# 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. Test drivers and test stubs are used to assist in Integration Testing.

Definition by ISTQB

- integration testing: Testing performed to expose defects in the interfaces and in the

interactions between integrated components or systems. See also component integration

testing, system integration testing.

- component integration testing: Testing performed to expose defects in the interfaces and interaction between integrated components.
- system integration testing: Testing the integration of systems and packages; testing

interfaces to external organizations (e.g. Electronic Data Interchange, Internet).

Analogy

During the process of manufacturing a ballpoint pen, the cap, the body, the tail and clip, the ink cartridge and the ballpoint are produced separately and unit tested separately. When two or more units are ready, they are assembled and Integration Testing is performed. For example, whether the cap fits into the body or not.

Method

Any of Black Box Testing, White Box Testing and Gray Box Testing methods can be used. Normally, the method depends on your definition of ‘unit’.

Tasks

1. Integration Test Plan

- Prepare
- Review
- Rework
- Baseline

2. Integration Test Cases/Scripts

- Prepare
- Review
- Rework

- Baseline

3. Integration Test

- Perform

When is Integration Testing performed?

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

Who performs Integration Testing?

Developers themselves or independent testers perform Integration Testing.

Approaches

- Big Bang is an approach to Integration Testing where all or most of the units are combined together and tested at one go. This approach is taken when the testing team receives the entire software in a bundle. So what is the difference between Big Bang Integration Testing and System Testing? Well, the former tests only the interactions between the units while the latter tests the entire system.

- Top Down is an approach to Integration Testing where top-level units are tested first and lower level units are tested step by step after that. This approach is taken when top-down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

- Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper-level units step by step after that. This approach is taken when bottom-up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

- Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches.

Tips

- Ensure that you have a proper Detail Design document where interactions between each unit are clearly defined. In fact, you will not be able to perform Integration Testing without this information.

- Ensure that you have a robust Software Configuration Management system in place. Or else, you will have a tough time tracking the right version of each unit, especially if the number of units to be integrated is huge.

- Make sure that each unit is unit tested before you start Integration Testing.

- As far as possible, automate your tests, especially when you use the Top Down or Bottom Up approach, since regression testing is important each time you integrate a unit, and manual regression testing can be inefficient.

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

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.

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.

Brief Description of Integration Test Plans:

It includes the following attributes:

- Methods/Approaches to testing (as discussed above).
- Scopes and Out of Scopes Items of Integration Testing.
- Roles and Responsibilities.
- Pre-requisites for Integration testing.
- Testing environment.
- Risk and Mitigation Plans.

Entry and Exit Criteria to Integration testing phase in any software development model

Entry Criteria:

- Unit Tested Components/Modules
- All High prioritized bugs fixed and closed
- All Modules to be code completed and integrated successfully.
- Integration tests Plan, test case, scenarios to be signed off and documented.
- Required Test Environment to be set up for Integration testing

Exit Criteria:

- Successful Testing of Integrated Application.
- Executed Test Cases are documented
- All High prioritized bugs fixed and closed
- Technical documents to be submitted followed by release Notes.

Best Practices/ Guidelines for Integration Testing

- First, determine the Integration Test Strategy that could be adopted and later prepare the test cases and test data accordingly.
- Study the Architecture design of the Application and identify the Critical Modules. These need to be tested on priority.
- Obtain the interface designs from the Architectural team and create test cases to verify all of the interfaces in detail. Interface to database/external hardware/software application must be tested in detail.
- After the test cases, it's the test data which plays the critical role.
- Always have the mock data prepared, prior to executing. Do not select test data while executing the test cases.