Software Testing Mentor

www.softwaretestingmentor.com

ISTQB Foundation Level and Software Testing Training

Module 2
Testing throughout the software life cycle

Session 2 – Test Levels

Test Levels

In this session we will understand test levels in more detail

- Component Testing or Unit Testing
- Integration Testing
- System Testing
- Acceptance testing

Component Testing/Unit Testing

Unit is the smallest testable part of the software system.

Unit testing is done to verify that the lowest independent entities in any software are working fine.

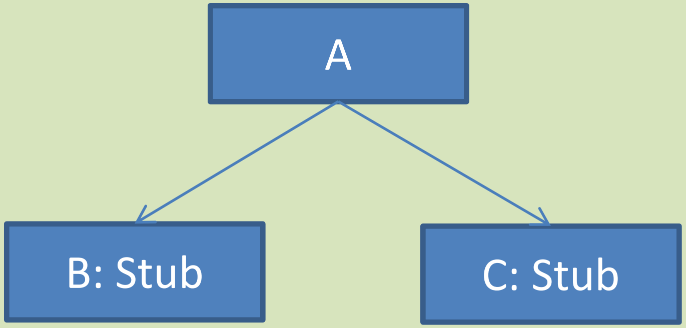
The smallest testable part is isolated from the remainder code and tested to determine whether it works correctly.

STUBS and DRIVERS are used to replace the missing components in unit testing

STUBS and DRIVERS

A STUB is called from the software component

Example:

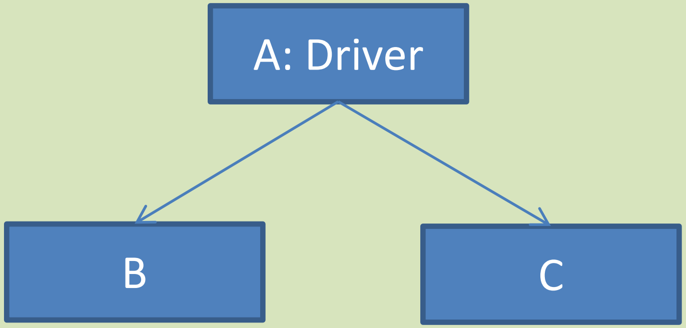


- Suppose module A calls functions from Module B and C which are not ready.
- Developer will write a dummy module which simulates B and C and returns values to module A.
- This dummy module code is known as stub.

STUBS and DRIVERS Cont.

A DRIVER calls a component to be tested

Example



- Suppose you have modules B and C ready but module A which calls module B and C is not ready
- Developer writes a dummy piece of code for module A which takes care of the call for module A
- This dummy piece of code is known as driver.

Component Testing/Unit Testing

Component testing may include testing of functional & non-functional characteristics like:

- Resource behavior (e.g. memory leaks)
- Robustness testing (e.g. invalid inputs or stressful environmental conditions)
- Structural testing (e.g. branch coverage)

Component Testing/Unit Testing

Component testing is typically done by developers

Needs access to the code & done with the support of the development environment

Defects are typically fixed as soon as they are found

No formal defect logging process is followed.

Integration Testing

Integration testing tests interfaces between components

In integration testing testers concentrate only on the integration

Testers should build integration tests in the order required for most efficient integration testing

Integration testing tests interactions to different parts of a system like, File system, System Hardware, Operating system, Interfaces between other systems

Levels of Integration Testing

There may be more than one level of integration testing, for example

- Component Integration Testing
- System Integration Testing

Component Integration Testing

Component integration testing tests the interactions between software components

Component integration testing is done after Component /Unit testing

System Integration Testing

System integration testing tests the interactions between different systems

Done mostly after system testing is complete

Done to identify the cross-platform issues which may arise after integrating the system to other system

Approaches of Integration Testing

Big bang integration testing

Incremental Integration Testing

- Top-down approach
- Bottom-up approach
- Functional incremental

Big bang integration testing

All components or systems are integrated simultaneously, after which everything is tested as a whole

Advantage of big bang approach is that everything is finished before integration testing starts. So, STUBS and DRIVERS are not required

Disadvantage is that it is difficult to trace cause of failures due to late integration

Incremental Integration Testing

All programs are integrated one by one and testing is done after every step

Advantage is that defects are found early and easy to find root cause for those defects

Disadvantage is that it is time consuming because STUBS and DRIVERS are required

Top down approach

Testing takes place from top to bottom

Follows the control flow or architectural structure (e.g. It starts from the GUI or main menu).

The top down approach utilises STUBS to replace the unfinished components

The top level component of the hierarchy is tested first with the lower unfinished components it integrates with being replaced by STUBS

Testing continues down the hierarchy by replacing the stubs with code

Bottom up approach

Takes place from the bottom of the control flow upwards

The lowest level components are tested first

Missing components are substituted with DRIVERS

As soon as code for other module gets ready, these drivers are replaced with the actual module.

System Testing

Testing the behavior of the whole software/system as defined by scope is known as system testing

Purpose of system testing is to find as many defects as possible

It is often carried out by specialist testers that form an independent test team

The system test environment should correspond to the production environment as much as possible to minimize the risk of environment-specific failures not found by previous testing cycles

System Testing

System testing should investigate both functional and non-functional requirements of the system

Black box techniques (Specification – based) should be used to system test the functional requirements

White box techniques (Structure-based techniques) may then be used to assess the thoroughness of the testing

Test cases for System Testing

System testing includes test cases based on:

- Software Requirements specifications (SRS)
- Business processes
- End user scenarios
- High level description of system behaviors
- Interactions with the operating system
- Risks involved for the system

Acceptance Testing

Acceptance testing starts after system testing is done and all major defects are resolved

Acceptance testing is most often responsibility of customer but other stakeholders are also involved

The main goal in acceptance testing is to establish confidence in the system

Test environment for acceptance testing should be in most aspects representative of production environment (as-if production)

Finding defects should not be the main focus of acceptance testing, its main focus should be to determine if the system is fit for purpose.

Acceptance Testing

Acceptance testing occurs at more than just a single test level.

For example:

- A Commercial off the shelf (COTS) software product may be acceptance tested when it is installed or integrated
- Acceptance testing of the usability of a component may be done during component testing
- Acceptance testing of a new functional enhancement may come before system testing

Forms of Acceptance Testing

Acceptance Testing can be classified into different forms like:

- User acceptance testing
- Operational acceptance testing
- Contract acceptance testing
- Compliance acceptance testing
- Alpha testing
- Beta testing

User Acceptance Testing

Focuses on functionality and validates fitness-for-use of the system

Business users and other stakeholders should be heavily involved

UAT Test environment should be very close to the production environment

Operational Acceptance Testing

Operational acceptance testing validates if the system meets operational requirements, it is also known as “Production Acceptance Testing”

Some of the tests to include in operational acceptance testing are:

- Testing backup/restore
- High availability & Disaster recovery
- User management
- Testing maintenance tasks
- Periodic checks of security vulnerabilities

Contract Acceptance Testing

It is performed against the contracts acceptance criteria

Acceptance criteria for contract acceptance testing is formally defined before signing the contract

Compliance Acceptance Testing

Compliance acceptance testing is also know and regulation acceptance testing.

Compliance acceptance testing is performed against the regulations (Government, legal, safety, medical) which must be adhered to.

Alpha Testing

Alpha testing is done at the developers site by independent test team, potential end users and stakeholders.

Alpha Testing is mostly done for COTS(Commercial Off the Shelf) software to ensure internal acceptance before moving the software for beta testing.

Alpha testing provides feedback to the developers directly from the customers

Beta Testing or Field Testing

Beta Testing is done after alpha testing.

Beta testing is done by the potential or existing users, customers and end users at the external site without developers involvement.

Beta testing is done to acquire feedback from mass market

Conclusion

To conclude in this session we learned about different test levels in detail.

- Component Testing or Unit Testing
- Integration Testing
- System Testing
- Acceptance testing

Thank You!!!