

B.Tech

Software Engineering

KCS-601

With
Notes

UNIT-4

Software Testing

(in one video)

AKTU Exam

Topics to be covered...

Testing Concepts

Unit Testing

Integration Testing

Acceptance Testing

Regression Testing

Error vs Fault vs Failure

Test Drivers and Test Stubs

Structural Testing (White Box Testing)

Functional Testing (Black Box Testing)

Alpha and Beta Testing of Products

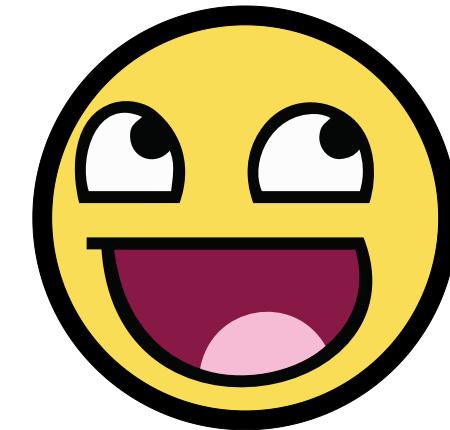
Formal Technical Reviews (Peer Reviews)

Walk Through

Code Inspection

Compliance with Design and Coding Standards

Happy Ending!



Engineering in One Video (EIOV)

Watch video on  YouTube



Testing Concepts



Testing Concepts

Software testing is a process of identifying the correctness of software by considering its all attributes (Reliability, Scalability, Portability, Re-usability, Usability) and evaluating the execution of software components to find the software bugs or errors or defects.

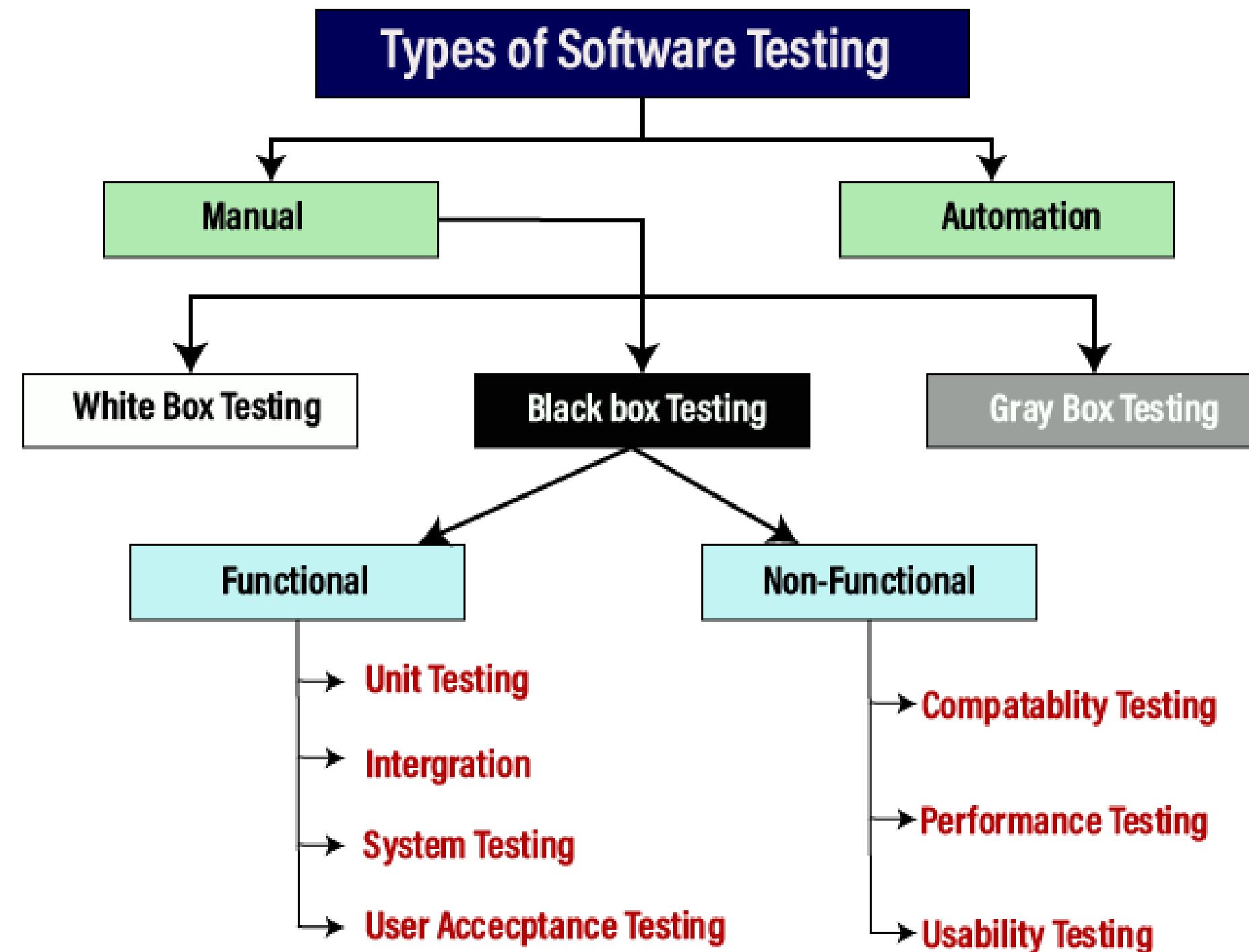
- It is a method to check whether the actual software product matches expected requirements and to ensure that the software product is Defect free.
- Some prefer saying Software testing definition as a White Box and Black Box Testing.

Testing Objectives:

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

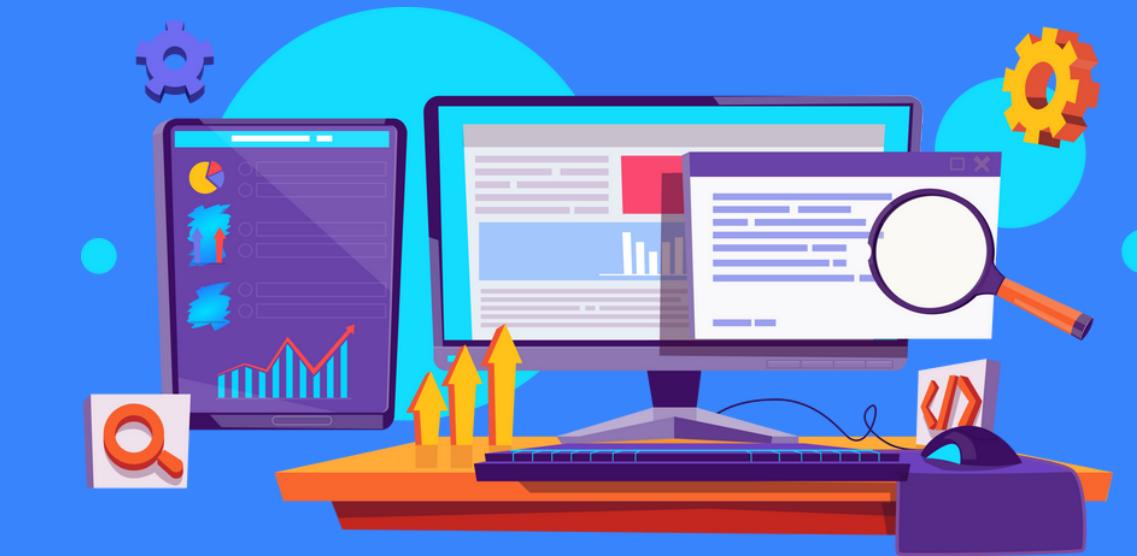


Testing Concepts



Engineering in One Video (EIOV)

Watch video on  YouTube



Unit Testing

Unit Testing

- Unit testing is the first level of software testing, which is used to test if software modules are satisfying the given requirement or not.
- The first level of testing involves analyzing each unit or an individual component of the software application.
- It is a WhiteBox testing technique that is usually performed by the developer.
- Unit Testing is important because software 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.

Myths about Unit Testing:

Truth is Unit testing increases the speed of development. Programmers think that Integration Testing will catch all errors and not execute the unit test. Once units are integrated, very simple errors which could have very easily found and fixed in unit tested take a very long time to be traced and fixed.

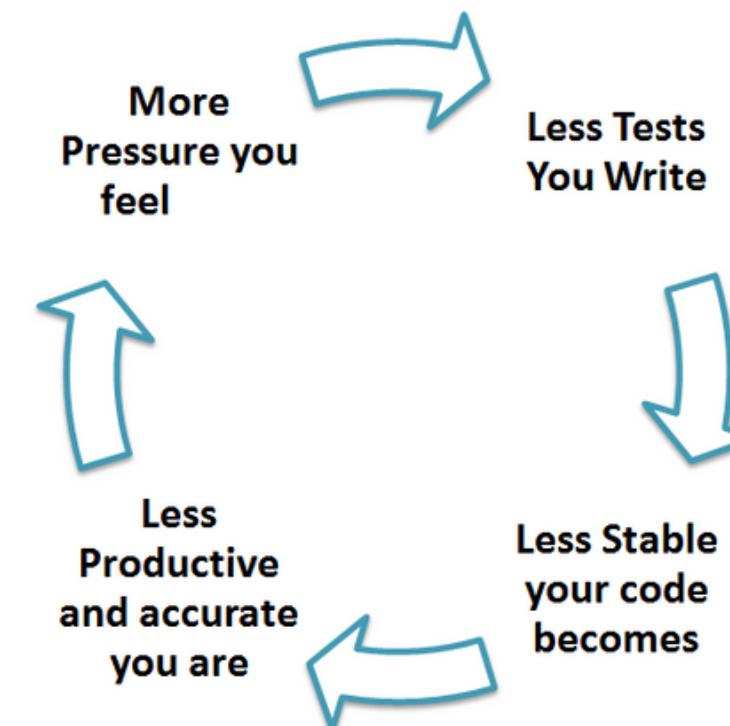
Unit Testing

Advantages:

- Unit testing allows the programmer to refactor code at a later date, and make sure the module still works correctly (i.e. Regression testing).
- 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.



Engineering in One Video (EIOV)

Watch video on  YouTube



Integration Testing

Integration Testing

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

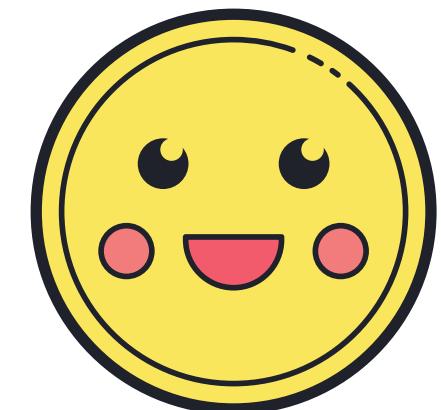
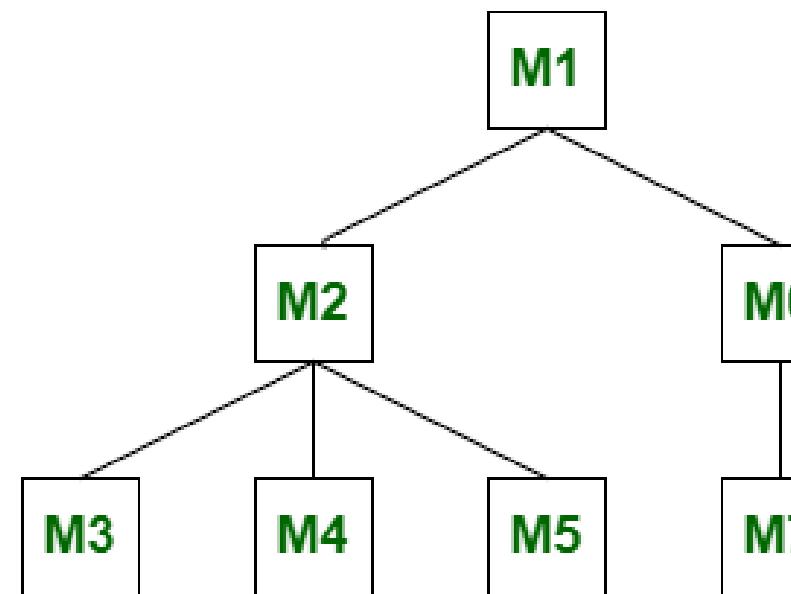
The various software integration techniques:

- Top-Down integration testing
- Bottom-up integration testing
- Big-bang testing



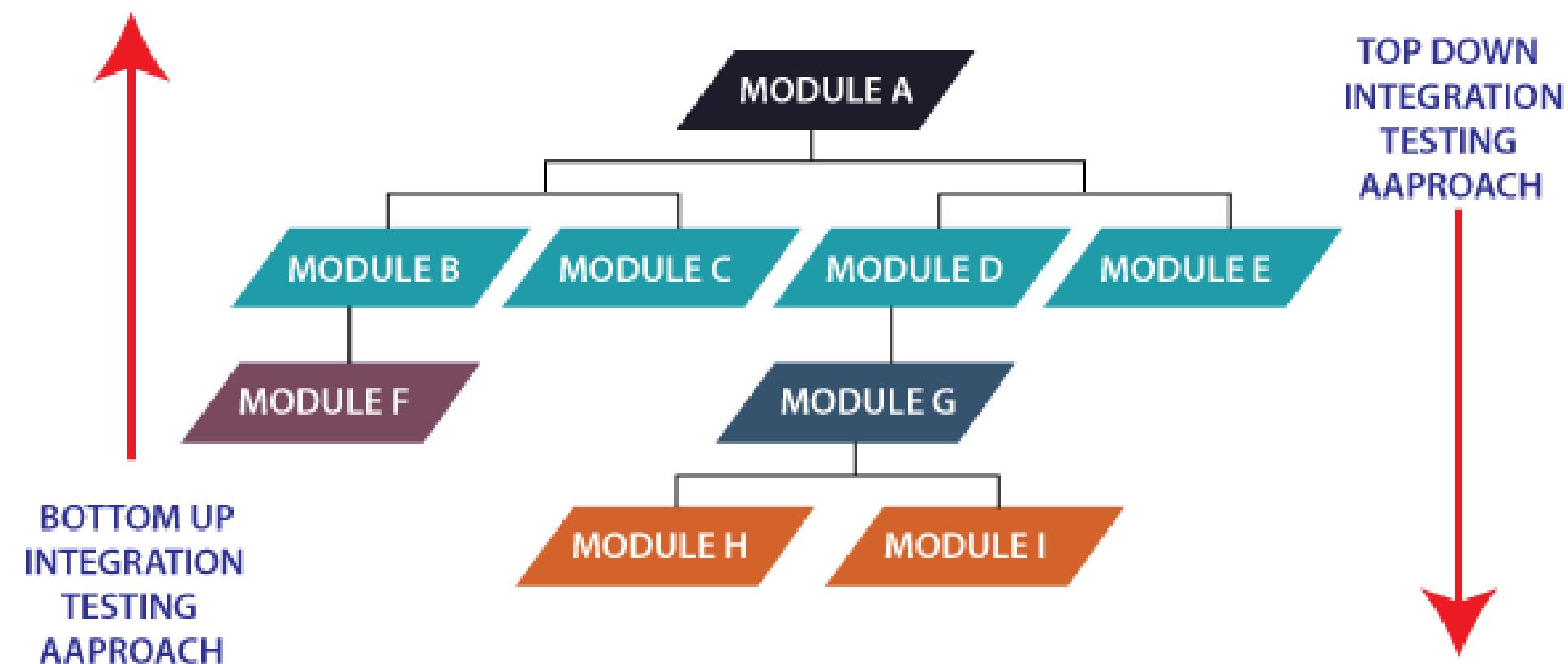
1. Top-Down Integration Testing

- Top-down testing is a type of incremental integration testing approach in which testing is done by integrating or joining two or more modules by moving down from top to 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.



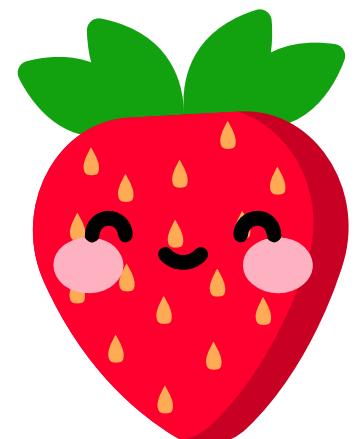
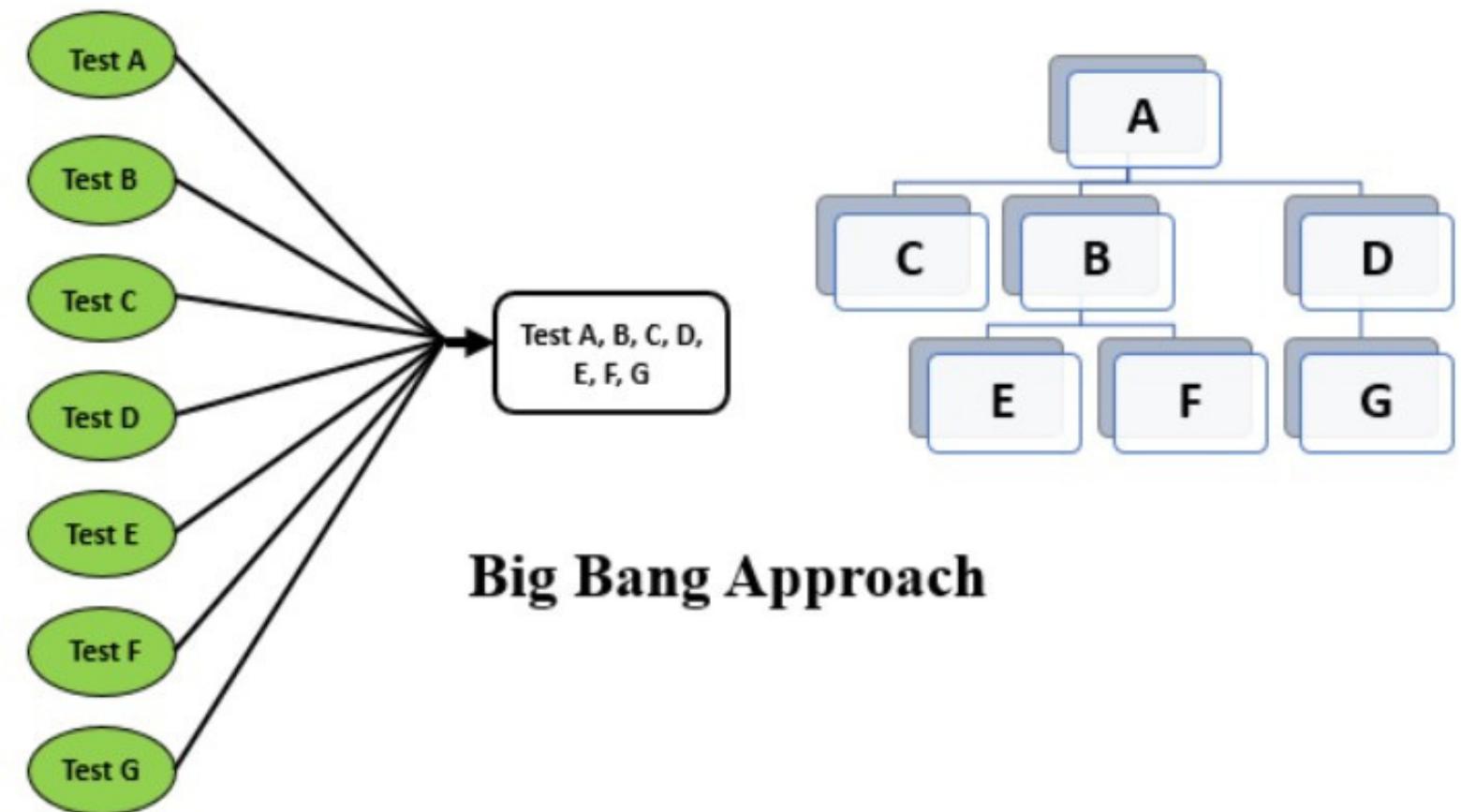
2. Bottom-up Integration Testing

- This type of testing method deals with how lower-level modules are tested with higher-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.



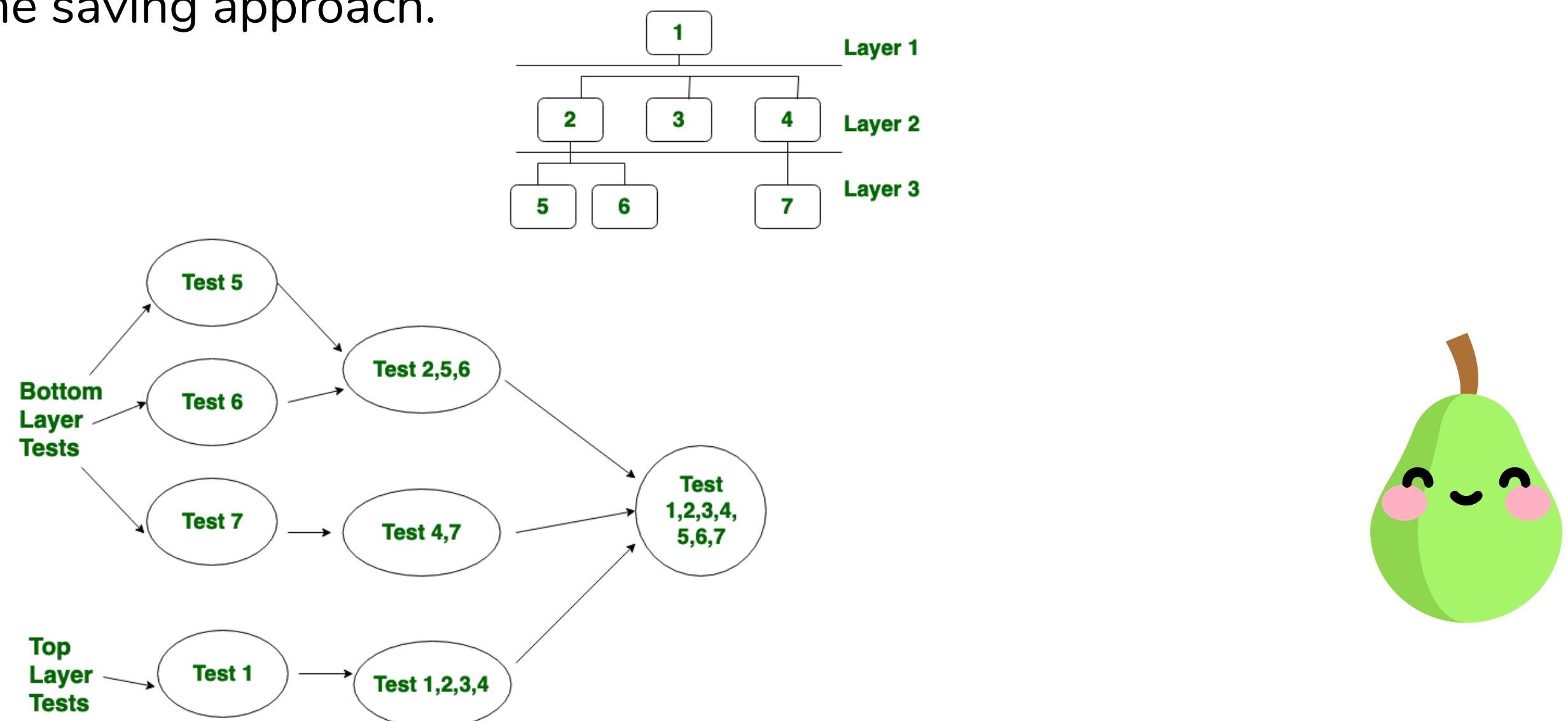
3. Big-bang Integration Testing

- In this approach, the modules of the system are integrated only after 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 Integration Testing.
- It makes use of both stubs as well as drivers.
- Sandwich Testing approach is used in very large projects having sub projects.
- It allows parallel testing.
- Sandwich testing is time saving approach.





Acceptance Testing

Acceptance Testing

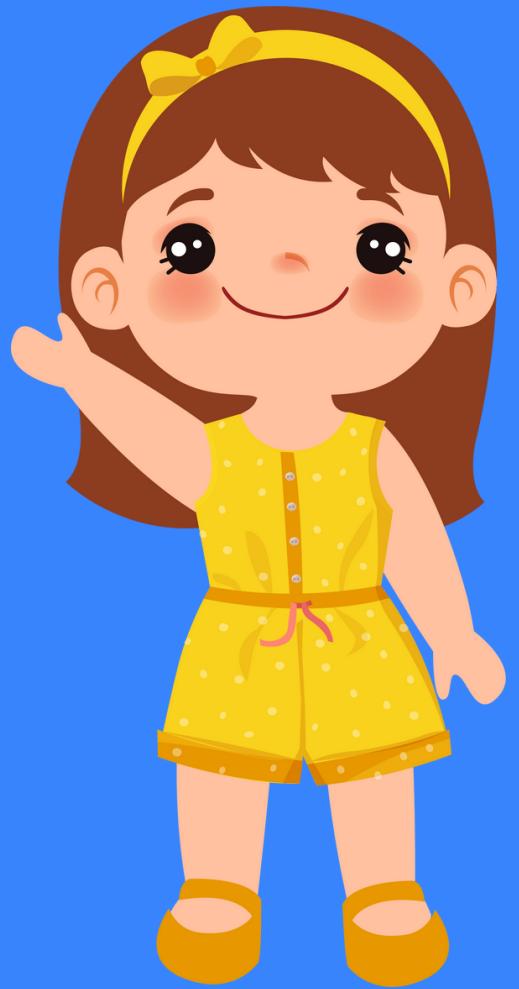
- Acceptance testing is a level of software testing where a system is tested for acceptability.
- testing done by users, customers, or other authorised entities to determine application/software needs and business processes.
- Acceptance testing is the most important phase of testing as this decides whether the client approves the application/software or not.
- Acceptance Testing is the last phase of software 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

- Regression testing is a software testing practice that ensures an application still functions as expected **after any code changes**, updates, or improvements. Regression testing is the final step, as it verifies the product behaviors 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.

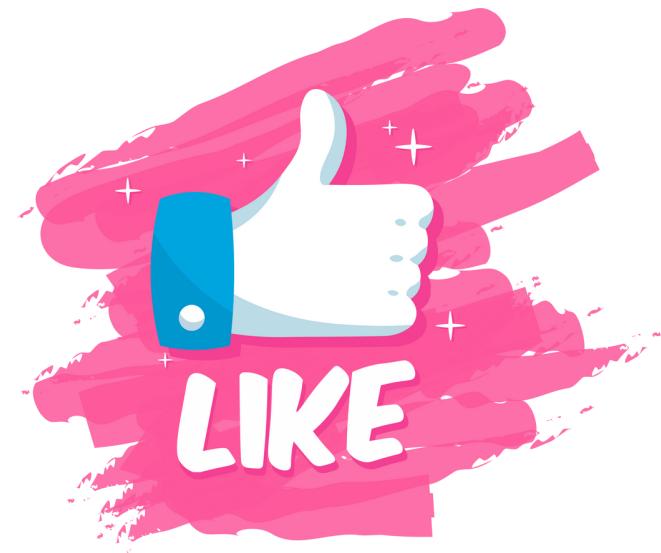
Need of Regression Testing:

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





Error vs Fault vs Failure



Error vs Fault vs Failure

Error:

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

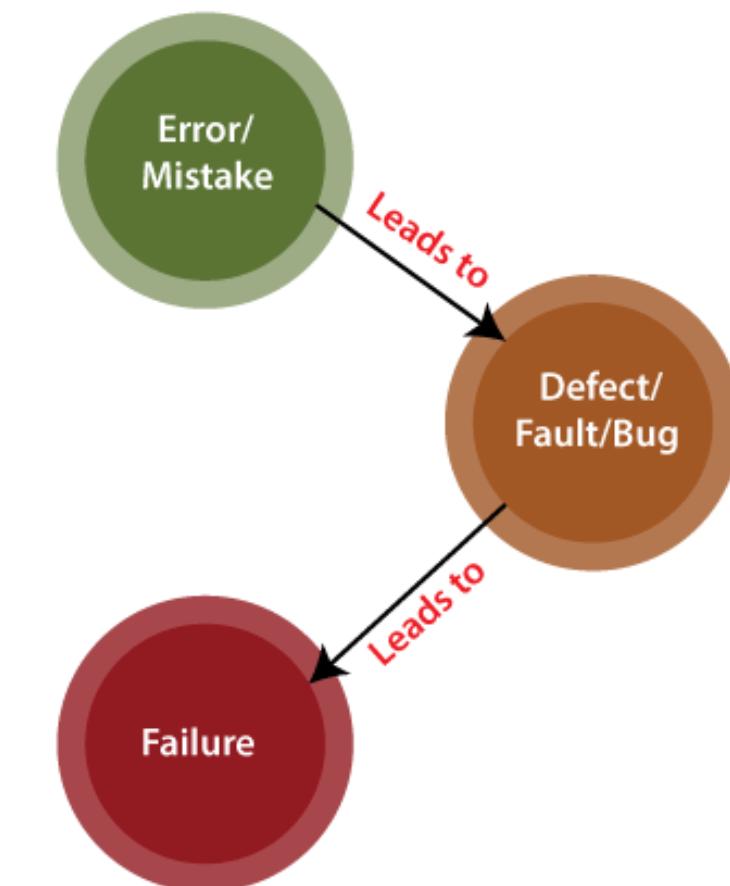


Fault:

- 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 inability of a system or a component to perform its required functions within specified performance requirements.
- Failure occurs when fault executes.

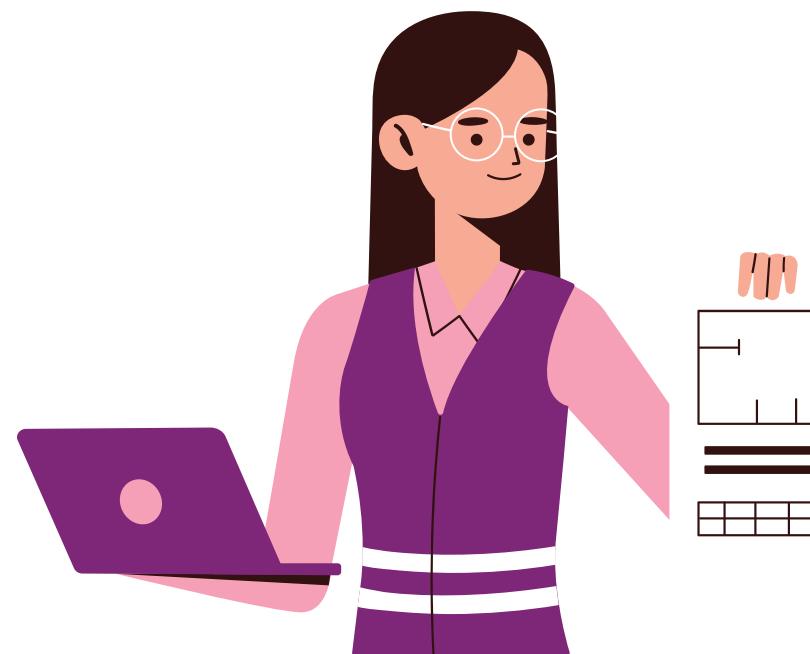




Test Drivers and Test Stubs

Test Drivers and Test Stubs

- The Stubs and Drivers are considered as elements which are equivalent to to-do modules that could be replaced if modules are in their developing stage, missing or not developed yet, 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.

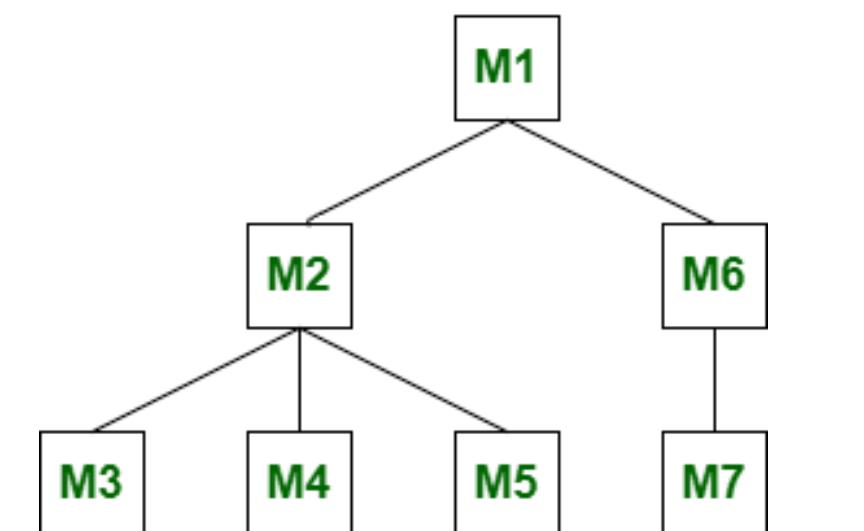


Stubs

- Stubs are developed by software 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 message 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.

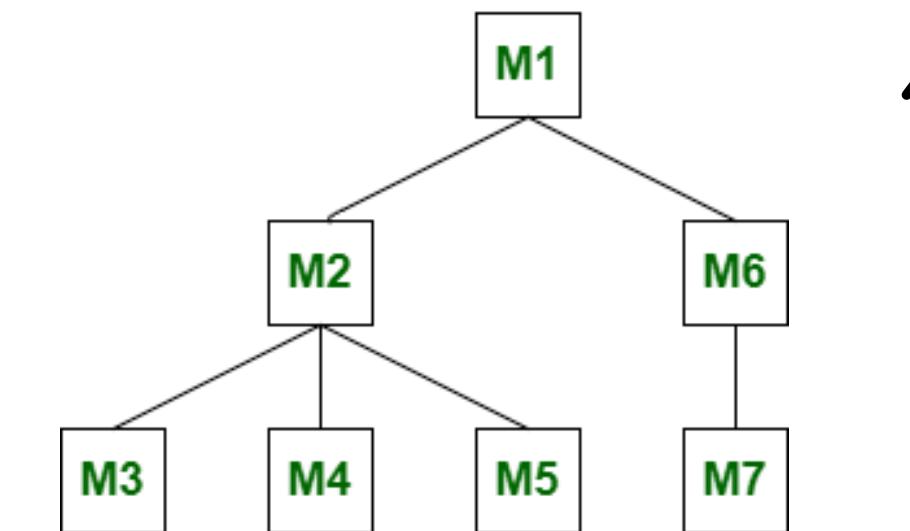


Drivers

- 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.

Example:

- Module-A : Login page website,
- Module-B : Home page of the website
- Module-C : Profile setting
- Module-D : Sign-out page



Engineering in One Video (EIOV)

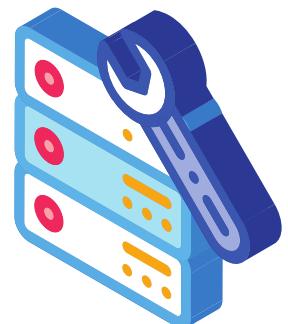
Watch video on  YouTube



Structural Testing (White Box Testing)

Structural Testing (White Box Testing)

- White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output 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 technique is also known as 'white-box' or 'glass-box' testing technique because here the testers require knowledge of how the software is implemented, how it works. In white-box testing the tester is concentrating on how the software does it.
- It is one of two parts of the Box Testing approach to software testing. Its counterpart, Blackbox testing, involves testing from an external or end-user type perspective. On the other hand, White box testing in software engineering is based on the inner workings of an application and revolves around internal testing.
- Code optimization by finding hidden errors.
- White box test cases can be easily automated.



White Box Testing techniques:

Statement coverage:

- Testing every possible statement in the code is executed at least once.
- **Tools:** To test the statement coverage the [Cantata++](#) can be used.

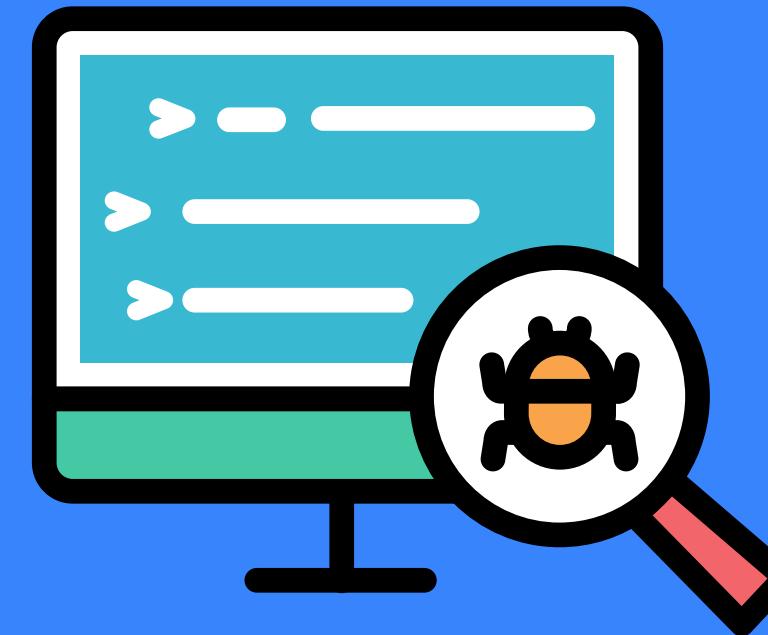
Decision coverage:

- Testing every possible decision conditions like if-else, for loop and other conditionals 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 software application.

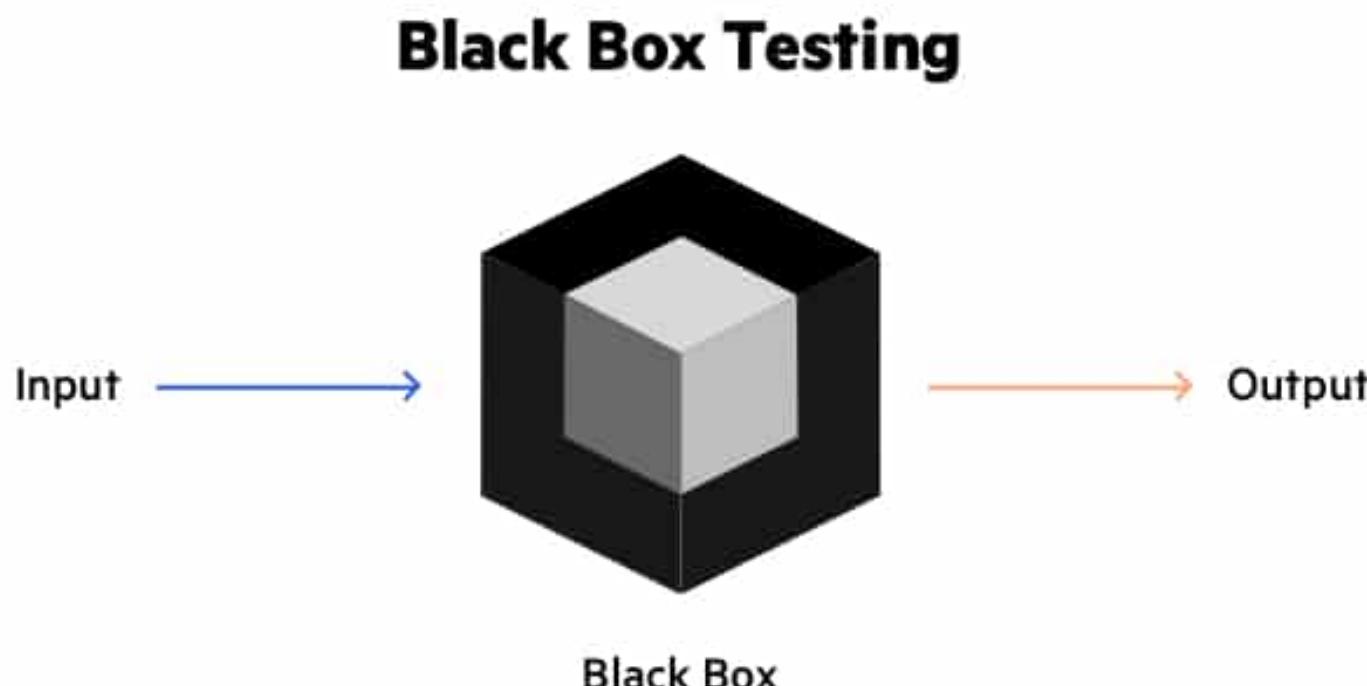




Functional Testing (Black Box Testing)

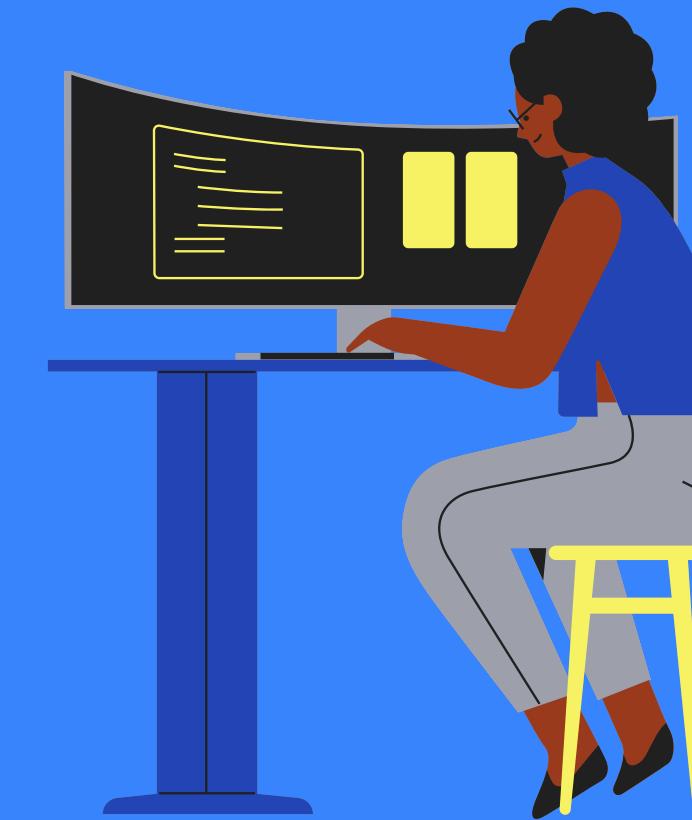
Functional Testing (Black Box Testing)

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



Engineering in One Video (EIOV)

Watch video on  YouTube



Alpha and Beta Testing of Products

Alpha Testing

- Alpha testing is one of the most common software testing strategies used in software development. It is specially used by product development organizations.
- 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 software at an early stage.
- Helps simulate real time user behavior and environment.

Disadvantages:

- In depth functionality of the software 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/software 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 software application in a "real environment" and can be considered as a form of external User Acceptance Testing.
- Beta version of the software 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.

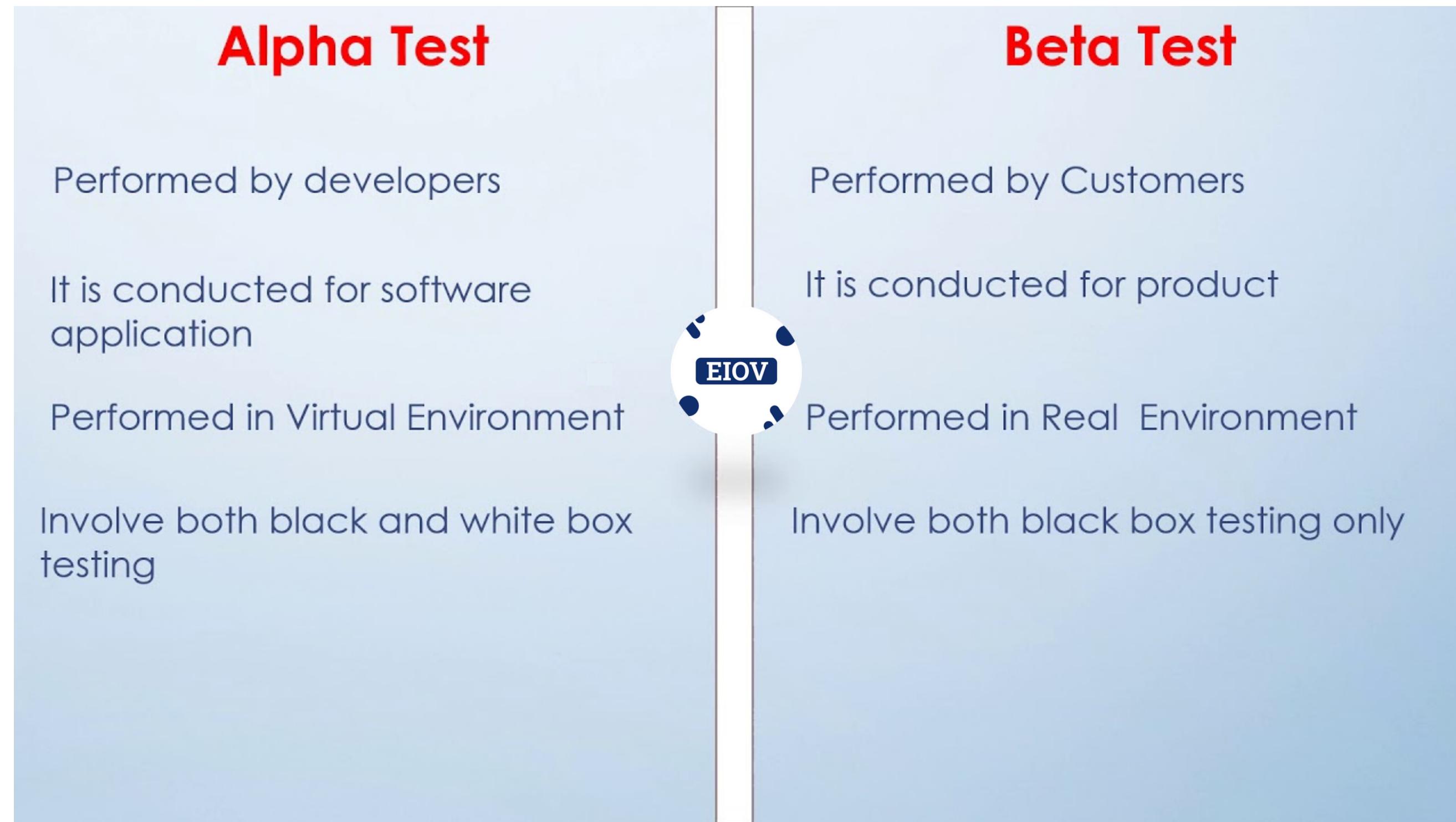
Advantages:

- 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 vs Beta Testing



Engineering in One Video (EIOV)

Watch video on  YouTube



Formal Technical Reviews (Peer Reviews)

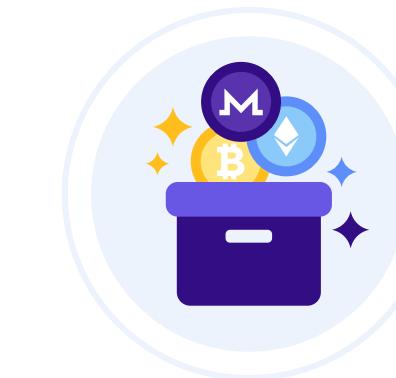
Formal Technical Reviews (Peer Reviews)

- 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 between one and two hours reviewing the product, making notes
- The review meeting is attended by the review leader, all reviewers and the producer. One of the reviewers acts 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.

Formal Technical Reviews (Peer Reviews)

Objectives:

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



Engineering in One Video (EIOV)

Watch video on  YouTube



Walk Through

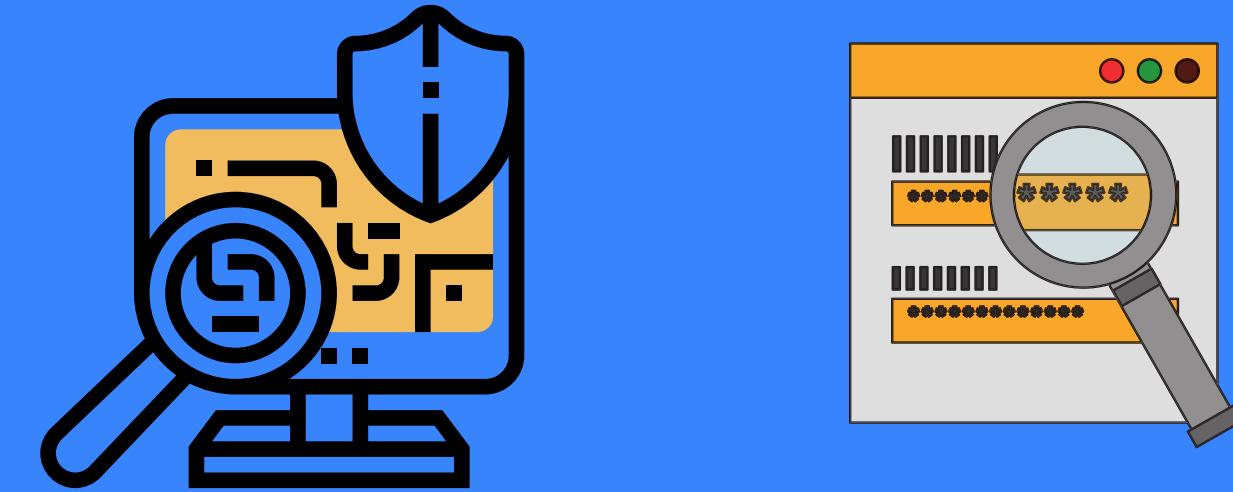


Walk Through

- 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 software in advance of the review. Having at least one senior programmer as a reviewer 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 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

- Code inspection is a process of examining the code of program for identification of certain errors which are not identifiable by the code walkthrough.
- Code inspection is done to find 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.

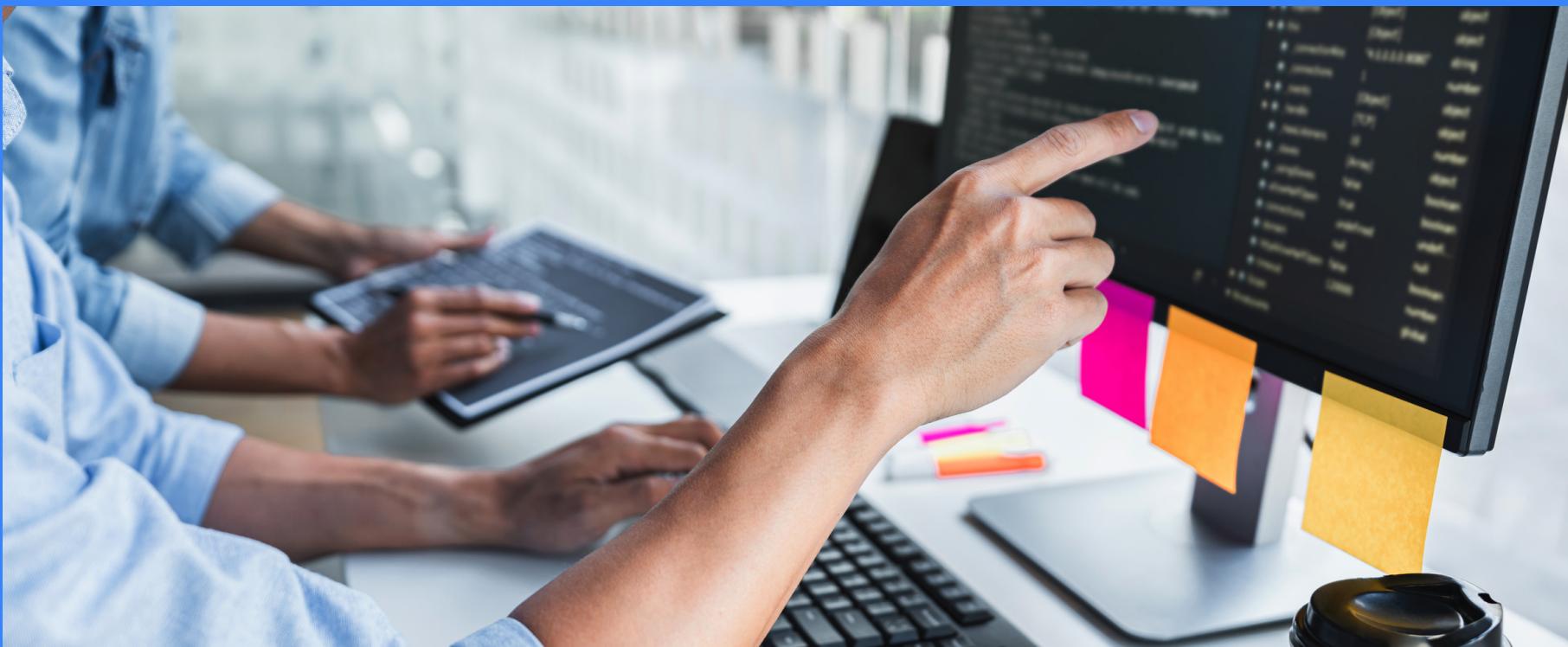
Programming errors can be removed during code inspection:

- Use of un-initialized variables.
- Jumps within loop(for loop, while loop, do while loop etc).
- Non-terminating loops.
- Mismatched assignment.
- Size of array is not initialized.
- Improper allocation and de-allocation.



Engineering in One Video (EIOV)

Watch video on  YouTube



Compliance with Design and Coding Standards

Compliance with Design

- Compliance Testing is performed to maintain and validate the compliant state for the life of the software. 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 objective 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 software development organizations normally require their programmers to adhere to some well-defined and standard style of coding called coding standards.

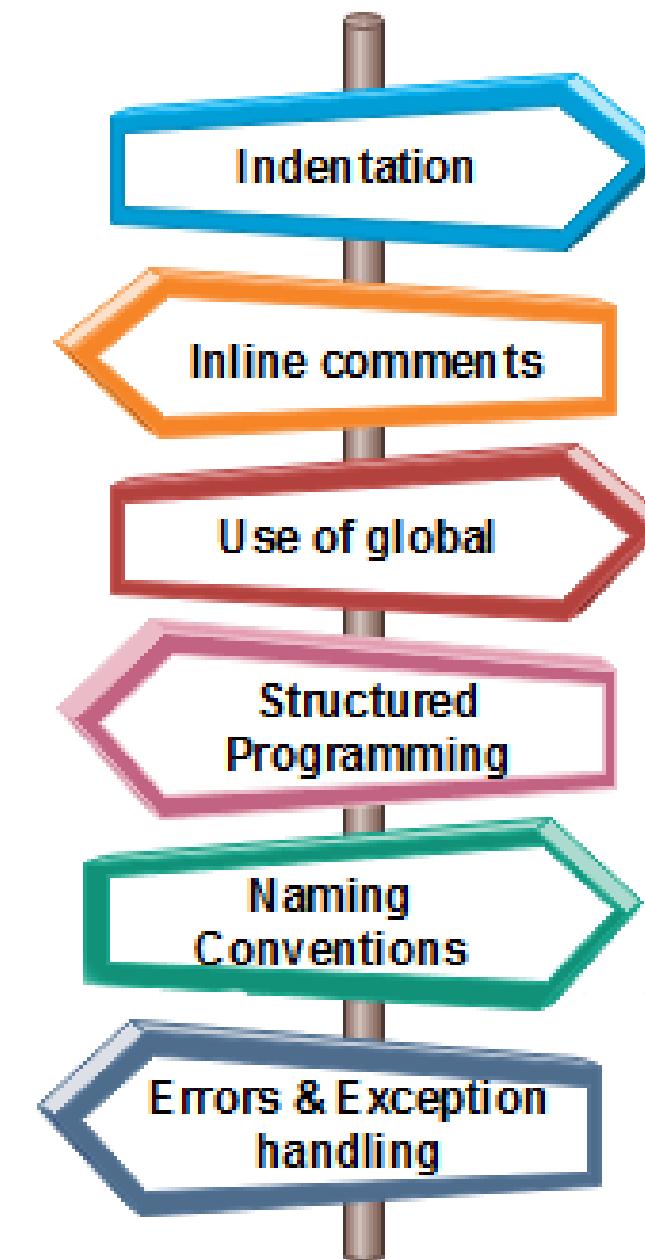
Coding Standards:

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



Coding Standards

Coding Standards



Coding Standards

Coding Guidelines:

1. Line length
2. Spacing
3. Code is well documented
4. Length not exceed 10 source lines
5. Don't use goto statement
6. Inline comments
7. Error messages



Happy Ending!



Congratulations!

