product may be acceptance tested when it is installed or integrated.
  o Acceptance testing of the usability of a component may be done during component testing
  o Acceptance testing of a new functional enhancement may come before system testing
- The **types of acceptance testing** are:
    - The **User Acceptance test:** focuses mainly on the functionality thereby validating the fitness-for-use of the system by the business user. The user acceptance test is performed by the users and application managers.
    - The **Operational Acceptance test:** ~ Production acceptance test
        - validates whether the system meets the requirements for operation.

    - In most of the organization the operational acceptance test is performed by the system administration before the system is released.
    - The operational acceptance test may include testing of backup/restore, disaster recovery, maintenance tasks and periodic check of security vulnerabilities.
- **Contract Acceptance testing**: It is performed against the contract's acceptance criteria for producing custom developed software. Acceptance should be formally defined when the contract is agreed. Acceptance criteria should be defined when the parties agree to the contract.
- **Compliance acceptance testing:** It is also known as regulation acceptance testing is performed against the regulations which must be adhered to, such as governmental, legal or safety regulations.