Software Testing Fundamentals

Agenda

- Introduction
- Importance of testing in SDLC
- Testing life cycle
- Test planning
- Types of testing
- Verification & Validation
- Quality Assurance & Control
- Bug reporting

Introduction to Software Testing

Software Testing

Software testing is a process used to identify the correctness, completeness and quality of developed computer software.

It is theprocess of executing a program / application under positive and negative conditions by manual or automated means. It checks for the :-

- Specification
- Functionality
- Performance

Why Software Testing ?

Software Testing is important as it may cause mission failure, impact on operational performance and reliability if not done properly.

Effective software testingdelivers quality software products satisfying user's requirements, needs and expectations.

Who Should Test?

- Developer
  - Understands the system
  - But, will test gently
  - And, is driven by deadlines

- Independent tester
  - Must learn system
  - But, will attempt to break it
  - And, is driven by "quality"

What ….????

…is an ”ERROR”??

….is a ”Bug”??

….is Fault,Failure ??

Bug, Fault & Failure

A person makes an **Error**
That creates a **fault** in software
That can cause a **failure** in operation

Error : **An error is a human action that produces the incorrect result that results in a fault.**

Bug : **The presence of error at the time of execution of the software.**

Fault : **State of software caused by an error.**

Failure : **Deviation of the software from its expected result. It is an event.**

Who is a Software Tester??..

Software Tester is the one who performs testing and find bugs, if they exist in the tested application.

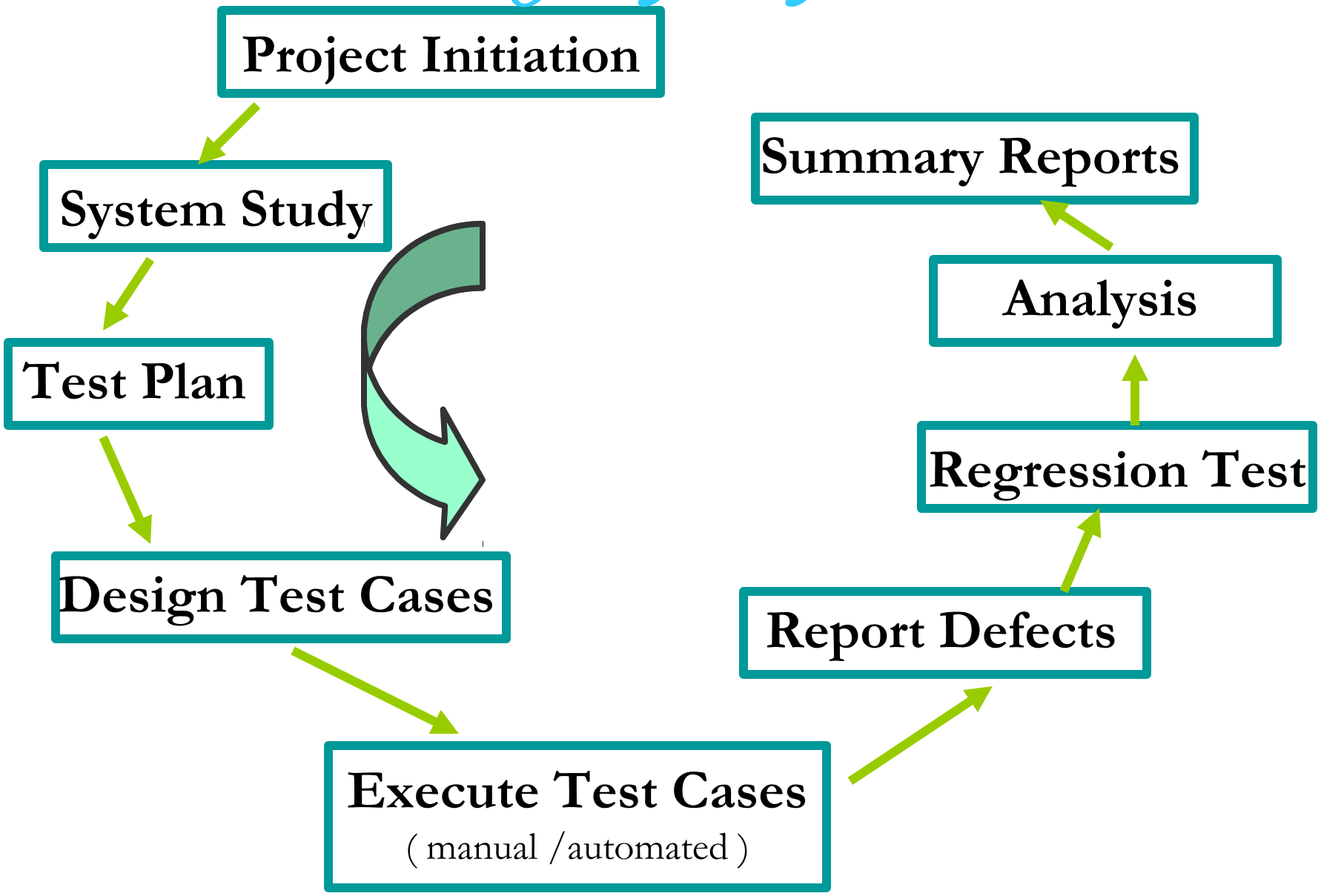
Importance Of Testing In SDLC

When to Start Testing in SDLC

- **Requirement**
- **Analysis**
- **Design**
- **Coding**
- **Testing**
- **Implementation**
- **Maintenance**

ϖ ***Testing starts from Requirement Phase***

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.

It will include:

- Identification of specific requirement tested
- Test case success/failure
- criteria Specific steps to
- execute test Test Data

Levels Of Testing

Unit Testing

1. Test each module individually.
2. Follows a white box testing (Logic of the program)
3. Done by Developers

Unit Testing (White Box)

Individual components are tested.
It is a path test.
To focus on a relatively small segment of code and aim to exercise a high percentage of the internal path
Disadvantage: the tester may be biased by previous experience. And the test value may not cover all possible values.

Integration Testing

After completing the unit testing and dependent modules development, programmers connect the modules for Integration Testing

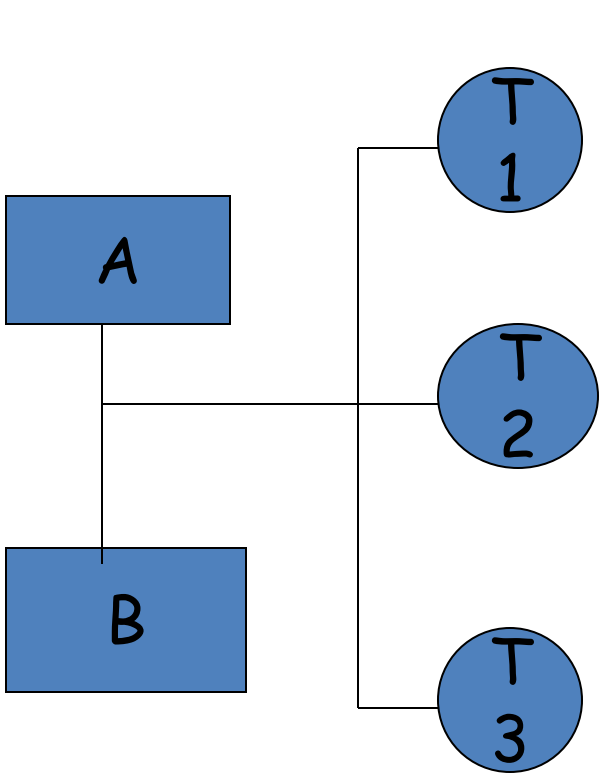
Integration Testing

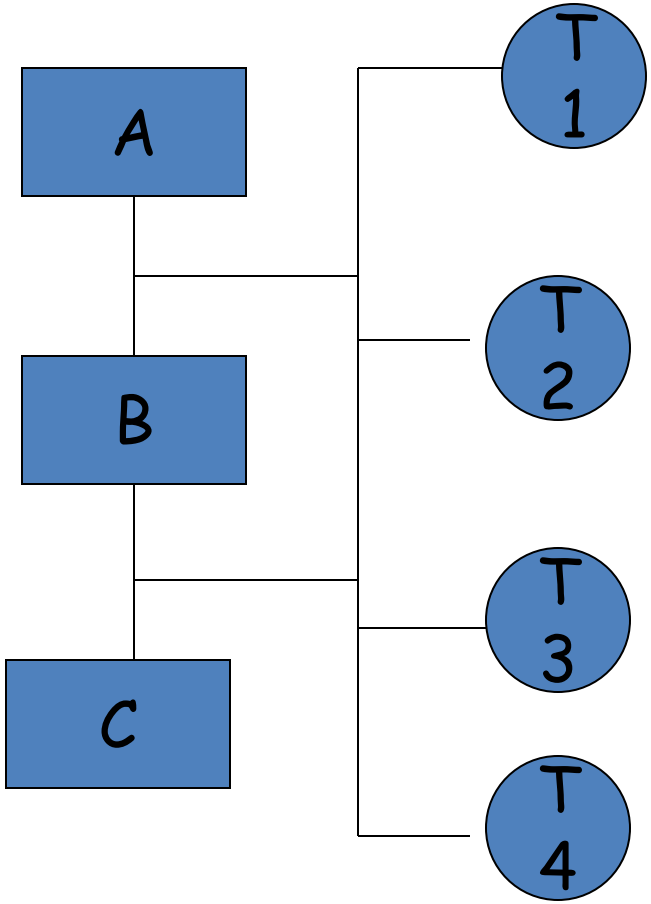
Top-down Integration Test
Bottom-up Integration Test

Top-down Integration Test

The control program is tested first.
Modules are integrated one at a time.
Emphasize on interface testing
Advantages: No test drivers needed
Interface errors are discovered early
Modular features aid debugging
Disadvantages: Test stubs are needed
Errors in critical modules at low levels are found late.

Top-down Testing





Bottom-up Integration Test

Allow early testing aimed at proving feasibility

Emphasize on module functionality and performance

Advantages: No test stubs are needed

Errors in critical modules are found early

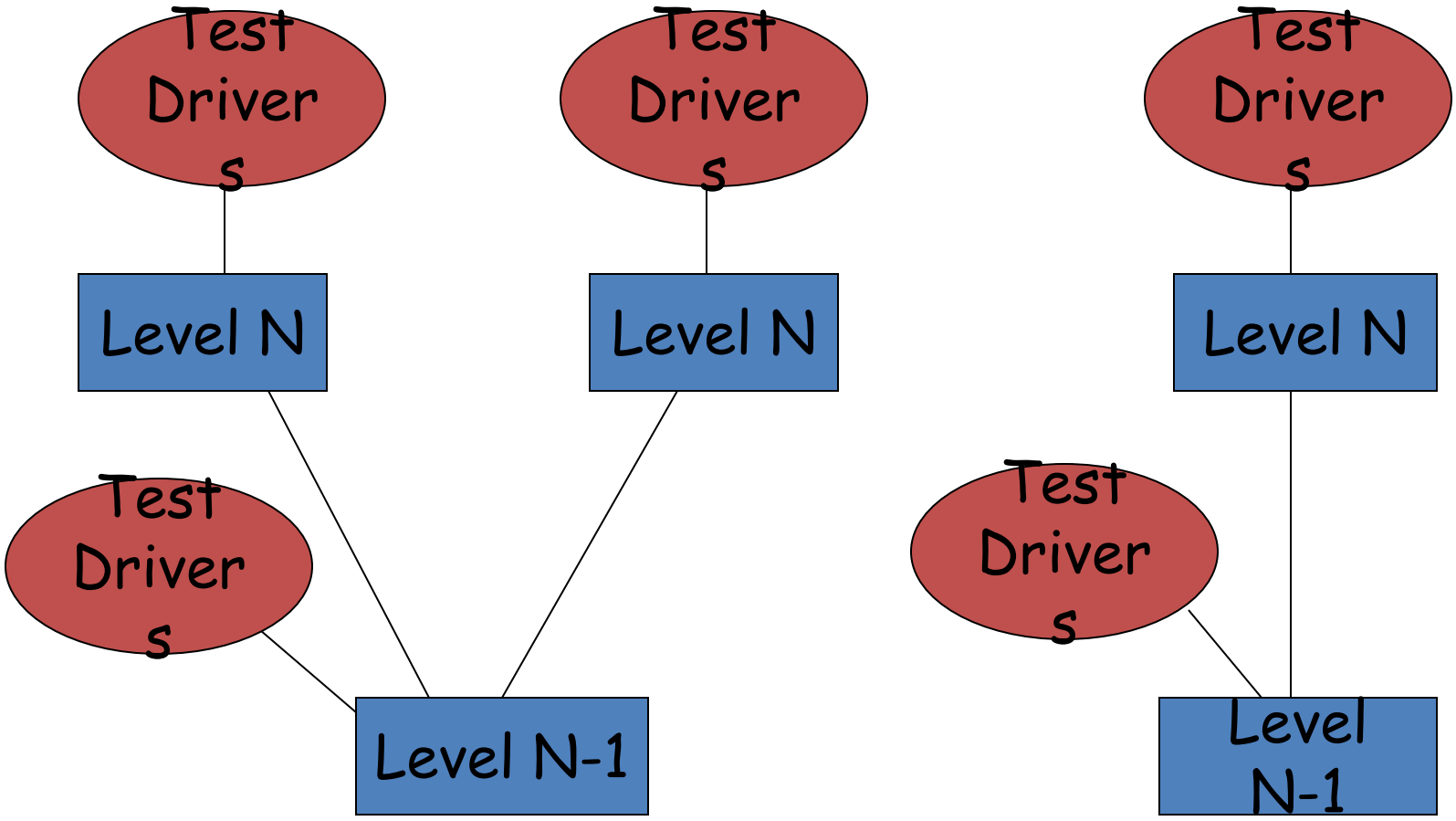
Disadvantages: Test drivers are needed

Interface errors are discovered late

Stubs and drivers

Stubs and drivers both are dummy modules and are only created for software test purposes. Stubs are used in top down testing approach, when one has the major module ready to test, but the sub modules are still not ready yet. Drivers are used in bottom up testing approach.

Bottom-up 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

Non-Functional Testing

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.

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 LOAD (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

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.

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.

Beta Testing

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

2. 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.

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.

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.

Verification & Validation

Verification

- **Verification is the process confirming that -software meets its specification, done through inspections and walkthroughs**

***Use* – To identify defects in the product early in the life cycle**

Validation

- **Validation is the process confirming that it meets the user's requirements. It is the actual testing.**

Verification : Is the Product Right

Validation: Is it the Right Product

Quality Assurance
&
Quality Control

Why Quality ?

- **Quality is the important factor affecting an organization's long term performance.**
- **Quality improves productivity and competitiveness in any organization.**

What is Quality ?

Quality is defined as meeting the customer's requirements and according to the standards

The best measure of Quality is given by FURPS

- **Functionality**
- **Usability**
- **Reliability**
- **Performance**
- **Scalability**

Quality Assurance

Quality Assurance is a planned and systematic set of activities necessary to provide adequate confidence that products and services will conform to specified requirements and meets user needs.

- It is process oriented.
- Defect prevention based.
- Throughout the Life Cycle.
- It's a management process.

Quality Control

Quality control is the process by which product quality is compared with the applicable standards and the action taken when non conformance is detected.

- It is product oriented
- Defect detection based

QA vs. QC

- Quality Assurance makes sure that we are doing the right things, the right Way.
- QA focuses on building in quality and hence preventing defects.
- QA deals with process.
- QA is for entire life cycle.
- QA is preventive process.

- Quality Control makes sure the results of what we've done are what we expected .
- QC focuses on testing for quality and hence detecting defects.
- QC deals with product.
- QC is for testing part in SDLC.
- QC is corrective process.

Bug Life Cycl