




Levels Of Testing

- 
- Unit Testing
 - Integration Testing
 - System Testing
 - User Acceptance Testing

Levels Of Testing

- Software Testing is done at various stages of SDLC (comprising varied scope) to ensure that the software/product meets the requirement specifications at each stage.
- Scope of testing at each level is different.
- Various Levels of Testing are:
 - Unit Testing (Component Testing)
 - Integration Testing
 - System Testing
 - User Acceptance Testing

Unit Testing

- Individual units of source code are tested to determine if it conforms to the technical specification document.
- The smallest unit of code will be tested. For instance, individual program, function stored procedure.
- In object-oriented programming, the smallest unit is a method, which belongs to base/super class/abstract class/derived child class.
- The field validations, presence of menu controls & layout of the page are tested with different inputs after coding /UI Design.
- Code is executed & tested to ensure that each line of code is run for the required unit test cases.

Unit Testing

- Lowest level of testing
- Part of the construction phase of the SDLC
- Done once coding for the unit or program is completed
- Generally performed using White Box Testing

Unit Testing

- Who performs it?
 - Unit Testing is generally performed by software developers themselves or their peers
- Test Artefacts created are:
 - Unit Test case documents
 - Unit Test Logs
 - Unit Testing Defect report

Unit Testing

- Entry Criteria :
 - Unit Test Plan is baseline
 - Coding is completed
 - Code is reviewed and baseline (optional)

- Exit Criteria :
 - 100% statement coverage
 - Acceptable (predetermined) levels of code coverage obtained

Integration Testing

- Integration testing is done to check the dataflow between the modules(components or units) or between the systems.

- Integration testing is categorized into two:
 - Unit Integration Testing
 - System Integration Testing



Unit Integration Testing

- The individual software modules/pages are integrated and tested for their functionalities as one single unit.
- Ensures that the individual units work together as a whole and the data flows (back and forth) across the units as per the requirement specification.
- Done after two or more programs or application components have been successfully unit tested.
- Any black box, white box testing methodology can be used to test.
- Also referred to as Assembly Testing and API Testing.

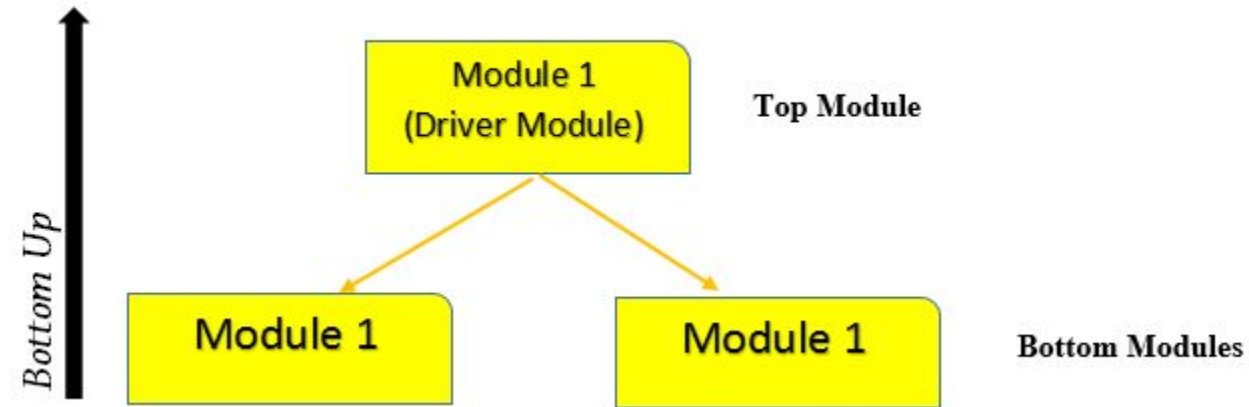
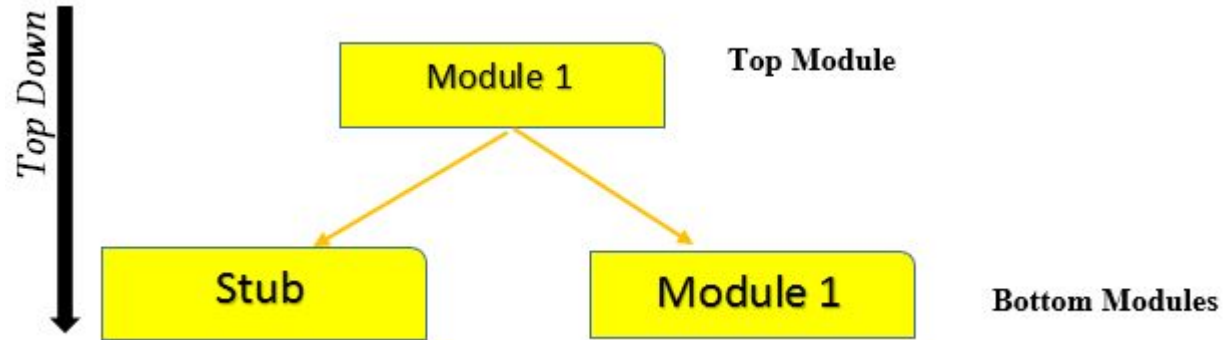
Unit Integration Testing

- Entry Criteria
 - Integration Test Plan is base-lined
 - Code and unit testing is completed
 - Unit test errors are fixed and verified
 - Code is reviewed and base-lined
- Exit Criteria :
 - 100% thread coverage achieved
- There are two types of integration testing
 - Incremental testing
 - Continuous testing of an application as new functionality is added
 - Non-incremental testing
 - Testing of combined parts of an application to determine if they function together correctly

Unit Integration Testing - Approaches

- Depends upon organization we have different approaches which are
 - Top-Down Approach
 - Bottom-Up Approach
 - Sandwich Approach
 - Big-Bang Approach

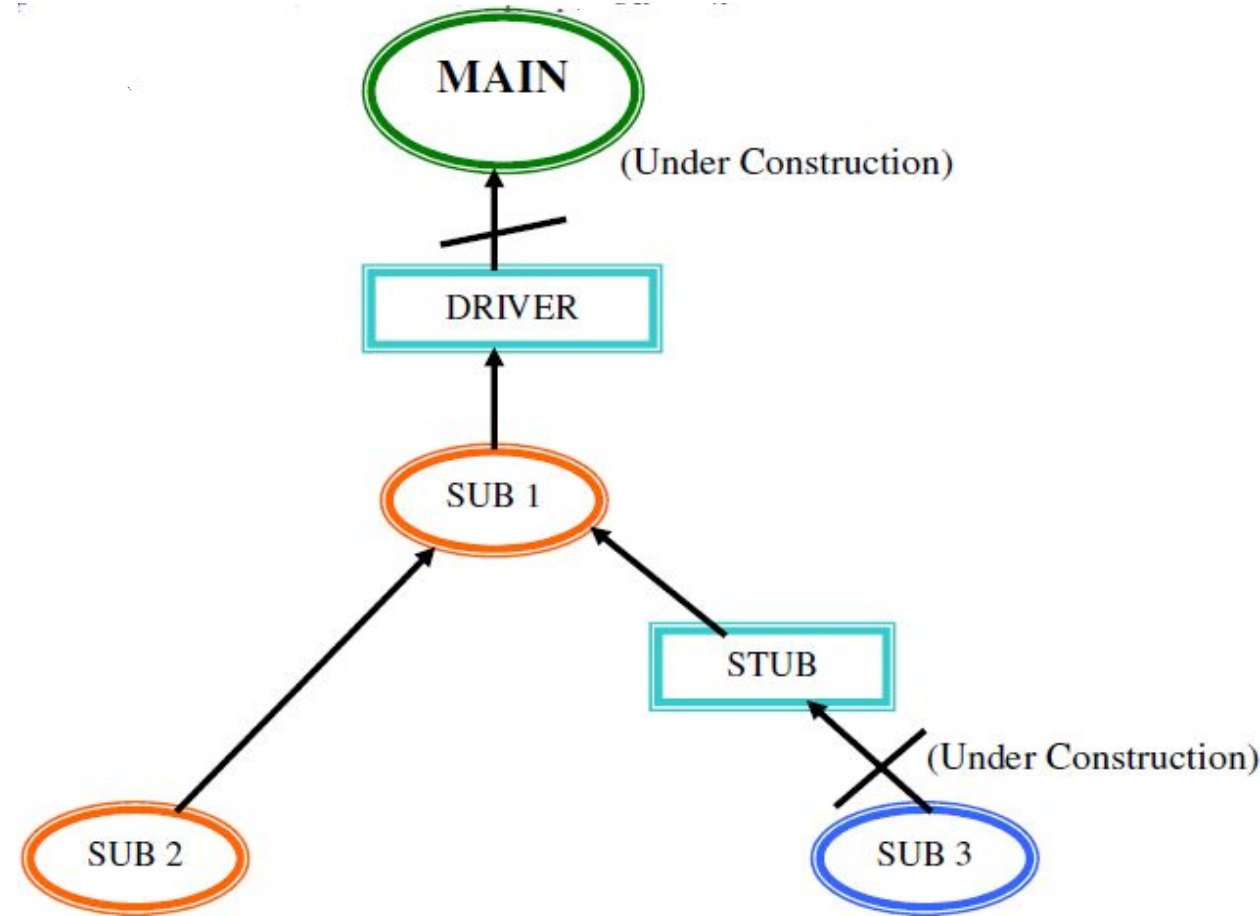
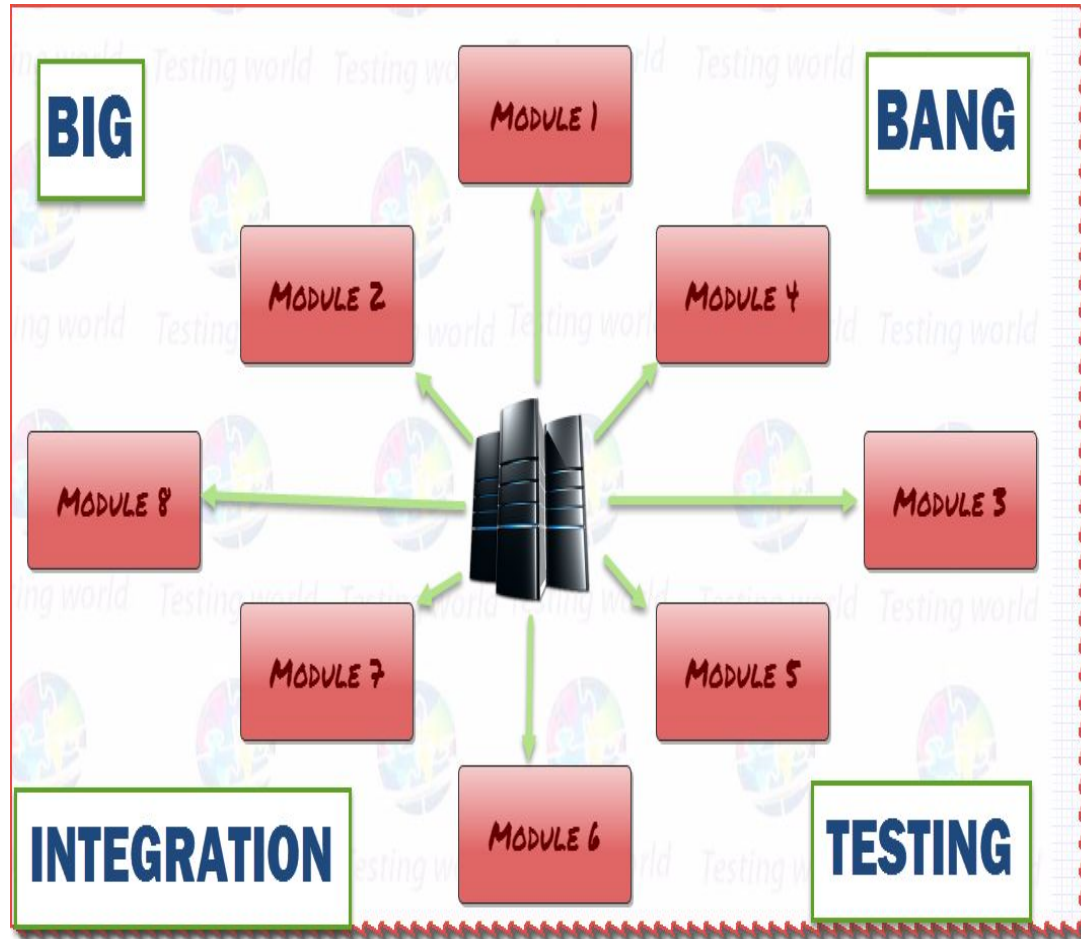
Top-Down & Bottom-Up Approach



Stubs And Drivers

- ❑ Stubs are dummy pieces of code.
- ❑ On the other hand, drivers are the ones, which are the "calling" programs.
- ❑ Drivers are dummy code used in bottom up testing approach.
- ❑ Drivers are used when the sub modules are ready but the main module is still not ready.

Big-Bang & Sandwich Approach



Sandwich
Approach

Unit Integration Testing

- Who performs it?
 - Either Developers themselves or independent Testers
- Test Artefacts created are:
 - Component Integration Test case documents
 - Component Integration Test Logs
 - Component Integration Testing Defect report

System Testing

- Testing of Complete, integrated system/software.
- Performed on the entire system where the system is validated against the Functional Requirement Specification(s) (FRS) and/or a System Requirement Specification (SRS).
- Black box Testing is preferred to carry out the system testing.
- Defects that are missed to be caught in the unit, Assembly testing are uncovered.
- Some of the scenarios will be covered for test only during system testing.
- Individual modules may work as desired but may fail as a system – this can be uncovered during system testing.

System Testing

- Who performs it?
 - Performed by an independent Testing team
- Test artefacts created are :
 - System Test case documents
 - System Test Logs
 - System Testing Defect report

System Integration Testing

- ❑ Carried out to check if the system works in conjunction with the other systems/interfaces(as below) and the data flows across them as desired:
- ❑ LAN/WAN, communications middleware
- ❑ Other internal systems (billing, stock, personnel, overnight batch, branch offices, other countries)
- ❑ External systems (stock exchange, news, suppliers)
- ❑ Intranet, internet / www
- ❑ Third party packages
- ❑ Electronic data interchange (EDI)
- ❑ Printers

- ❑ Generally performed after System Testing or in parallel.

System Integration Testing - Approaches

- Risk Based:
 - Identify the System Integration points that are critical to the business
 - Prioritize them and test sequentially
- “Divide and conquer”:
 - Test the outside first (at the interface to your system, e.g. test a package on its own)
 - Test the connections one at a time first (your system and one other)
 - Combine incrementally - safer than “big bang” (non-incremental)

System Integration Testing—Methods

- Who performs System Integration Testing?
 - Independent testing team

- Test artefacts created are:
 - System Integration Test cases
 - System Integration Test logs
 - System Integration Testing defect report

User Acceptance Testing

- Testing done by end users allowing them to verify day- to-day business scenarios and validate if the system developments satisfies their needs.
- Validates if the software meets a set of agreed acceptance criteria.
- Validates if the application is fit for deployment.
- Performed generally in the customer's (more close to the actual) environment.
- Performed after the ST, SIT, or in parallel.
- Helps assure that the Software application meets the needs of the end users/business or not.
- Helps determine whether to accept a software product.

User Acceptance Testing

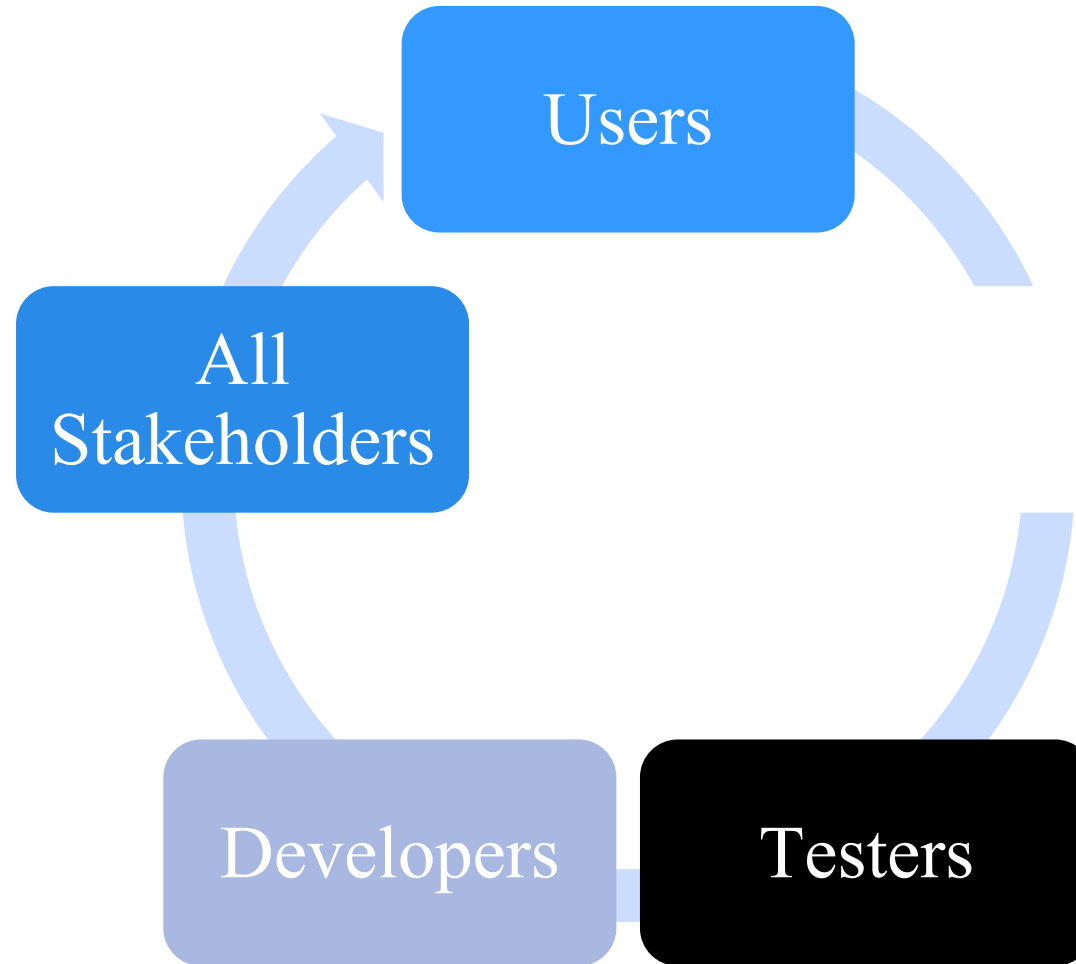
- Who performs Acceptance Testing?
 - Customers/Business Users/independent testing team (with high business knowledge) which is identified to augment the customer's requirements.

- Test artefacts created are:
 - Acceptance Test cases
 - Acceptance Test logs
 - Acceptance Testing defect report

User Acceptance Testing

- It will be performed in two ways
 - Alpha Testing
 - Beta Testing

People Who Do The Testing





1. Which of these levels of testing is typically done by an independent test team?

Unit Testing

Integration Testing

System Testing

User Acceptance
Testing



2. During which of these levels of testing, is the knowledge of programming languages required?

Unit Testing

Integration Testing

System Testing

User Acceptance Testing

3. Which testing is used to check data-flow between the applications

Unit Testing

Integration Testing

System Testing

User Acceptance Testing

4. In which type
of testing client
is involved

Unit Testing

Integration Testing

System Testing

User Acceptance
Testing

5. Which type
of testing is
getting
conducted in
production
Environment


Unit Testing

Integration Testing

System Testing

Alpha Testing

Beta Testing

- 
- Unit Testing
 - Integration Testing
 - System Testing
 - User Acceptance Testing