Software Testing

Presented by
Saptarshi Dutta Chowdhury

MCA, Heritage Institute of Technology

INTRODUCTION

- It is the process used to identify the correctness, completeness and quality of developed computer software.

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

  - Specification
  - Functionality
  - Performance

OBJECTIVES

- Uncover as many as errors (or bugs) as possible in a given product.

- Demonstrate a given software product matching its requirement specifications.

- Validate the quality of a software testing using the minimum cost and efforts.

- Generate high quality test cases, perform effective tests, and issue correct and helpful problem reports.

Error, Bug, Fault & Failure

Error : It is a human action that produces the incorrect result that produces a fault.

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

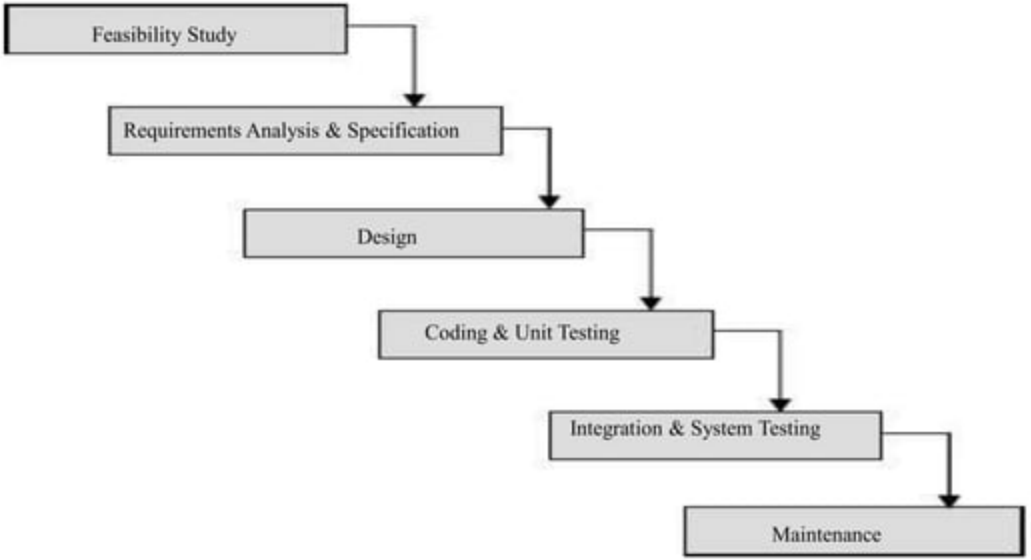
Fault : State of software caused by an error.

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

SDLC(Software Development Life Cycle)

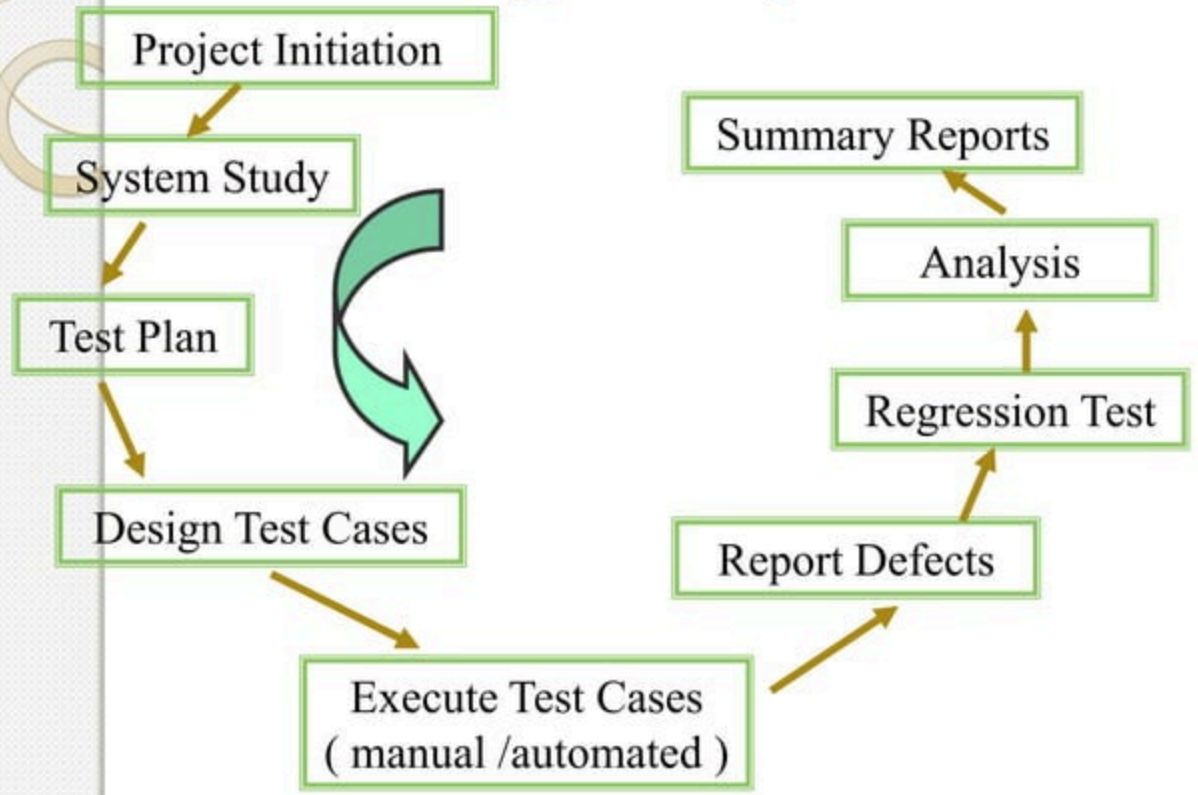
- Standard model used word wide to develop a software.
- A framework that describes the activities performed at each stage of a software development project.
- Necessary to ensure the quality of the software.
- Logical steps taken to develop a software product.

Classical Waterfall Model



It is the oldest and most widely used model in the field of software development.

Testing Life Cycle



Test Plan

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

Test Case

It 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

Verification vs Validation

- **Verification**: The software should confirm to its specification (Are we building the product right?)
- **Validation**: The software should do what the user really requires (Are we building the right product?)

Testing Methodologies

Black box testing

White box testing

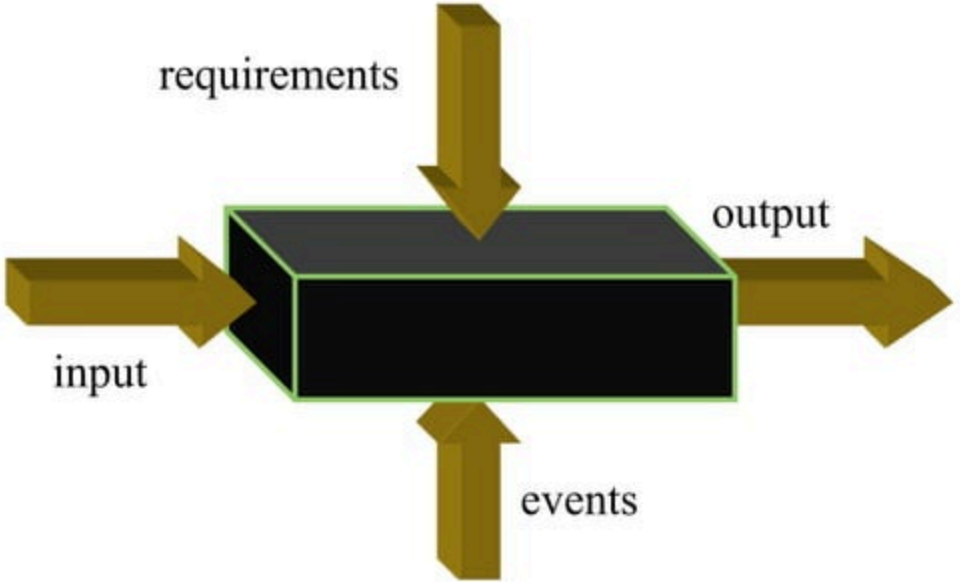
Black box testing

- No knowledge of internal program design or code required.
- Tests are based on requirements and functionality.

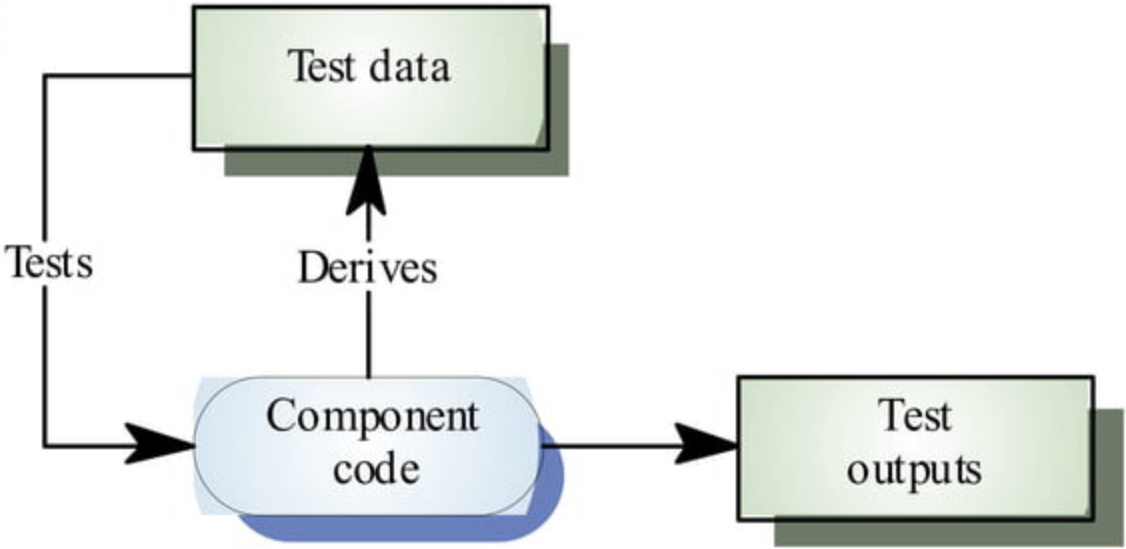
White box testing

- Knowledge of the internal program design and code required.
- Tests are based on coverage of code statements, branches, paths, conditions.

Black box testing



White box testing



Testing Levels

- Unit testing
- Integration testing
- System testing

UNIT TESTING

- Tests each module individually.
- Follows a white box testing (Logic of the program).
- Done by developers.

INTEGRATION TESTING

- Once all the modules have been unit tested, integration testing is performed.
- It is systematic testing.
- Produce tests to identify errors associated with interfacing.

Types:

- Big Bang Integration testing
- Top Down Integration testing
- Bottom Up Integration testing
- Mixed Integration testing

SYSTEM TESTING

- The system as a whole is tested to uncover requirement errors.

- Verifies that all system elements work properly and that overall system function and performance has been achieved.

Types:

- Alpha Testing
- Beta Testing
- Acceptance Testing
- Performance Testing

Alpha Testing

It is carried out by the test team within the developing organization .

Beta Testing

It is performed by a selected group of friendly customers.

Acceptance Testing

It is performed by the customer to determine whether to accept or reject the delivery of the system.

Performance Testing

It is carried out to check whether the system meets the nonfunctional requirements identified in the SRS document.

Types of Performance Testing:

- Stress Testing
- Volume Testing
- Configuration Testing
- Compatibility Testing
- Regression Testing
- Recovery Testing
- Maintenance Testing
- Documentation Testing
- Usability Testing

DISCUSSION

- In order to be cost effective, the testing must be concentrated on areas where it will be most effective.

- The testing should be planned such that when testing is stopped for whatever reason, the most effective testing in the time allotted has already been done.

- The absence of an organizational testing policy may result in too much effort and money will be spent on testing, attempting to achieve a level of quality that is impossible or unnecessary.

THANK YOU