

Unit-4

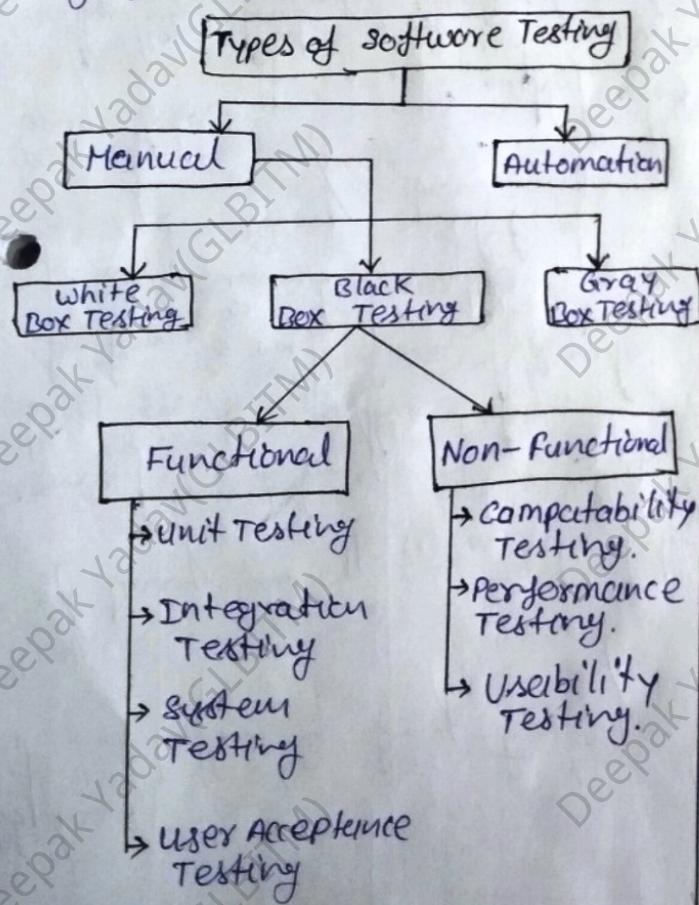
Testing Concept

Software testing is a process of identifying the correctness of SW by considering its all attributes and evaluating the execution of SW components to find the SW bugs or errors or defects.

- It is a method to check whether the actual SW product matches expected requirements and to ensure that the actual SW product is defect free.
- Some prefer saying SW testing definition as a White Box and Black Box Testing.

Testing objectives

- ① Finding defect which may get created by the programmer while developing the SW.
- ② Gaining confidence in and providing information about the level of quality.
- ③ To prevent defects.
- ④ To make sure that the end result fits the business and user requirements.
- ⑤ The objective of the testing is finding as many software defects as possible.
- ⑥ Ensure that the SW under test is bug free before release.



Unit Testing

- Unit Testing is the first level of SW testing, which is used to test if SW modules are satisfying the given requirement or not.
- The first level of testing involves analyzing each unit or an individual components of the SW application.
- It is a white box testing technique that is usually performed by the developer.
- Unit Testing is important because SW developers sometimes try saving time doing minimal unit testing and this is a myth because inappropriate unit testing leads to high cost defect fixing during System Testing, Integration testing and even Beta Testing after application is built.
- If proper unit testing is done in early development, then it saves time and money in the end.

Advantages

- Unit testing allows the programmer to refactor code at a later date, and make sure that the module still works correctly.
- The procedure is to write test cases for all functions and methods so that whenever a change causes a fault, it can be quickly identified and fixed.

Disadvantages:-

- Unit testing by its very nature focuses on a unit of code. Hence it can't catch integration errors or broad system level errors.

Integration Testing

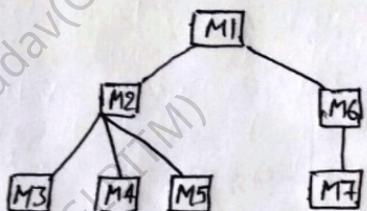
- Integration testing is the phase in SW testing in which individual SW modules are combined and tested as a group.
- A typical SW project consists of multiple SW modules, coded by different programmers.
- It occurs after unit testing and before validating.
- The purpose of this level of testing is to expose defects in the integration b/w the SW modules when they are integrated.

The various S/W integration techniques:-

- Top-Down integration Testing.
- Bottom-up integration Testing.
- Big-Bang testing.

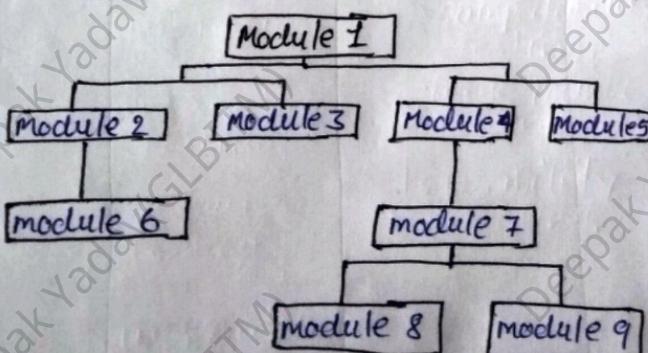
① Top-Down Integration Testing

- Top-down testing is a type of increment testing approach in which testing is done by integrating or joining two or more modules by moving down from top to down bottom through control flow of architecture structure.
- In these, high-level modules are tested first and then low-level modules are tested.
- Integration is done to ensure that system is working properly.
- Stubs and drivers are used to carry out this project.



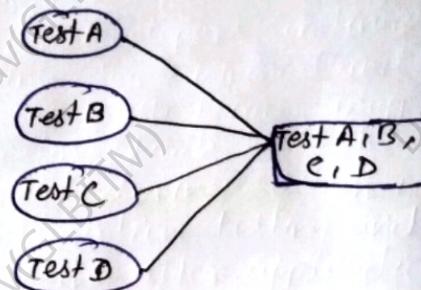
② Bottom-up Integration Testing

- This type of testing method deals with how lower-level modules until all the modules have been tested successfully.
- In bottom-up testing, the top-level critical modules are tested at last. Hence it may cause a defect.
- In simple words, we can say that we will be adding the modules from the bottom to the top.



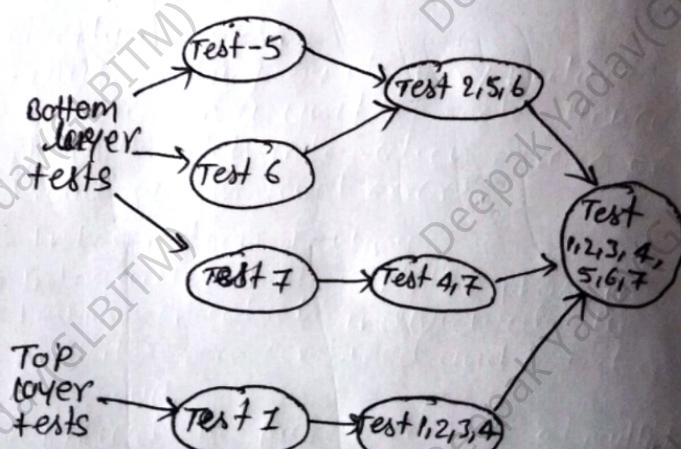
③ Big-Bang Integration Testing

- In this approach, the modules of the system are integrated only after integrating all the modules are complete. After integration, testing is carried out on the whole system to check for its working.
- Only one round of SIT is required.
- It is difficult to find the root cause of an error.



④ Sandwich Integration Testing

- Combination of Top-Down and Bottom-up.
- It is called Hybrid/Hybrid Integration testing.
- It makes use of both stubs as well as drivers.
- Sandwich Testing approach is used in very large projects having subprojects.
- It allows parallel testing.
- Sandwich testing is time-saving approach.



Acceptance testing

- Acceptance testing is a level of S/W testing where a system is tested for acceptability.
- Testing done by users, customers, or other authorized entities to determine

application / software needs and business processes.

- Acceptance testing is the most important phase of testing as this decides whether the client approves the application / SW or not.
- Acceptance testing is the last phase of SW testing performed after system testing and before making the system available for actual use.

Steps to perform Acceptance Testing

- Requirement Analysis.
- Test Plan Creation.
- Test Case Designing.
- Test Case Execution
- Confirmation of objectives.

Regression Testing

- Regression testing is a SW testing practice that ensures an application still functions as expected after any code changes, updates or improvements.
- It verifies the product behaviour as a whole.
- Regression testing verifies that recent code changes haven't destroyed the already existing functionality of a system. Regression testing examples include iteration regression and full regression and both can be covered with manual and automated test cases.

Need of Regression Testing

- It ensures that the fixed bugs & issues do not reoccur.
- New features is added to the SW.
- Defect fixing.
- Performance issue fixing.

Error:-

- Error is deviation from actual and expected value.
- It represents mistake made by people.

Fault

- Fault is incorrect step, process or data definition in a computer program which causes the program to behave in an unintended or unanticipated manner.

- It is the result of the error.

Failure

- Failure is the ~~not~~ inability of a system or a component to perform its required functions within specified performance.
- Failure occurs when fault executes.

Drivers

Test Drivers and Test Stubs

- The stubs and drivers are considered as elements which are equivalents which to ~~do~~ to-do modules that could be replaced if modules are in their developing stage, missing or not developed yet, ~~so~~ so that necessity of such modules could be met.
- Drivers and stubs simulate features and functionalities and have ability to serve features that a module can provide.
- This reduces useless delay in testing and makes the testing process faster.
- Stubs are mainly used in Top-Down integration testing while the Drivers are used in Bottom-up integration testing, thus increasing the efficiency of testing process.

Stubs

- Stubs are developed by SW developers to use them in place of modules, if the respective modules aren't developed, missing in developing stage, or are unavailable currently while Top-down testing of modules.
- Stubs are used when the lower-level modules are needed but are unavailable currently.

Stubs are divided into four basic categories based on what they do:-

- Shows the traced messages.
- Shows the displayed messages if any.
- Returns the corresponding values that are utilized by modules.
- Returns the value of the chosen parameters (arguments) that were used by the testing modules.

Drivers

- Drivers serve the same purpose as stubs, but drivers are used in Bottom-up integration testing and are also more complex than stubs.
- Drivers are also used when some modules are missing and unavailable at time of testing of a specific module because of some unavoidable reasons to act in absence of required module.
- Drivers are used when high-level modules are missing and can also be used when lower level modules are missing.

Structural Testing (White Box Testing)

- White Box Testing is SW testing technique in which internal structure, design and coding of SW are tested, code to verify flow of I/P-O/P and to improve design, usability and security. In white box testing code is visible to testers so it is also called clear box testing, open box testing, transparent box testing, code-based testing and glass box testing.
- Structure-based testing techniques is also known as "white-box" or "glass-box" testing technique because here the testers require knowledge of how the SW is implemented, how it works. In white-box testing approach the tester is concentrating on how SW does it.
- Code optimization by finding hidden errors.
- White box test cases can be easily automated.

Statement Coverage

- Testing every possible statement in the code is executed at least once.
- Tools:- To test the statement coverage the **cantate++** can be used.

Decision coverage

- Testing every possible decision conditions like if-else, for loop and other conditions loops in the code is executed at least once.

- Tools:- TCAT-PATH.

Multiple condition coverage

- In this testing we ensure that each entry point of the system is executed once.
- In the actual development process developers make use of the combination of techniques those are suitable for their SW application.

Functional Testing (Black Box Testing)

- Functional testing check app's functionality without looking at the internal structure of the code, hence it's called black box testing.
- Black box testing is a method of SW testing that examines that functionality of an applications. It is also known as specifications based testing.
- This method of test can be applied to each and every level of SW testing such as unit, integration, system and acceptance testing.
- Well suited and sufficient for large code segments.
- Code access is not required.

Alpha Testing

- Alpha testing is one of the most common SW testing strategies used in SW development. It is specially used by product development organization.
- This test takes place at the developer's site. Developers observe the users and note problems.
- Alpha testing is testing of an application when development is about to complete. Minor design changes can still be made as a result of alpha testing.

Advantages

- Provides better view about the reliability of the SW at an early stage.

- Helps simulate real time user behavior and environment.

Disadvantages

- In depth functionality of the SW cannot be tested as it is still under development stage.

Beta Testing

- Beta Testing is also known as field testing. It takes place at a customer's site. It sends the system SW to users who install it and use it under real-world working conditions.
- Beta Testing of a product is performed by real-users "of the SW applications in a real-environment" and can be considered as a form of external User Acceptance Testing.
- Beta version of the SW is released to a limited number of end-users of the product to obtain feedback on the product quality. Beta testing reduces product failure risks and provides increased quality of the product through customer validation.

Advantages

- Reduces product failure risk via customer validation.
- Improves product quality via customer feedback.

Disadvantages

- Finding the right beta users and maintaining their participation could be a challenge.

Alpha Test	Beta Test
• Performed by developers.	• Performed by customers.
• It's conducted for SW application.	• It's conducted for product.
• Performed in virtual environment.	• Performed in Real Environment.
• Involve both black and white box testing.	• Involve both black box testing only.

Formal Technical Reviews (Peer Reviews)

- The focus of FTR is on a work product that is requirement specification, a detailed component design, a source code listing for a component.
- The individual who has developed the work product i.e. the producer informs the project leader that the work product is complete and that a review is required.
- The project leader contacts a review leader, who evaluates the product for readiness, generates a copy of product material and distributes them to two or three review members for advance preparation.
- Each reviewer is expected to spend b/w one and two hours reviewing the product making notes.
- The reviews meeting is attended by the review leader, all reviewers and the producer. One of the reviewers act as a recorder, who notes down all important points discussed in the meeting.
- The meeting (FTR) is started by introducing the agenda of the meeting and then the producer introduces his product.

Objectives

- Useful to recover error in logic, function and implementation for any representations of the software.
- The purpose of FTR is to verify that the SW meets specified requirement.
- To ensure that software is represented according to predefined standards.
- It helps to review the uniformity in SW that is developed in a uniform manner.
- To make the product more manageable.

Walk Through

- Walkthrough is an activity in which author describes and explains the work product in an informal meeting to his peers or supervisors to get feedback.
- In a walkthrough, the programmer who wrote the code formally presents it to a small group of five or so other programmers and testers.
- The reviewers should receive copies of the SW in advance of the review. Having at least one senior programmer as a

reviewer. Having it is very important.

- The presenter reads through the code line by line, or function by function, explaining what the code does and why.
- The reviewers listen and question anything that looks suspicious.
- It's also very important that after the review the presenter writes a report telling what was found and how he plans to address address any bugs discovered.

A walkthrough has two broad objectives

- To gain feedback about the technical quality or content of the document.
- To familiarize the audience with the content.

Code Inspection.

- Code inspection is a process of examining the code of program for identification of certain errors which are not identified by the code walkthrough.
- Code inspection is done to find the out some common types of errors caused due to misunderstanding and improper programming.
- During identifying the errors through code inspection, the standard of coding is also checked.

Programming errors can be removed during code inspection:-

- Use of un-initialized variables.
- Jumps with loop.
- Non-terminating loops.
- Mismatched assignment.
- Size of array is not initialized.
- Improper allocation and de-allocation.

Compliance with Design

- Compliance Testing is performed to maintain and validate the compliant state for the life of the SW. Every industry has a regulatory

and compliance board that protects the end users.

- Software compliance refers to how well an application obeys the rules in a standard.

Checklists :-

- Professionals, who are knowledgeable and experienced, who understand the compliance must be retained.
- Understanding the risks and impacts of being non-compliant.
- Document the processes and follow them.
- Perform an internal audit and follow with an action plan to fix the issues.

Coding Standards

Coding :-

- The objectives of the coding phase is to transform the design of a system into code in a high level language and then to unit test this code.
- Good SW development organizations normally require their programmers to adhere to some well-defined and standard style of coding called coding standards.

Coding standards :-

- A coding standards gives a uniform appearance to the codes written by different engineers.
- It enhances code understanding.
- It encourage good programming practice.
- It agree upon standards for coding styles.
- Promotes ease of understanding and uniformity.

Coding Guidelines :-

- Line length.
- Spacing.
- Codes well documented.
- Length not exceed 10 source line.
- Don't use goto statement.
- Inline comments.
- Error messages.

* The End *