Integration Testing

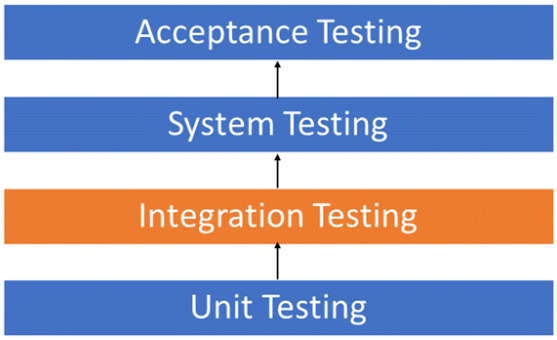
What is Integration Testing?

Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group.

A typical software project consists of multiple software modules, coded by different programmers. Integration Testing focuses on checking data communication amongst these modules.

Hence it is also termed as 'I & T' (Integration and Testing), 'String Testing' and sometimes 'Thread Testing'.

Why do Integration Testing?



Although each software module is unit tested, defects still exist for various reasons like:

- 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
- 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.
- Interfaces of the software modules with the database could be erroneous
- External Hardware interfaces, if any, could be erroneous
- Inadequate exception handling could cause issues.

Approaches/Methodologies/Strategies of Integration Testing:

Software Engineering defines variety of strategies to execute Integration testing, viz.

- Big Bang Approach
- Incremental Approach: which is further divided into the following
    1. Top Down Approach
    2. Bottom Up Approach
    3. Sandwich Approach - Combination of Top Down and Bottom Up

Below are the different strategies, the way they are executed and their limitations as well advantages.

Big Bang Approach: Here all components are integrated together at once and then tested.

Advantages: Convenient for small systems.

Disadvantages:

- Fault Localization is difficult.
- Given the sheer number of interfaces that need to be tested in this approach, some interfaces link to be tested could be missed easily.
- Since the Integration testing can commence only after "all" the modules are designed, the testing team will have less time for execution in the testing phase.
- Since all modules are tested at once, high-risk critical modules are not isolated and tested on priority. Peripheral modules which deal with user interfaces are also not isolated and tested on priority.

Incremental Approach: In this approach, testing is done by joining two or more modules that are logically related. Then the other related modules are added and tested for the proper functioning. The process continues until all of the modules are joined and tested successfully.

Incremental Approach, in turn, is carried out by two different Methods:

- Bottom Up
- Top Down

What is Stub and Driver?

Incremental Approach is carried out by using dummy programs called Stubs and Drivers. Stubs and Drivers do not implement the entire programming logic of the software module but just simulate data communication with the calling module.

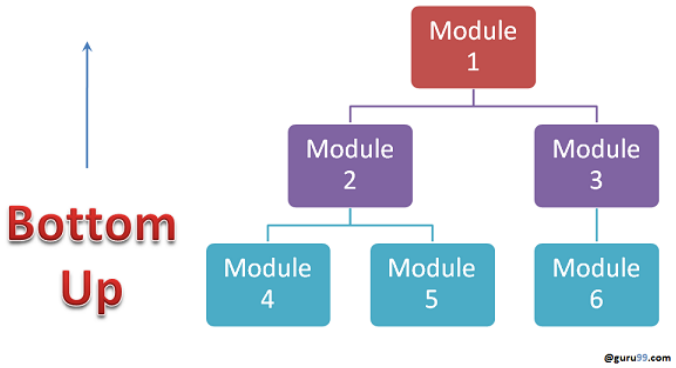
Stub: Is called by the Module under Test.

Driver: Calls the Module to be tested.

Bottom-up Integration

In the bottom-up strategy, each module at lower levels is tested with higher modules until all modules are tested. It takes help of Drivers for testing

Diagrammatic Representation:



Advantages:

- Fault localization is easier.
- No time is wasted waiting for all modules to be developed unlike Big-bang approach

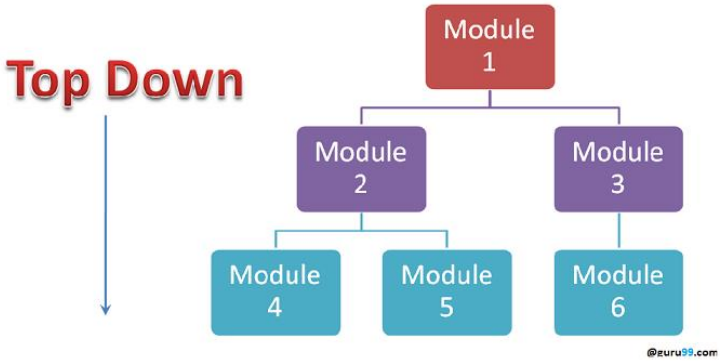
Disadvantages:

- Critical modules (at the top level of software architecture) which control the flow of application are tested last and may be prone to defects.
- An early prototype is not possible
-

Top-down Integration

In Top to down approach, testing takes place from top to down following the control flow of the software system. Take help of stubs for testing.

Diagrammatic Representation:



Advantages:

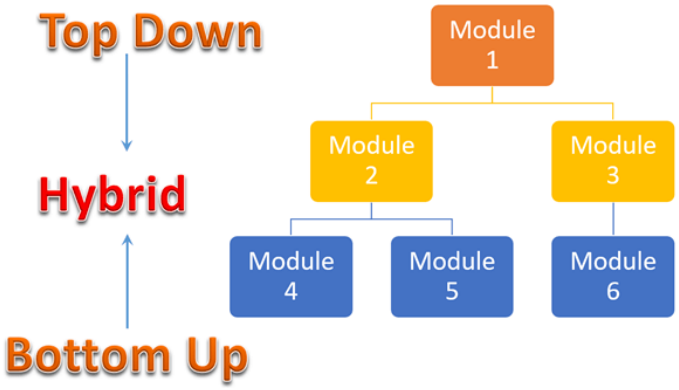
- Fault Localization is easier.
- Possibility to obtain an early prototype.
- Critical Modules are tested on priority; major design flaws could be found and fixed first.

Disadvantages:

- Needs many Stubs.
- Modules at a lower level are tested inadequately.

Hybrid/ Sandwich Integration

In the sandwich/hybrid strategy is a combination of Top Down and Bottom up approaches. Here, top modules are tested with lower modules at the same time lower modules are integrated with top modules and tested. This strategy makes use of stubs as well as drivers.



How to do Integration Testing?

- The Integration test procedure irrespective of the Software testing strategies (discussed above):
- Prepare the Integration Tests Plan
- Design the Test Scenarios, Cases, and Scripts.
- Executing the test Cases followed by reporting the defects.
- Tracking & re-testing the defects.
- Steps 3 and 4 are repeated until the completion of Integration is successful.