Unit: IV
Lecture: 7
Integration Testing

Integration testing is the process of testing the interface between two software units or module. It's focus on determining the correctness of the interface. The purpose of the integration testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integration testing is performed.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

The objective is to take unit tested components and build a program structure that has been dictated by design.

Integration test approaches:

There are four types of integration testing approaches. Those approaches are the following:

1. Big-Bang Integration Testing:

It is the simplest integration testing approach, where all the modules are combining and verifying the functionality after the completion of individual module testing.
In simple words, all the modules of the system are simply put together and tested. This approach is practicable only for very small systems.
If once an error is found during the integration testing, it is very difficult to localize the error as the error may potentially belong to any of the modules being integrated.
So, debugging errors reported during big bang integration testing are very expensive to fix.

Advantages:

- It is convenient for small systems.

Disadvantages:

- There will be quite a lot of delay because you would have to wait for all the modules to be integrated.
- High risk critical modules are not isolated and tested on priority since all modules are tested at once.

2. Bottom-Up Integration Testing:

In bottom-up testing, each module at lower levels is tested with higher modules until all modules are tested.
The primary purpose of this integration testing is, each subsystem is to test the interfaces among various modules making up the subsystem. This integration testing uses test drivers to drive and pass appropriate data to the lower level modules.

Advantages:

- In bottom-up testing, no stubs are required.
- A principle advantage of this integration testing is that several disjoint subsystems can be tested simultaneously.

Disadvantages:

- Driver modules must be produced.
- In this testing, the complexity that occurs when the system is made up of a large number of small subsystem.

3. Top-Down Integration Testing:

Top-down integration testing technique used in order to simulate the behaviour of the lower-level modules that are not yet integrated. In this integration testing, testing takes place from top to bottom. First high-level modules are tested and then low-level modules and finally integrating the low-level modules to a high level to ensure the system is working as intended.

Advantages:

- Separately debugged module.
- Few or no drivers needed.
- It is more stable and accurate at the aggregate level.

Disadvantages:

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

4. Mixed Integration Testing:

A mixed integration testing is also called sandwiched integration testing.
A mixed integration testing follows a combination of top down and bottom-up testing approaches. In top-down approach, testing can start only after the

top-level module have been coded and unit tested. In bottom-up approach, testing can start only after the bottom level modules are ready.
This sandwich or mixed approach overcomes this shortcoming of the top-down and bottom-up approaches

Advantages:

- Mixed approach is useful for very large projects having several sub projects.
- This Sandwich approach overcomes this shortcoming of the top-down and bottom-up approaches.

Disadvantages:

- For mixed integration testing, require very high cost because one part has Top-down approach while another part has bottom-up approach.
- This integration testing cannot be used for smaller system with huge interdependence between different modules.