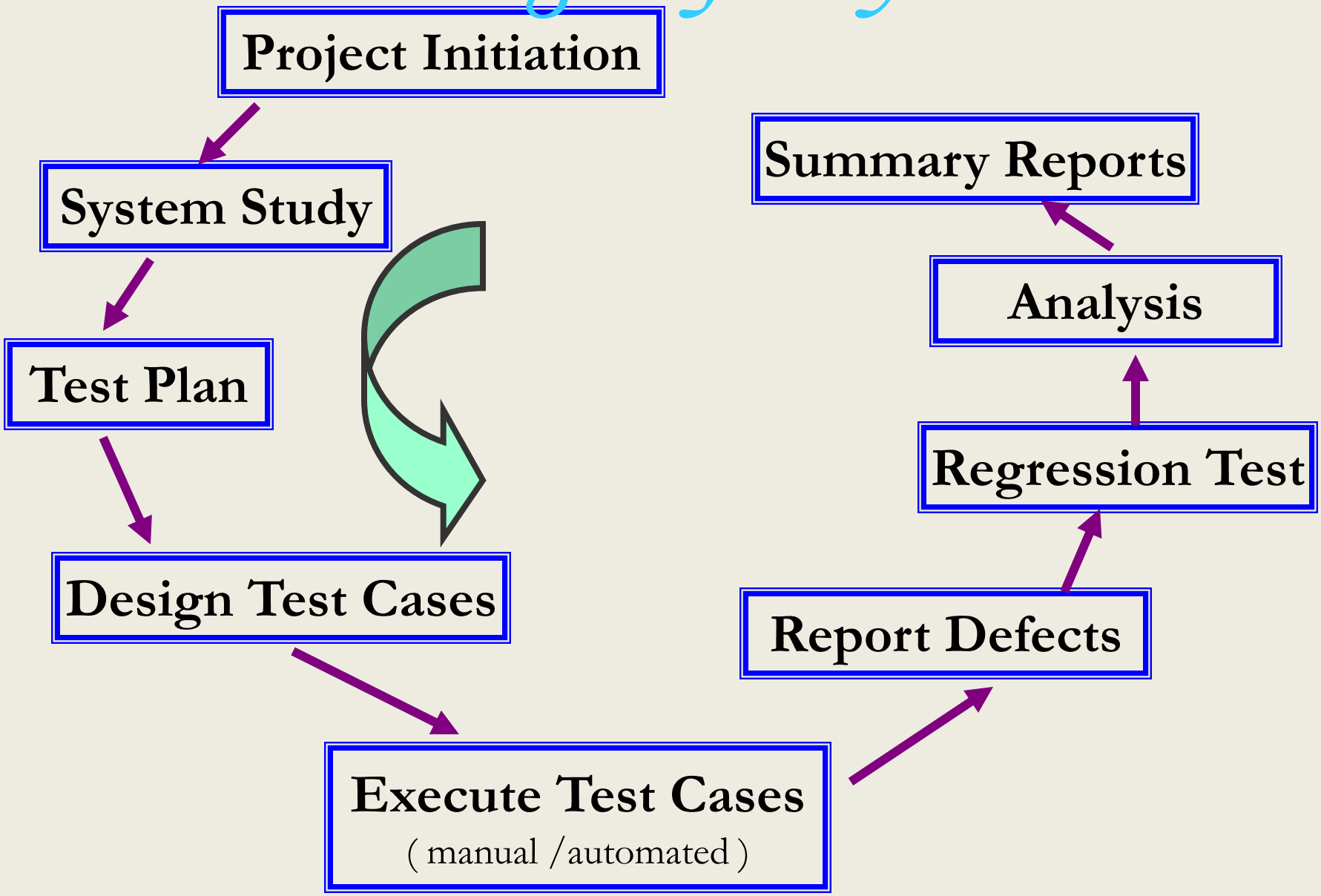
Section-C

Lecture-22

Dronacharya College of Engineering

Testing Life Cycle



Test Planning

Test Plan

A test plan is a systematic approach to testing a system i.e. software. The plan typically contains a detailed understanding of what the eventual testing workflow will be.

Test Case

A test case is a specific procedure of testing a particular requirement.

OR

A test case has components that describe an input, action or event and an expected response, to determine if a feature of an application is working correctly

Why we write Test Cases ?

The basic objective of writing test cases is to validate the testing coverage of the application. If you are working in any CMM company then you will strictly follow test cases standards.

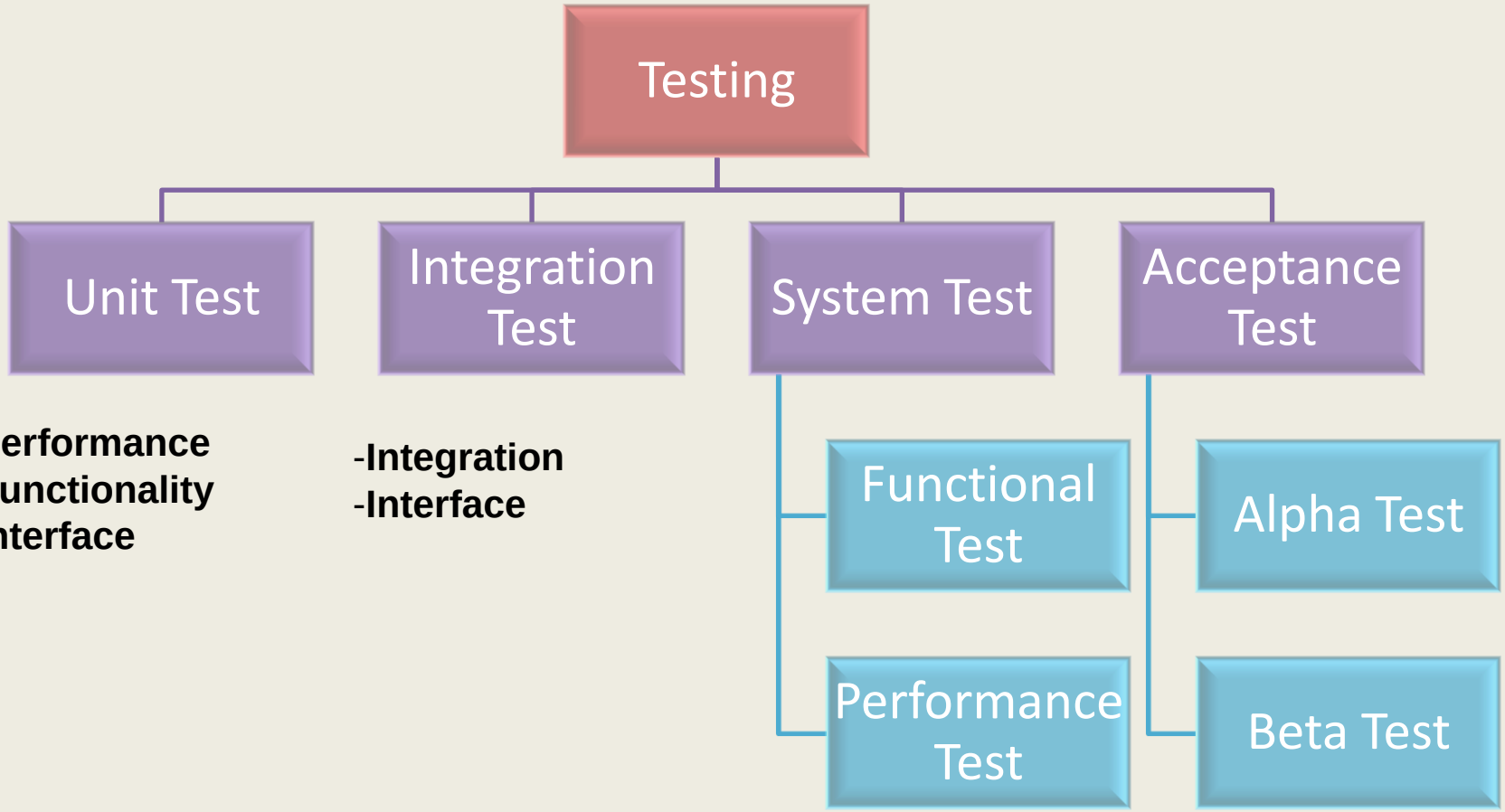
How to write a test cases?

There is a format to write a test case.

- Field in test cases:
- Test case id:
- Unit to test: ***what to be verified***
- Assumptions
- Test Data: ***variables and their values***
- Steps to be executed:
- Expected results:
- Actual result:
- Pass/Fail:
- Comments:

Levels Of Testing

Levels Of Testing



Unit Test

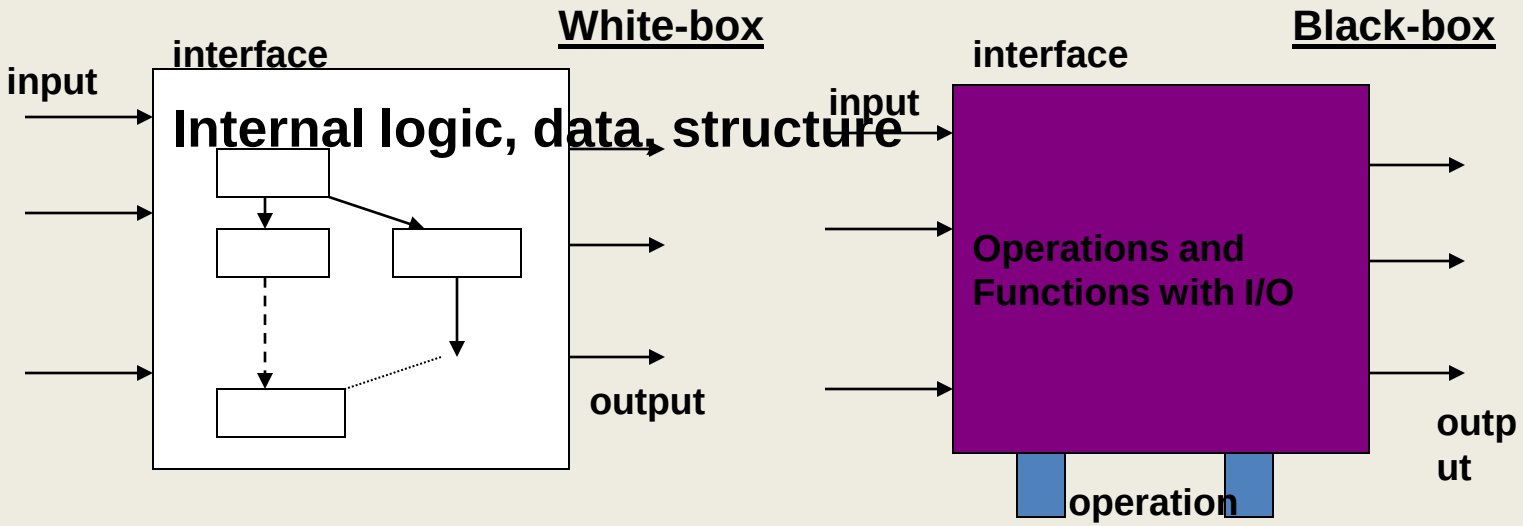
1. The firs level in the testing process is called unit testing.
2. Unit testing concerns testing smallest component of the software
3. Unit testing is done by Developer.

Unit Test (Component Level Test)

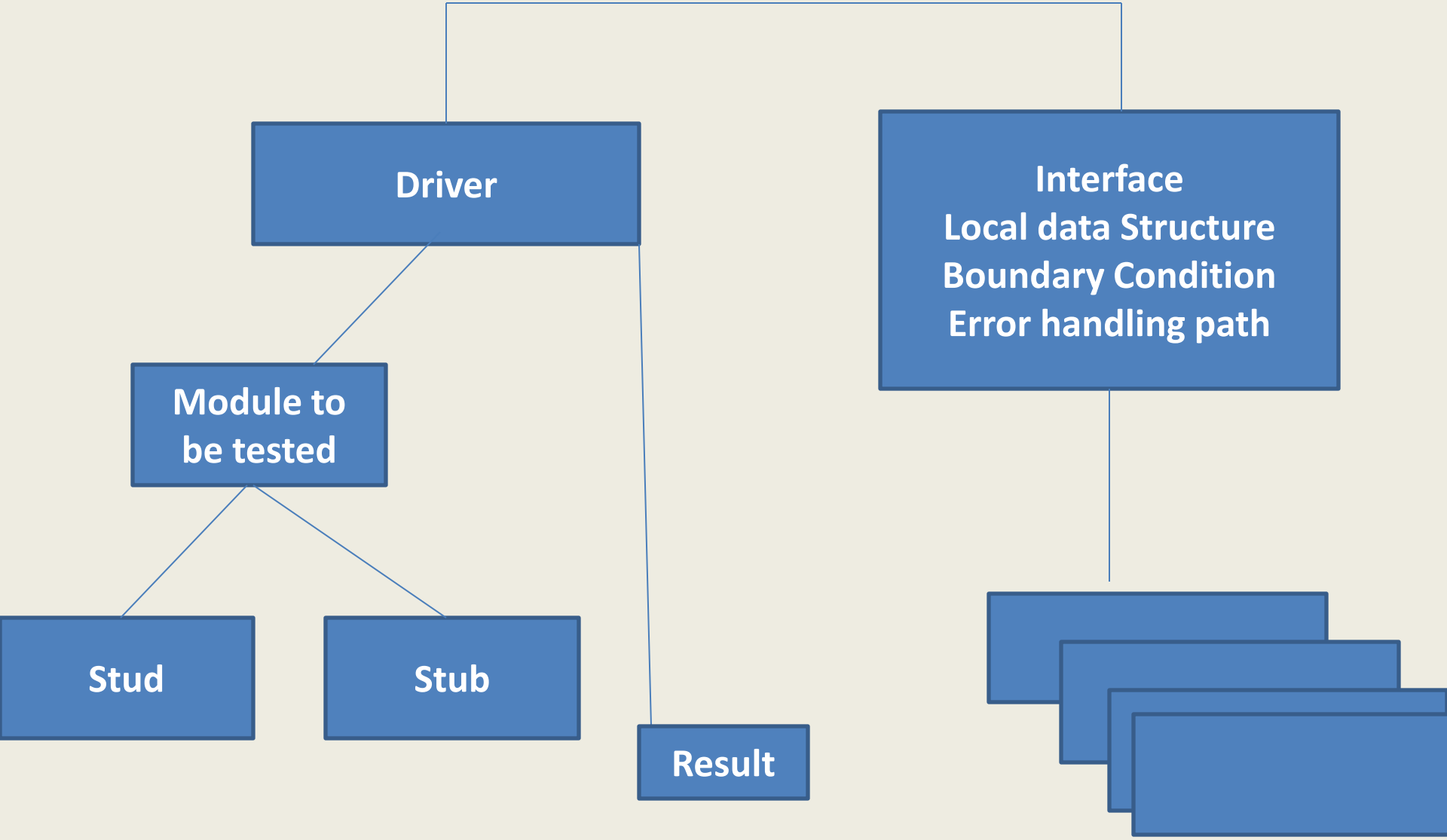
Unit testing: Individual components are tested independently to ensure their quality. The focus is to uncover errors in design and implementation, including

- **data structure in a component**
- **program logic and program structure in a component**
- **component interface**
- **functions and operations of a component**

Unit testers: developers of the components.



Unit Test Procedures



Unit Test Procedures

1. Each test case should be linked with a set of anticipated results.
2. As a module is not a stand alone program, driver and stub software must be produced for each test units.
3. In most of applications a *driver* is nothing than a "main program" that accept test case data, passes such data to the component(to be tested), and print relevant results.
4. *Stubs* serve to replace modules that are subordinate to the component to be tested

Integration Testing

After completing the unit testing and dependent modules development, programmers connect the modules with respect to HLD for Integration Testing through below approaches.

A study has shown that almost 40% of the error are due to integration and interface problems. there are number of strategies that can be followed to do integration testing.

- Incremental Strategy
- Non- Incremental Strategy
- Mixed Strategy

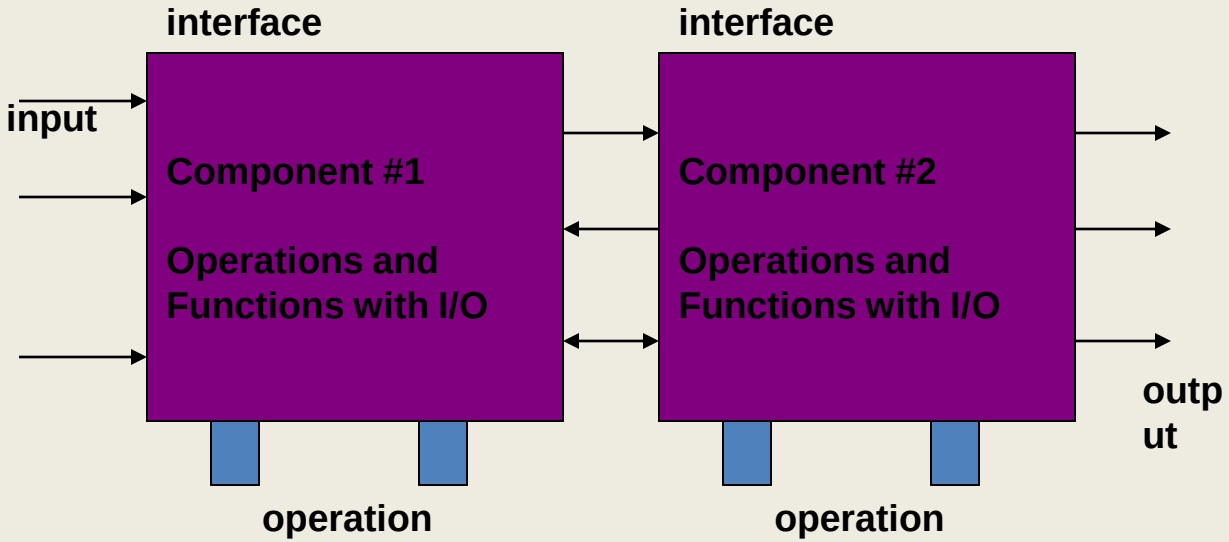
Integration Testing

Integration test: A group of dependent components are tested together to ensure their the quality of their integration unit.

The focus is to uncover errors in:

- Design and construction of software architecture
- Integrated functions or operations at sub-system level
- Interfaces and interactions between them
- Resource integration and/or environment integration

Integration testers: either developers and/or test engineers.



Approaches to Integration Testing

The various approaches used for integration testing are:

- Big Bang Approach
- Incremental Approach
- Top Down Integration Testing
- Bottom-up Integration Testing
- Sandwich Integration Testing
- Regression Testing

System Testing

After completing **Unit** and **Integration** testing through white box testing techniques development team release an .exe build (all integrated module) to perform black box testing.

- Usability Testing
- Functional Testing
- Performance Testing
- Security Testing

System Testing

System test: The system software is tested as a whole. It verifies all elements mesh properly to make sure that all system functions and performance are achieved in the target environment.

The focus areas are:

- System functions and performance
- System reliability and recoverability (recovery test)
- System installation (installation test)
- System behavior in the special conditions (stress and load test)
- System user operations (acceptance test/alpha test)
- Hardware and software integration and collaboration
- Integration of external software and the system

System testers: test engineers in ITG or SQA people.

When a system is to be marketed as a software product, a testing process called beta testing is often used.

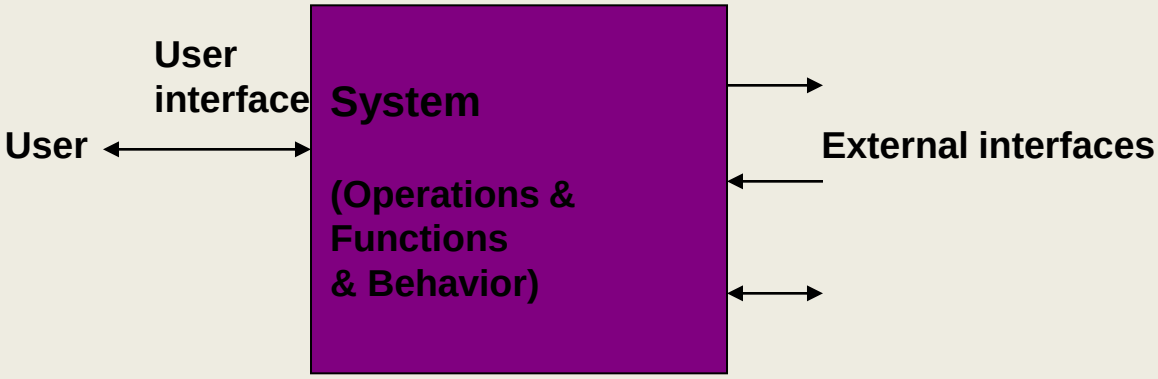
Function Validation Testing

Validation test: The integrated software is tested based on requirements to ensure that we have a right product.

The focus is to uncover errors in:

- System input/output
- System functions and information data
- System interfaces with external parts
- User interfaces
- System behavior and performance

Validation testers: test engineers in ITG or SQA people.



Usability Testing

During this test, testing team concentrates on the user friendliness of build interface. It consists of following sub tests.

- **User Interface Test:** Ease of use (screens should be understandable to operate by End User)
- **Look & Feel :-** attractive
- **Speed in interface :-** Less number of task to complete task
- **Manual Support Test :-** Context sensitiveness of user manual.

Functional Testing

- The process of checking the behavior of the application.
- It is geared to functional requirements of an application.
- To check the correctness of outputs.
- Data validation and Integration i.e. inputs are correct or not.

Performance Testing

- **LOAD TESTING –** Also Known as Scalability Testing. During this test, test engineers execute application build under customer expected configuration and load to estimate performance.

- **STRESS TESTING –** During this test, Test engineers estimates the peak load. To find out the maximum number of users for execution of out application user customer expected configuration to estimate peak load.

  PEAK LOAD > CUSTOMER LAOD (EXPECTED)

- **DATA VOLUME TESING --** Testing team conducts this test to find the maximum limit of data volume of your application.

Security Testing

Testing how well the system protects against unauthorized internal or external access, willful damage, etc, may require sophisticated testing techniques

Smoke testing

Smoke testing is non-exhaustive software testing, ascertaining that the most crucial functions of a program work, but not bothering with finer details.

Alpha Testing

1. The application is tested by the users who doesn't know about the application.
2. Done at developer's site under controlled conditions
3. Under the supervision of the developers.

Acceptance Testing

A formal test conducted to determine whether or not a system satisfies its acceptance criteria and to enable the customer to determine whether or not to accept the system.

It is the final test action before deploying the software. The goal of acceptance testing is to verify that the software is ready and can be used by the end user to perform the functions for which the software was built.

Beta Testing

1. This Testing is done before the final release of the software to end-users.

1. Before the final release of the software is released to users for testing where there will be no controlled conditions and the user here is free enough to do what ever he wants to do on the system to find errors.

Regression Testing

Testing with the intent of determining if bug fixes have been successful and have not created any new problems. Also, this type of testing is done to ensure that no degradation of baseline functionality has occurred.

Monkey Testing

Testing the application randomly like hitting keys irregularly and try to breakdown the system there is no specific test cases and scenarios for monkey testing.