Chapter 2
Software Testing Terminology and Methodology

OXFORD
HIGHER EDUCATION

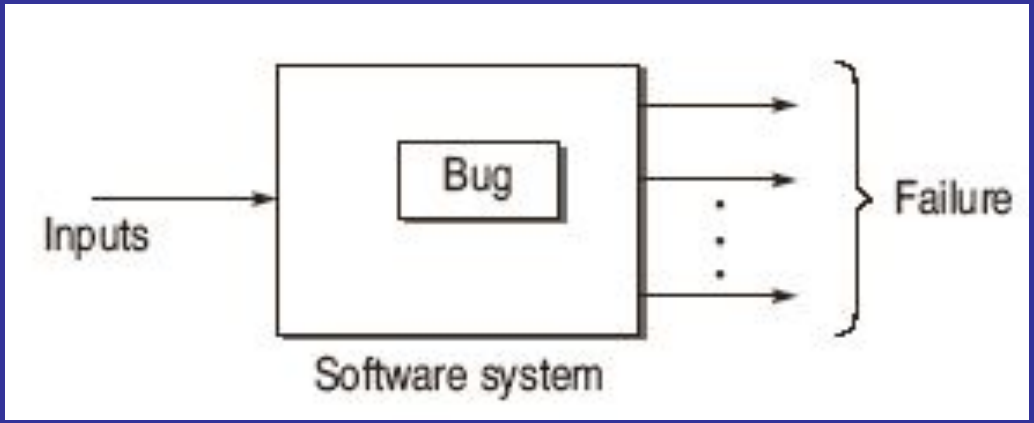
Objectives

- Difference between error, fault and failure.
- Life Cycle of a bug.
- How does a bug affect economics of software testing?
- How does a bug classified?
- Testing Principles
- Software Testing Life Cycle (STLC) and its models.
- Difference between verification and validation.
- Development of software testing methodology

Software Testing Terminology

- **Failure**
  The inability of a system or component to perform a required function according to its specification.

- **Fault / Defect / Bug**
  Fault is a condition that in actual causes a system to produce failure.

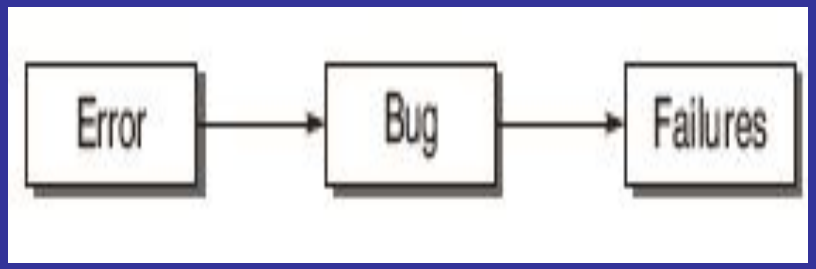


Software Testing Terminology

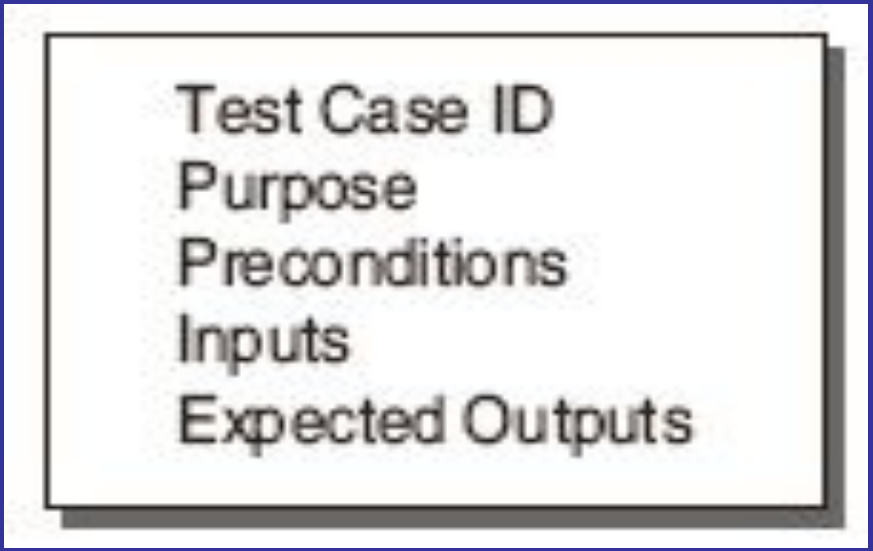
OXFORD
HIGHER EDUCATION

Error

Whenever a member of development team makes any mistake in any phase of SDLC, errors are produced. It might be a typographical error, a misleading of a specification, a misunderstanding of what a subroutine does and so on. Thus, error is a very general term used for human mistakes.



Software Testing Terminology



Software Testing Terminology

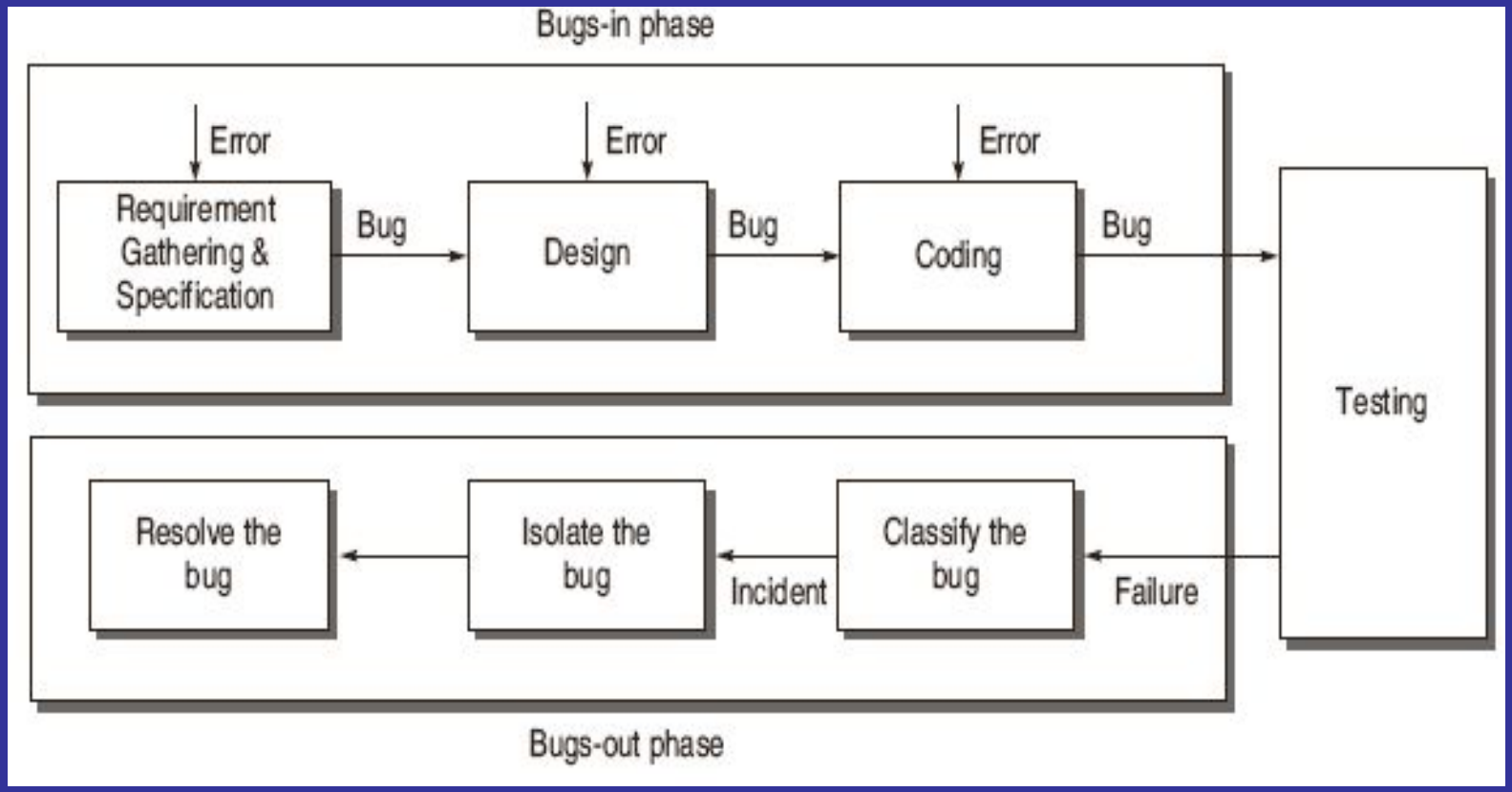
OXFORD
HIGHER EDUCATION

- **Testware**
  The documents created during the testing activities are known as Testware.

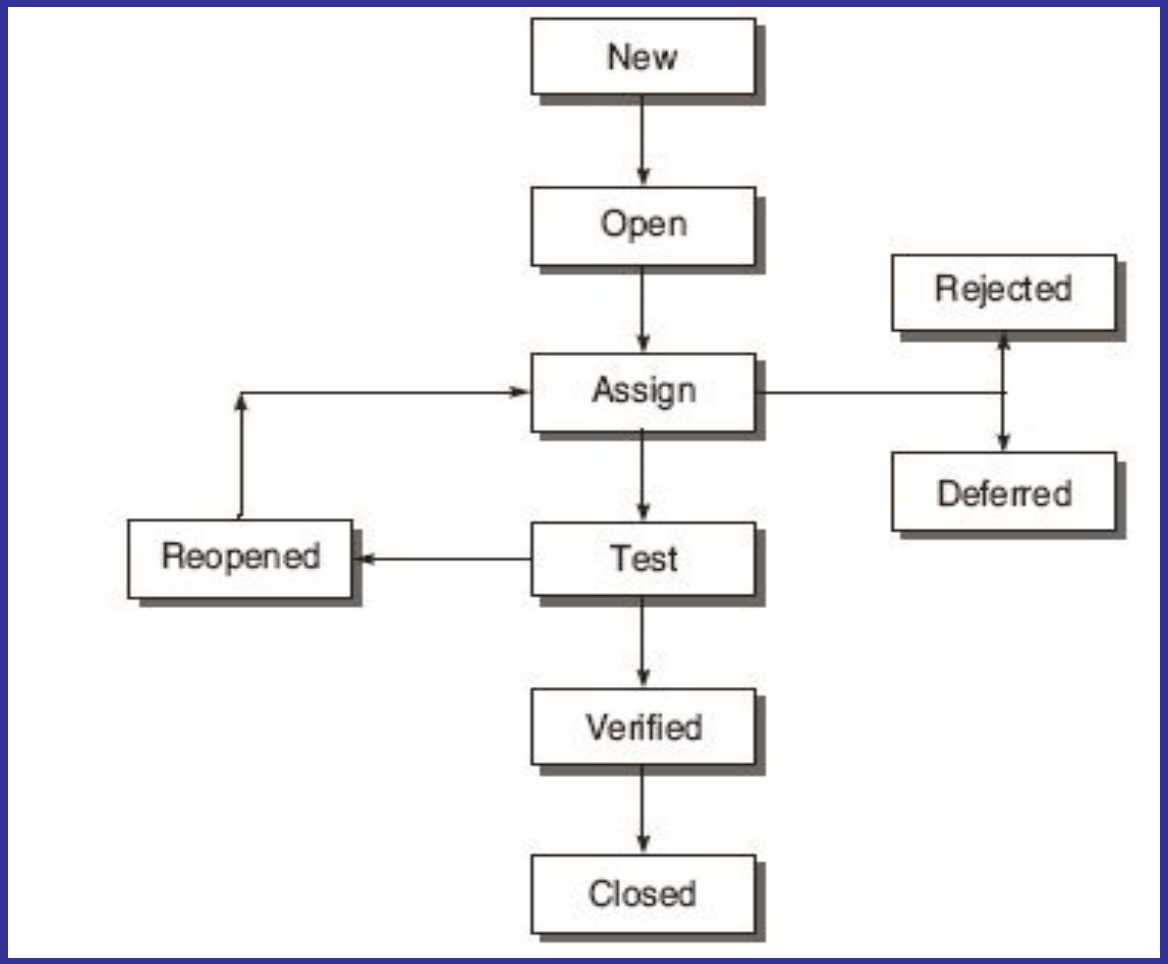
- **Incident**
  the symptom(s) associated with a failure that alerts the user to the occurrence of a failure.

- **Test Oracle**
  to judge the success or failure of a test,

Life Cycle of a Bug



States of a Bug



Bug Classification based on Criticality

OXFORD
HIGHER EDUCATION

- **Critical Bugs**

  the worst effect on the functioning of software such that it stops or hangs the normal functioning of the software.

- **Major Bug**

  This type of bug does not stop the functioning of the software but it causes a functionality to fail to meet its requirements as expected.

- **Medium Bugs**

  Medium bugs are less critical in nature as compared to critical and major bugs.

  Minor Bugs

Bug Classification based on SDLC

OXFORD
HIGHER EDUCATION

Requirements and Specifications Bugs

Design Bugs

- *Control Flow Bugs:Some control paths may be missing*
- *Logic Bugs: logical mistakes*
- *Processing Bugs:computation mistakes*
- *Data Flow Bugs:data errors e.g:uninitialized data*
- *Error Handling Bugs:exception handling mechanisim*
- *Race Condition Bugs:event A and B*
- *Boundary Related Bugs:integer values between 1 and 100*
- *User Interface Bugs"wrong content in the help context*

Coding Bugs:Undeclared data, undeclared routines

Interface and Integration Bugs:invalid inputs and outputs and incosistencies and incompatibility between modules

System Bugs: maximum no. of users maximum memory limit

Testing Bugs: failure to notice/report a problem

Testing Principles

OXFORD
HIGHER EDUCATION

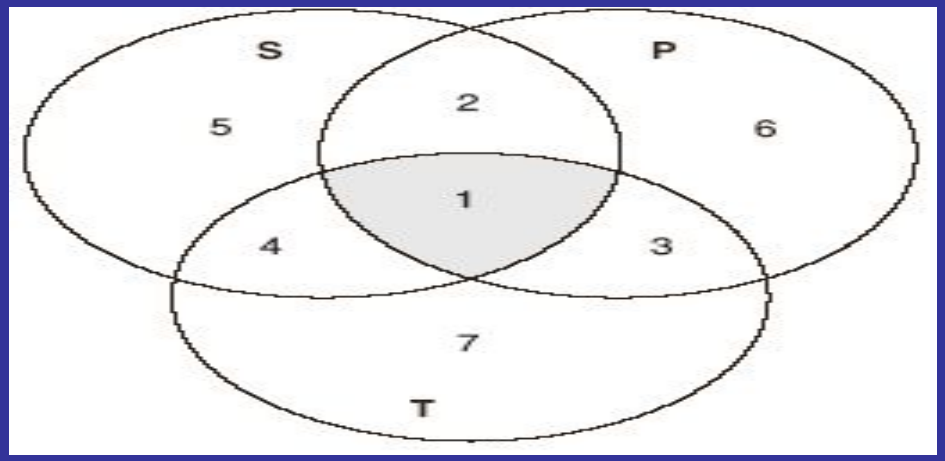
- Effective Testing not Exhaustive Testing
- Testing is not a single phase performed in SDLC
- Destructive approach for constructive testing
- Early Testing is the best policy.
- The probability of the existence of an error in a section of a program is proportional to the number of errors already found in that section.
- Testing strategy should start at the smallest module level and expand toward the whole program.

Testing Principles

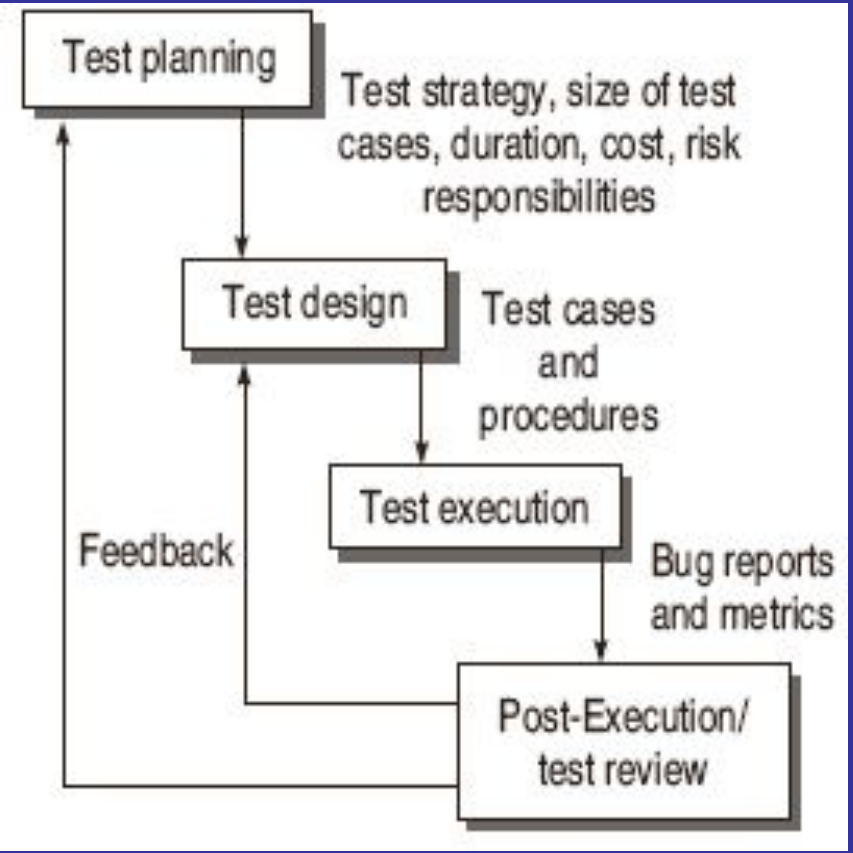
Testing should also be performed by an independent team. Everything must be recorded in software testing.

Invalid inputs and unexpected behavior have a high probability of finding an error.

Testers must participate in specification and design reviews.



Software Testing Life Cycle (STLC)



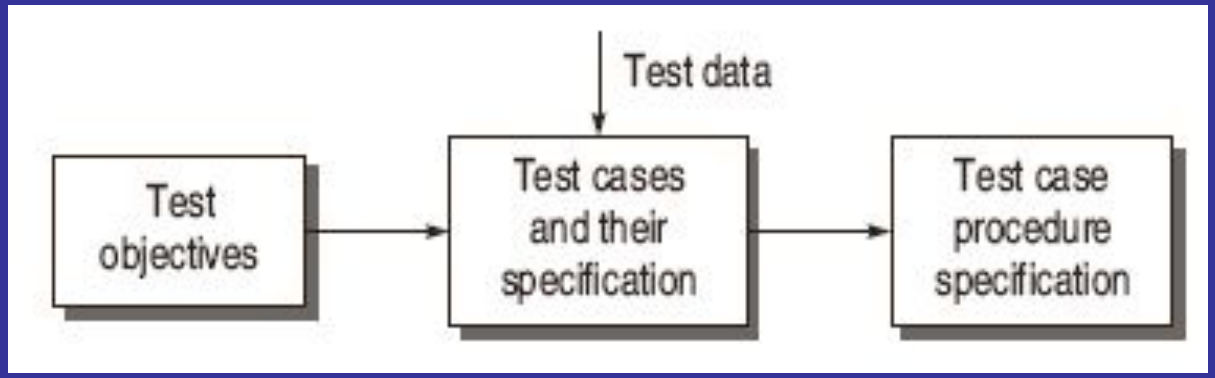
Test Planning

OXFORD
HIGHER EDUCATION

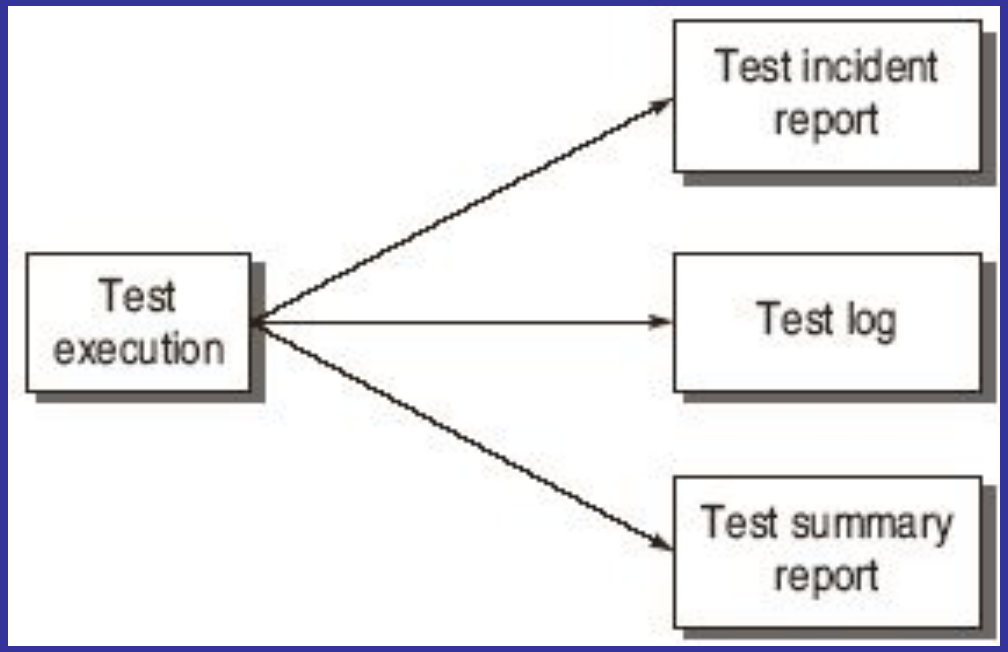
- Defining the Test Strategy
- Estimate of the number of test cases, their duration and cost.
- Plan the resources like the manpower to test, tools required, documents required.
- Identifying areas of risks.
- Defining the test completion criteria.
- Identification of methodologies, techniques and tools for various test cases.
- Identifying reporting procedures, bug classification, databases for testing, Bug Severity levels, project metrics

Test Design

- *Determining the test objectives and their Prioritization*
- *Preparing List of Items to be Tested*
- *Mapping items to test cases*
- *Selection of Test case design techniques*
- *Creating Test Cases and Test Data*
- *Setting up the test environment and supporting tools*
- *Creating Test Procedure Specification*



Test Execution

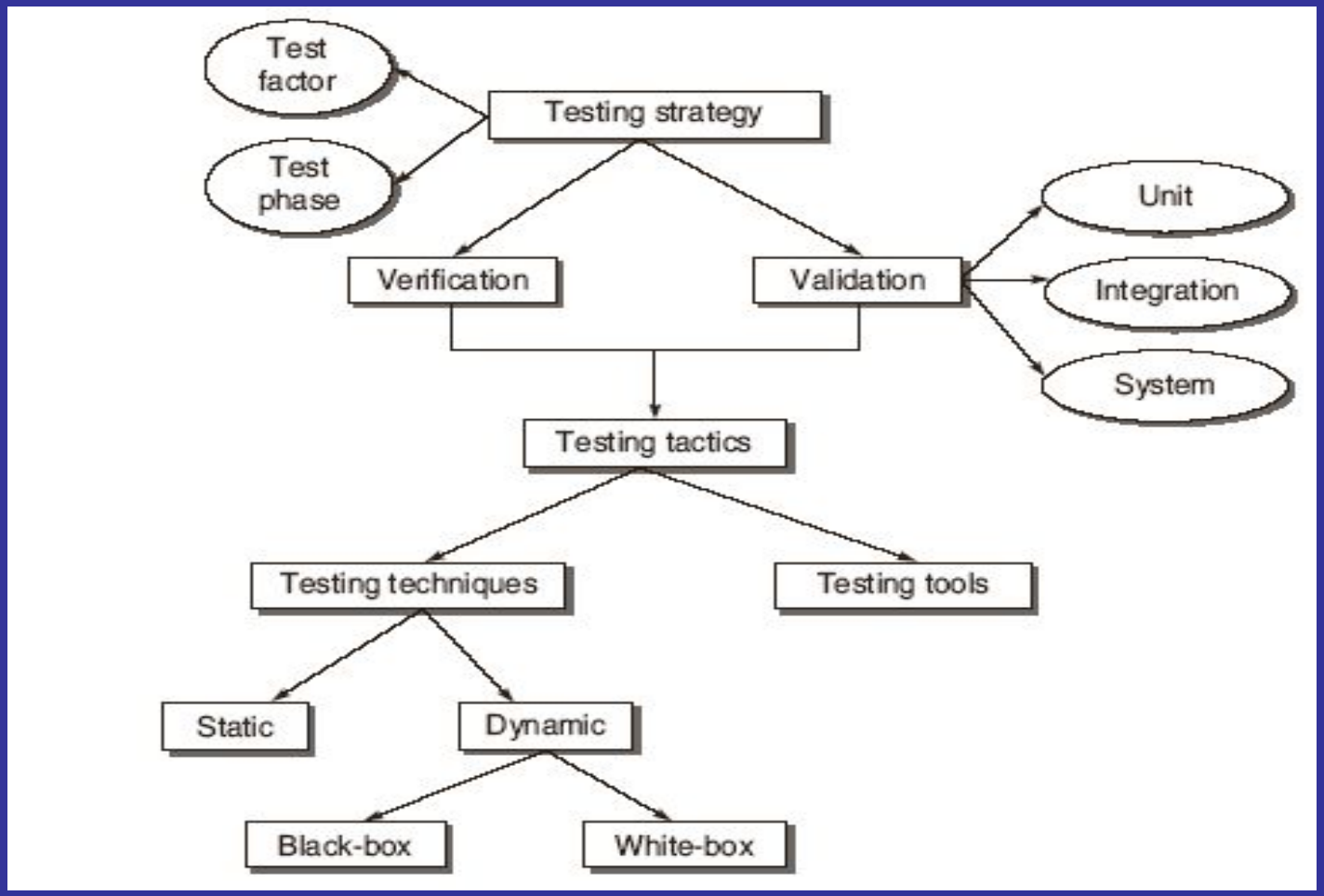


Post-Execution / Test Review

OXFORD
HIGHER EDUCATION

- Understanding the Bug
- Reproducing the bug
- Analyzing the nature and cause of the bug
- *Reliability analysis*
- *Coverage analysis*
- *Overall defect analysis*

Software Testing Methodology



Test Strategy Matrix

- Select and Rank Test Factors
- Identify the System Development Phases
- Identify the Risks associated with System under Development

Test Factors	Test Phase					
	Requirements	Design	Code	Unit test	Integration test	System test
Portability	Is portability feature mentioned in specifications according to different hardware?					Is system testing performed on MIPS and INTEL platforms?
Service Level	Is time frame for booting mentioned?	Is time frame incorporated in design of the module?				

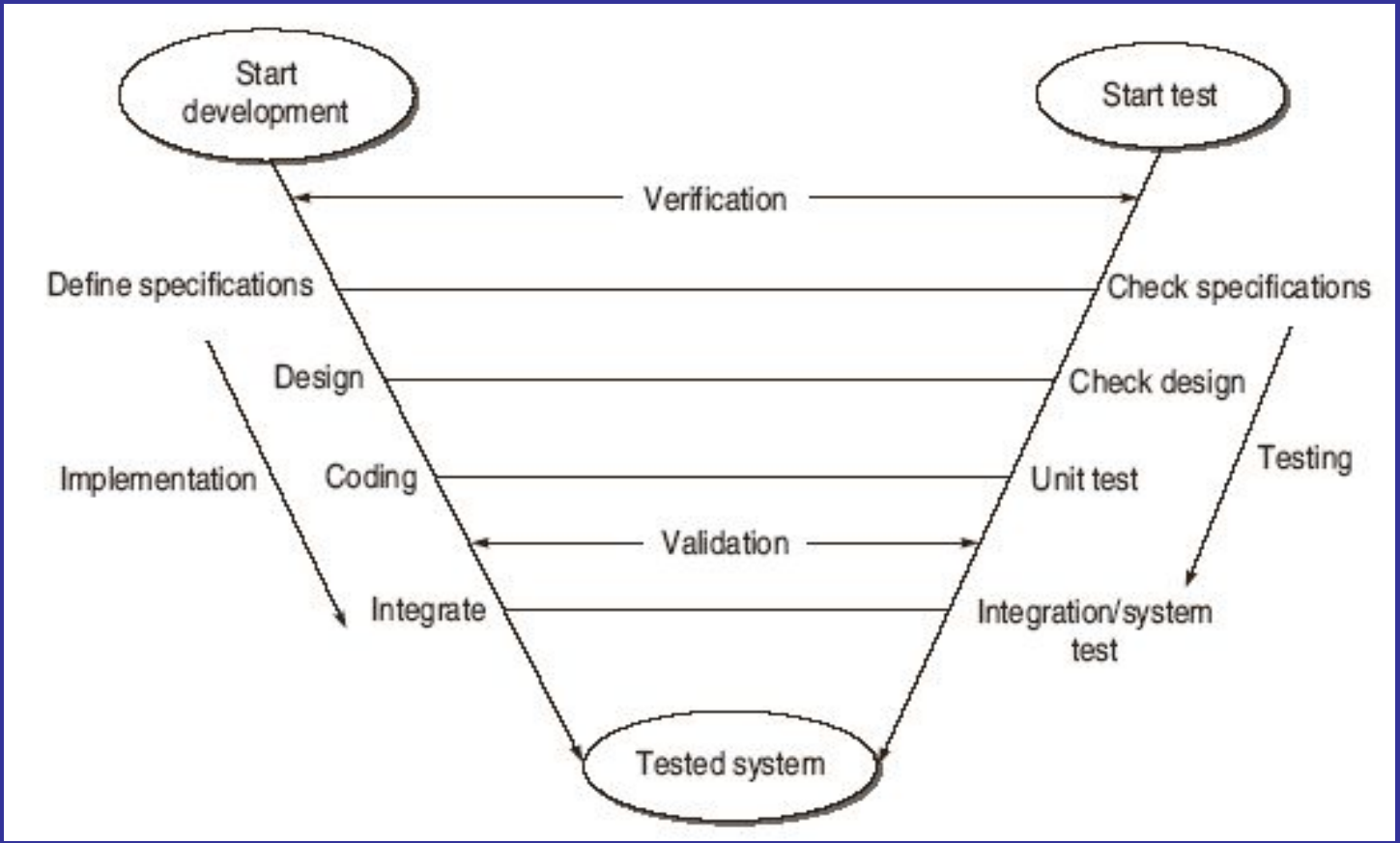
Development of Test Strategy

OXFORD
HIGHER EDUCATION

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

V Testing Life Cycle Model



Validation Activities

- Unit Testing
- Integration Testing
- Function Testing
- System Testing
- Acceptance Testing