- **What is software testing?**

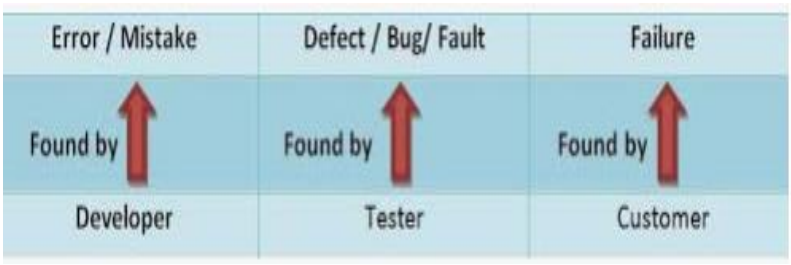
  Software Testing is the process of executing a program with the intent of finding errors

  - Finding defects
  - Trying to break the system
  - Finding and reporting defects
  - Demonstrating correct functionality
  - Demonstrating incorrect functionality
  - Demonstrating robustness, reliability, security, maintainability,
  - Measuring performance, reliability,
  - Evaluating and measuring quality
  - Proving the software correct
  - Executing pre-defined test cases
  - Automatic error detection

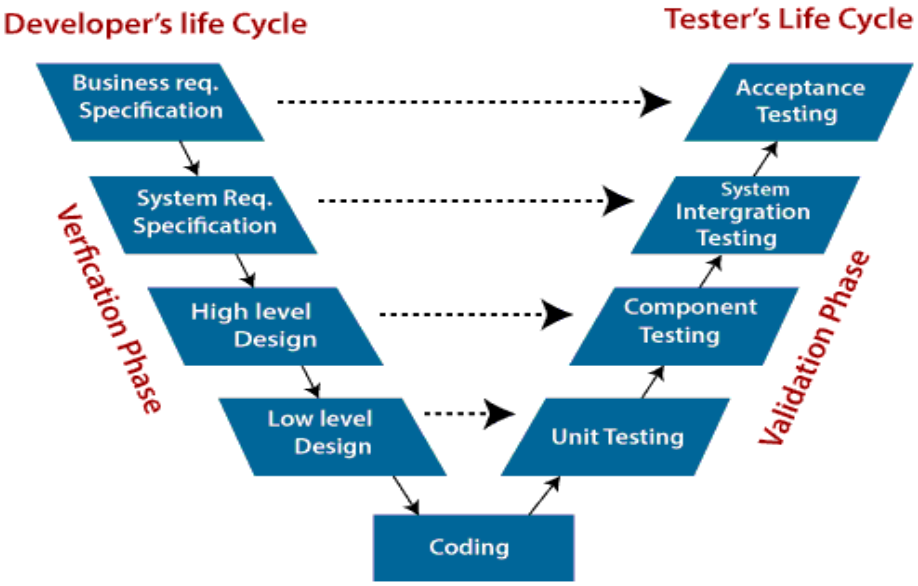
- **List all skills of software tester**.
- Communication skills • Domain knowledge • Desire to learn • Technical skills • Analytical skills • Planning • Integrity• Curiosity• Think from users perspective
- Be a good judge of your product

- **Enlist objectives of software testing.**
  - **Direct Objectives**
    - Identify and reveal as many errors as possible
    - Make the software of **acceptable quality** by removing the errors and retesting.
    - Perform the test effectively and efficiently within budget and schedule constraints.
  - **Indirect Objectives**
    - Compile a record of software errors for use in error prevention in future.

- **Explain the terms Mistake, Error, Defect, Bug, Fault and Failure in relation with software testing.**
- When a defect reaches the end customer it is called a failure and if the defect is detected internally and resolved it's called a defect.

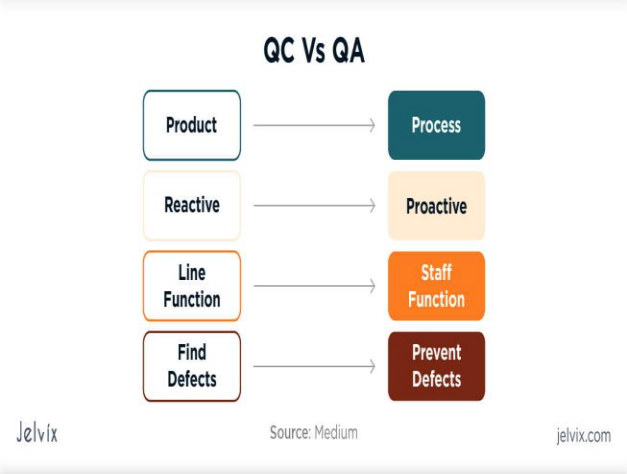


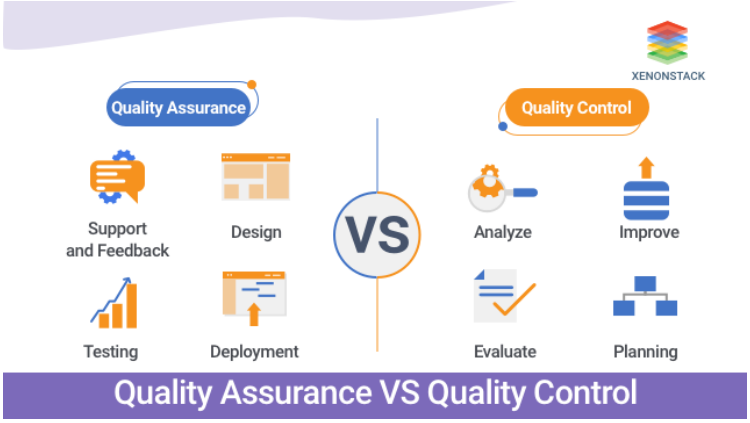
-
- **Enlist different techniques for finding defects and describe any one technique with an example**
- 1) Meeting requirements. 2) Technology expectations. 3) Training / skills. 4) Management aspects
-
- **Describe V-model with labeled diagram.**

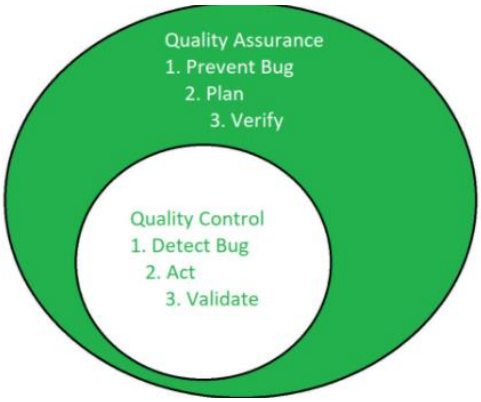


-
  - Advantages:
    - Every stage is tested
  - Disadvantages
    - It assumes that the requirements do not change.
    - The design is not authenticated.
    - The Requirements are not verified.
    - At each stage there is a potential of errors.
    - The first testing is done after the design of modules which is very late and costs a lot.

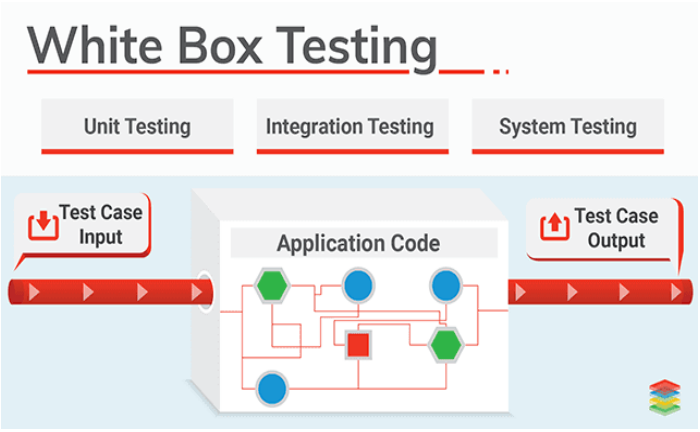
- **Explain quality assurance and quality control**

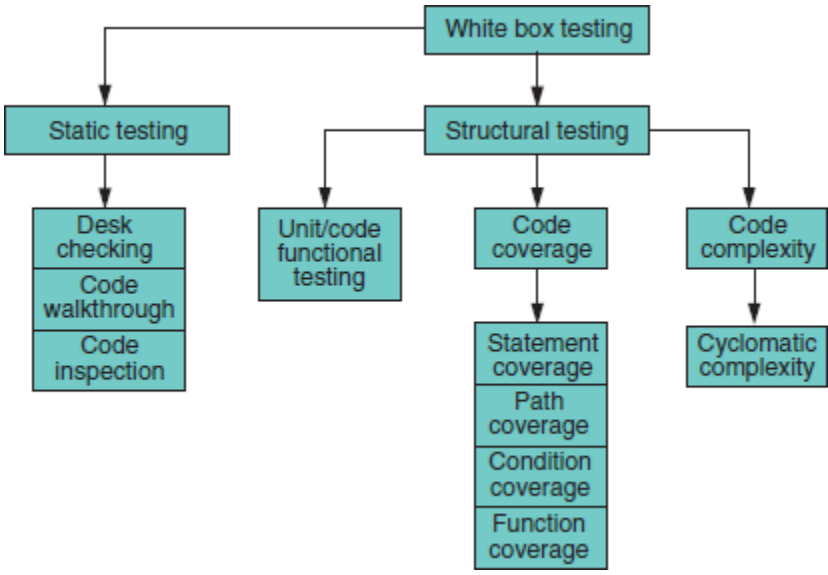






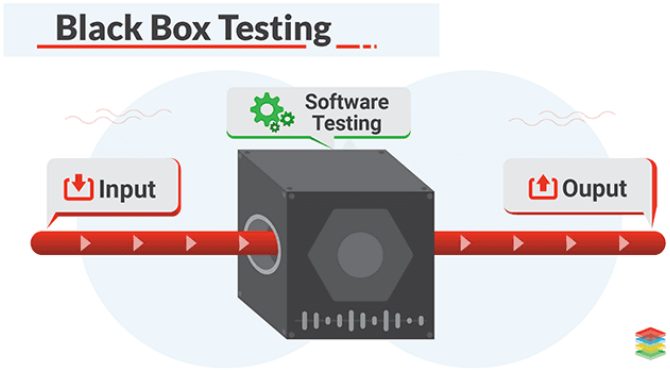
- **Draw classification of White Box testing.**

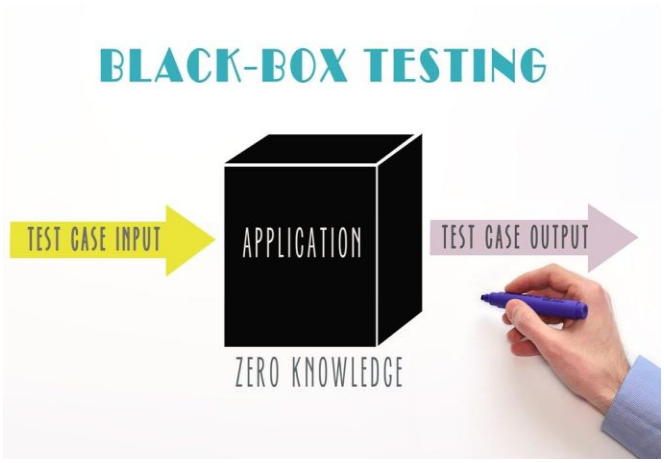


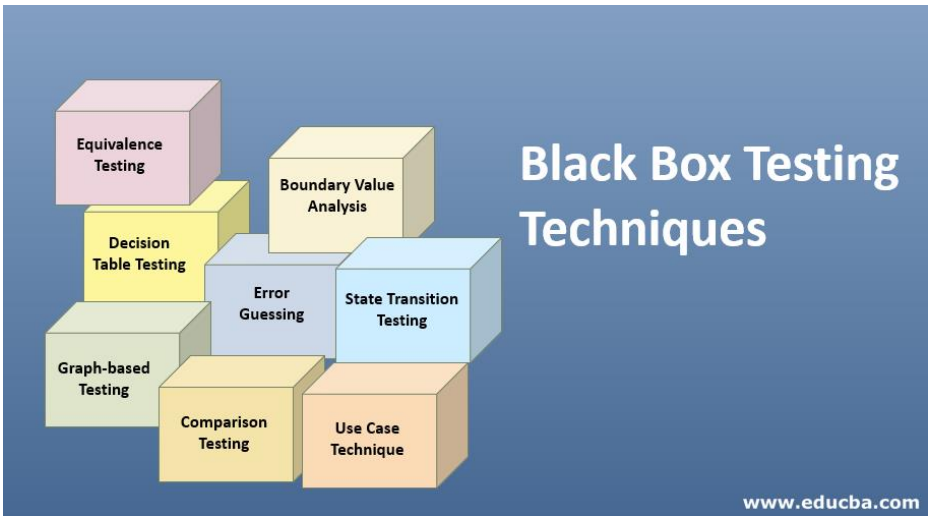


-
-

- **What is Black Box testing?**





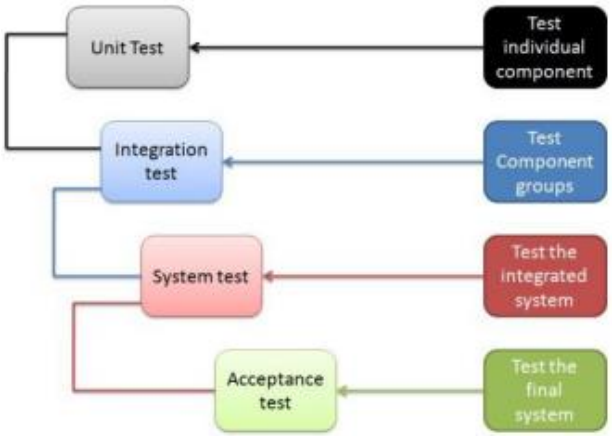


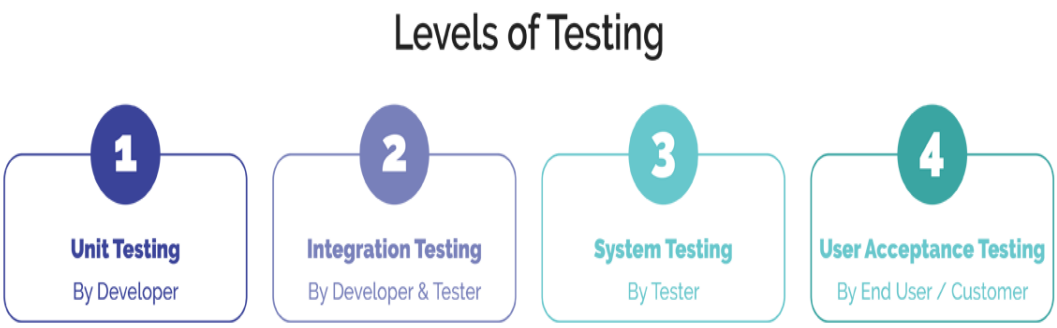
- **Difference between Inspection and Walkthrough**

Walkthrough vs. Inspection

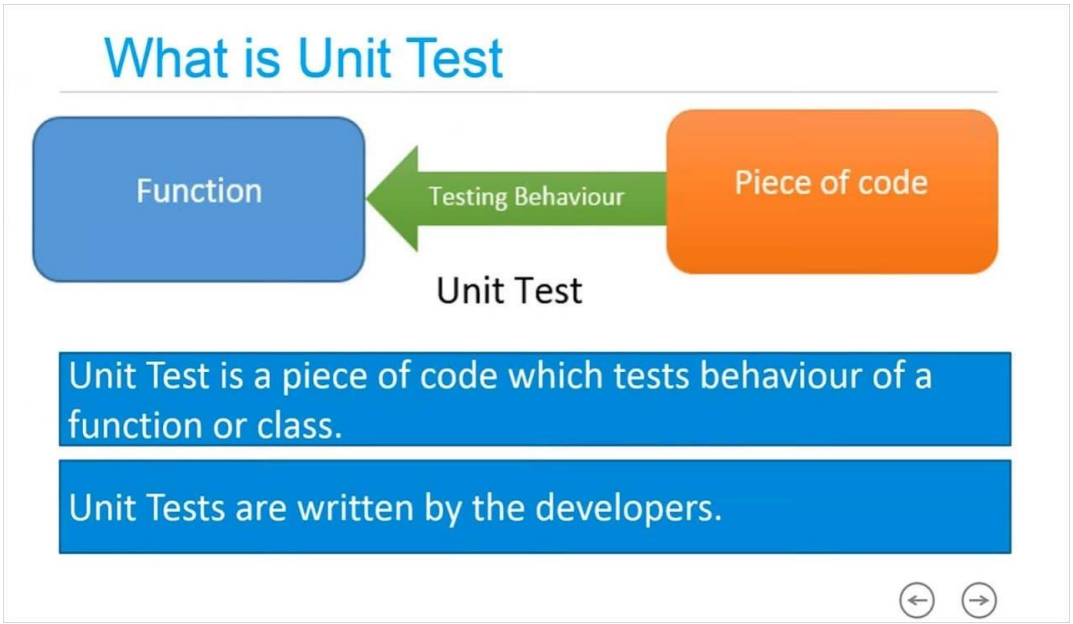
	Walkthrough	Inspection
Focus	**Improve product**	**Find defects**
Activities	Find defects Examine alternatives **Forum for learning** **Discussion**	**Find defects** Only defect explanation allowed **Learning through defects and inspection**
Process	Informal	**Formal**
Quality	Variable; personalities can modify outcome	Repeatable with fixed process
Time	**Preparation ad-hoc, less formal**	Preparation required, efficient use of time

Levels of Testing

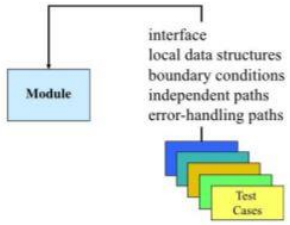


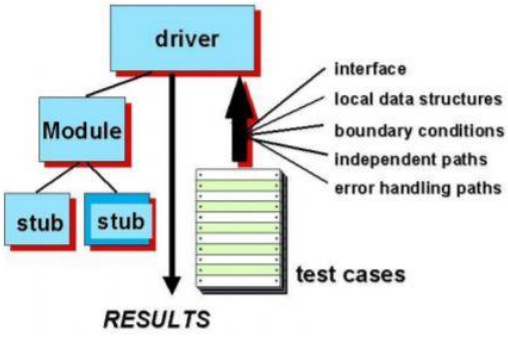


Unit Testing

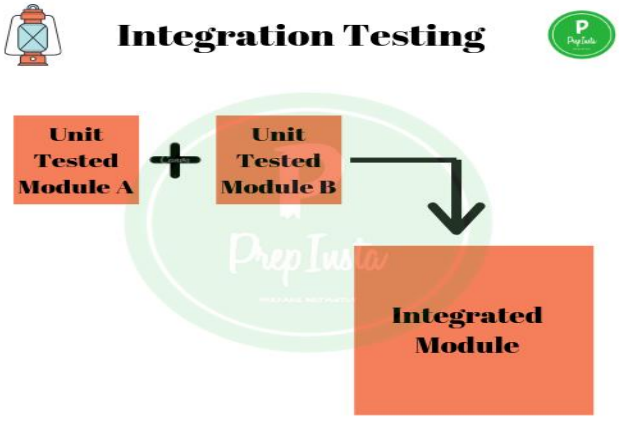


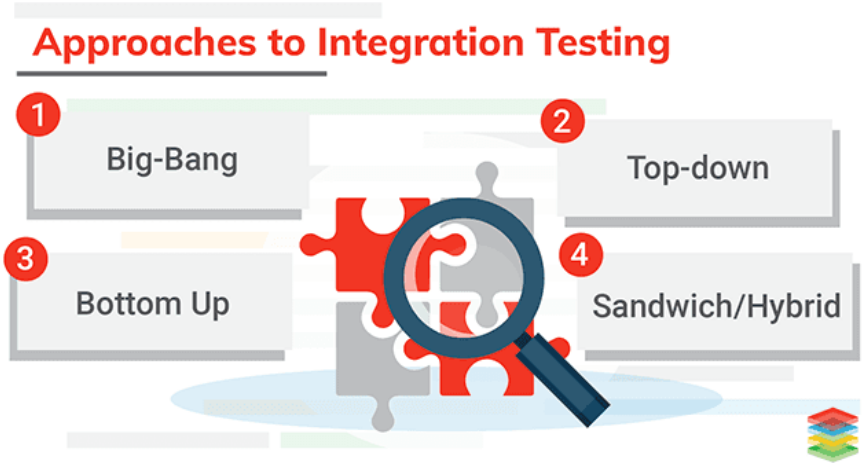
unit testing stub and driver



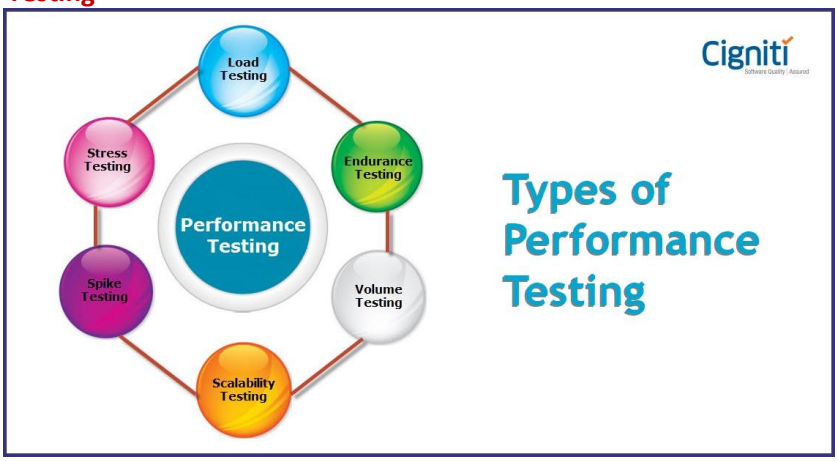


Integration Testing

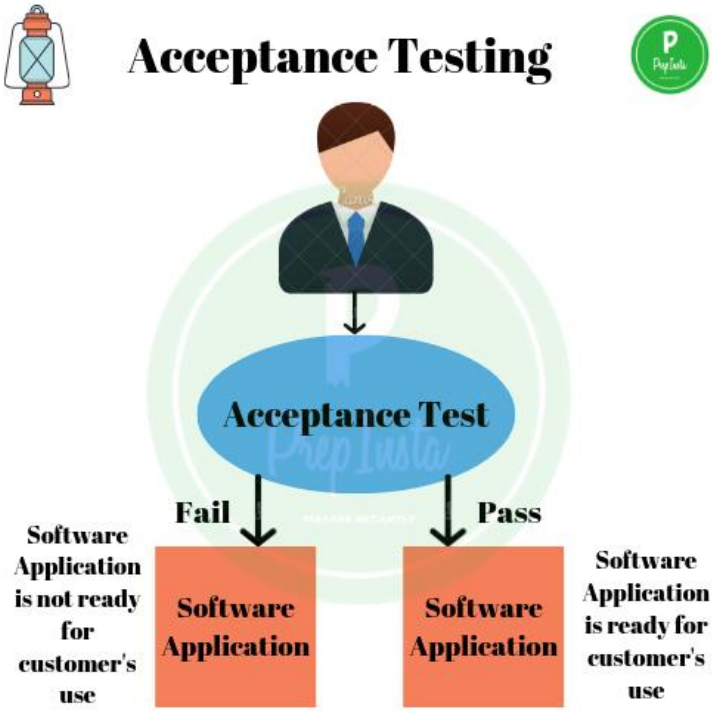




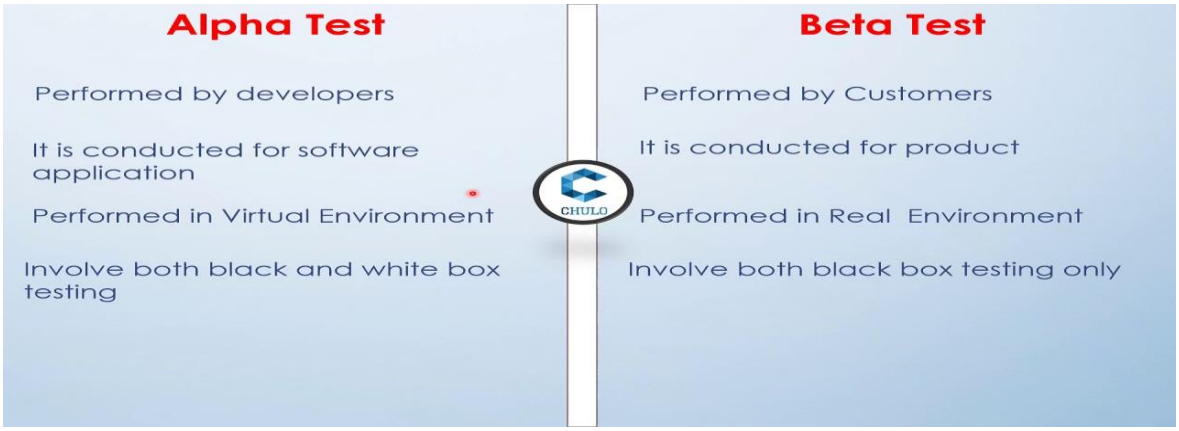
Performance Testing



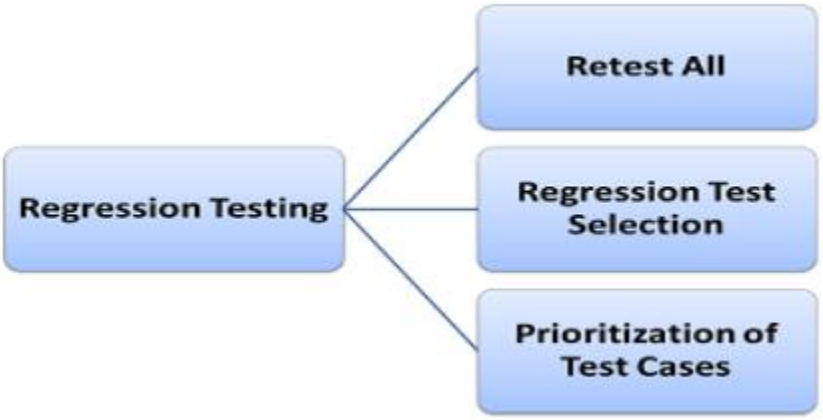
Acceptance Testing



Alpha testing Vs Beta testing

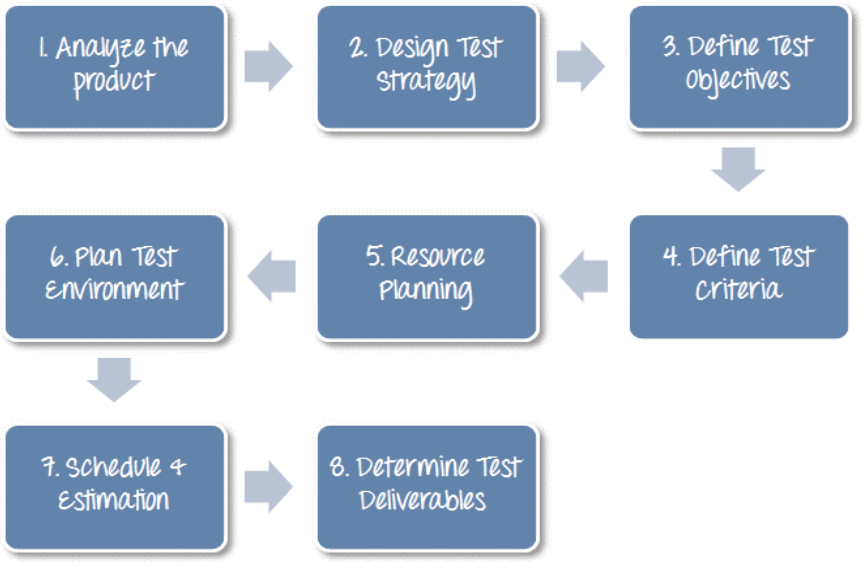


Regression Testing



Chapter 3

Steps for preparing a test plan



TEST PLAN TYPES

Master Test Plan

Testing Level Specific Test Plans : Plans for each level of testing.

– Unit Test Plan

– Integration Test Plan

– System Test Plan

– Acceptance Test Plan

TEST PLAN GUIDELINES

Make the plan concise.

- Be specific.
- Make use of lists and tables wherever possible. Avoid lengthy paragraphs.
- Have the test plan reviewed a number of times prior to base lining it or sending it for approval.
- Update the plan as and when necessary

What is test plan? What is its need? List test planning activities.

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.

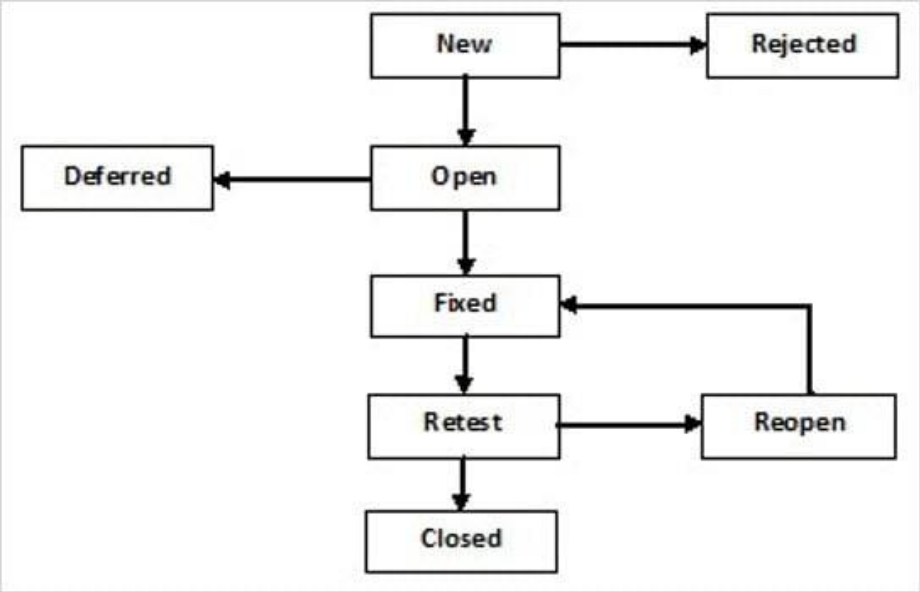
Need of test plan:

- Test Plan Ensures all Functional and Design Requirements are implemented as specified in the documentation.
- To provide a procedure for Unit and System Testing.
- To identify the documentation process for Unit and System Testing.
- To identify the test methods for Unit and System Testing.

Activities

1. Preparing test plan
2. Scope management
3. Deciding Test approach/ strategy
4. Setting up criteria for testing
5. Identifying responsibilities, staffing & Training needs:
6. Identifying Resource Requirement
7. Identifying Test Deliverables
8. Testing task

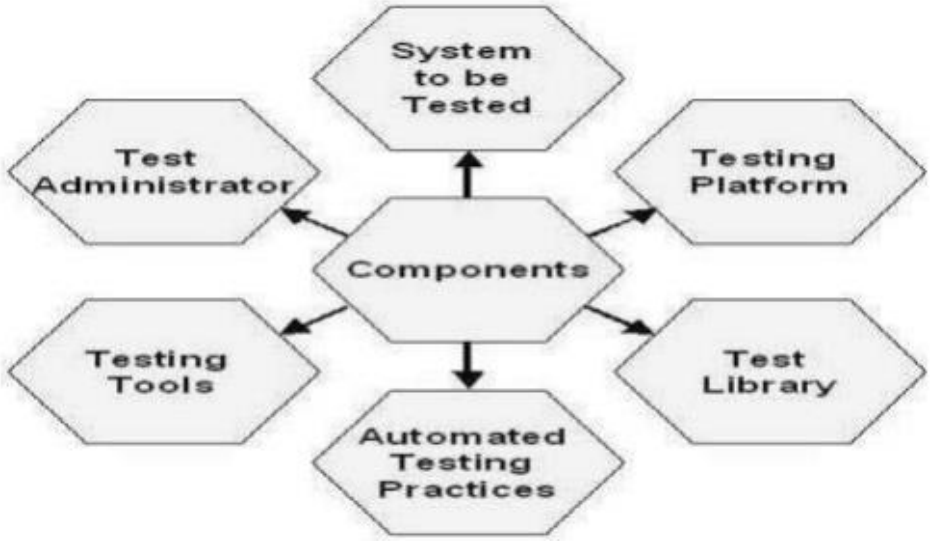
Defect life cycle



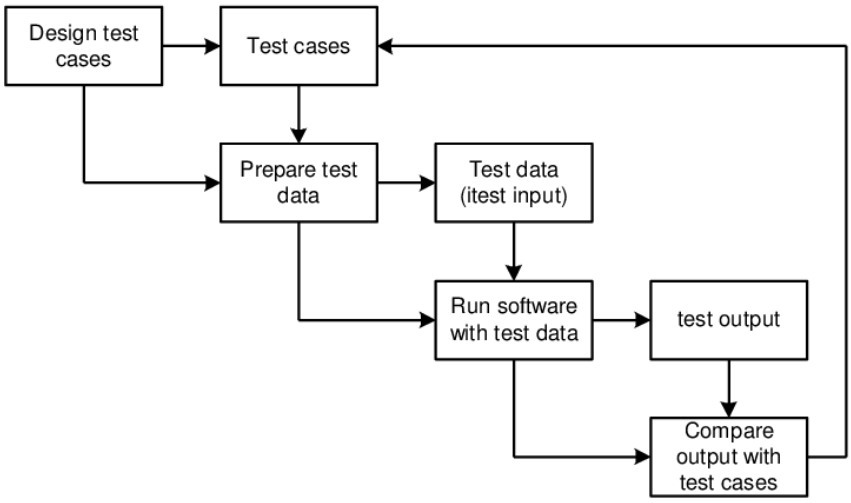
Types of Test deliverables

- Test Plan
- Test cases Documents
- Testing Strategy
- Test Scripts
- Test Data
- Test Traceability Matrix
- Test Results/reports
- Test summary report
- Install/config guides
- Defect Reports
- Release notes

Test Infrastructure Management



Test Process



Test Reporting

- **Test incident report**
- **Test cycle report**
- **Test summary report**

Test Case Specification

Test Case Specification

Test Specification – It is a detailed summary of what scenarios will be tested, how they will be tested, how often they will be tested, and so on and so forth, for a given feature.

Contents of a Test Specification:

Revision History - This section contain information like Who created the test specification? When was it created? When was the last time it was updated?
Feature Description – A brief description of what area is being tested.
What is tested? – An overview of what scenarios are tested.
What is not tested? - Are there any areas that are not being tested.
Nightly Test Cases – A list of the test cases and high-level description of what is tested whenever a new build becomes available.
Breakout of Major Test Areas - It is the most interesting part of the test specification where testers arrange test cases according to what they are testing.
Specific Functionality Tests – Tests to verify the feature is working according to the design specification. This area also includes verifying error conditions.
Security tests – Any tests that are related to security.
Accessibility Tests – Any tests that are related to accessibility.
Performance Tests - This section includes verifying any performance requirements for your feature.
Localization / Globalization - tests to ensure you're meeting your product's Local and International requirements.
NOTE: Test Specification document should prioritize the test case easily like nightly test cases, weekly test cases and full test pass etc:
Nightly - Must run whenever a new build is available.
Weekly - Other major functionality tests run once every three or four builds.
Lower priority - Run once every major coding milestone.

What is defect?

A defect is an error or a bug, in the application which is created

different techniques to find defects

1. Static Techniques: Testing that is done without physically executing a program or system. A code review, walkthrough, inspections etc. are the examples of static testing technique.

2. Dynamic Techniques: Testing in which system components are physically executed to identify defects. Execution of test cases is an example of a dynamic testing technique.

3. Operational Techniques: An operational system produces a deliverable containing a defect found by users, customers, or control personnel i.e., the defect is found as a result of a failure.

List of Defect Classification:

1. Requirements and specification defect

2. Design Defects

3. Coding Defects

4. Testing Defect

Software Defects/ Bugs are normally classified as per:

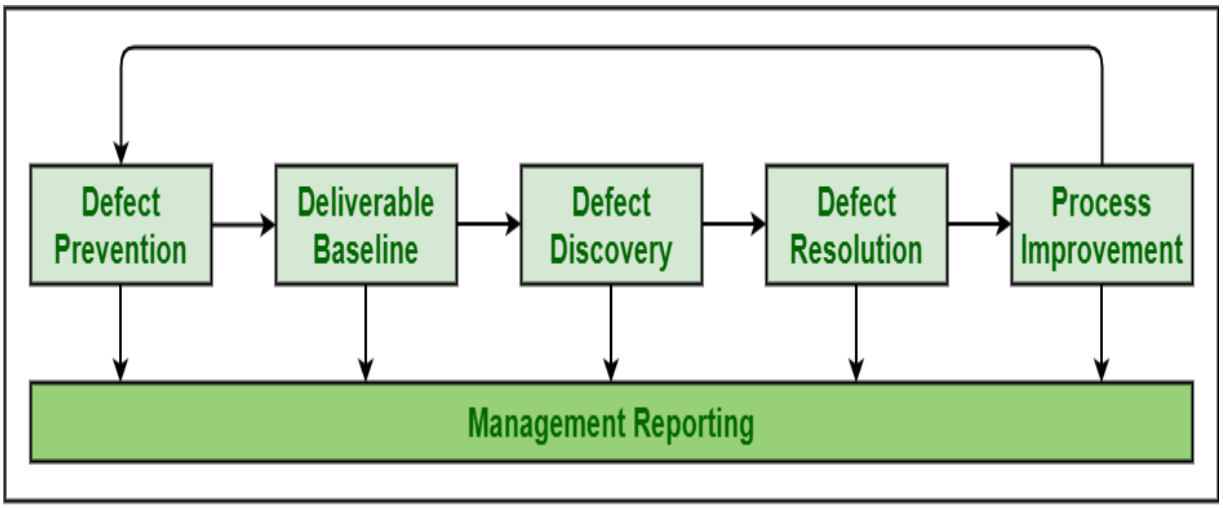
- **Severity / Impact**
- **Probability / Visibility**
- **Priority / Urgency**
- **Related Dimension of Quality**
- **Related Module / Component**
- **Phase Detected**
- **Phase Injected**

techniques of finding bugs.

1.Static testing

2.Dynamic testing

3.Operational testing



Defect Management Process

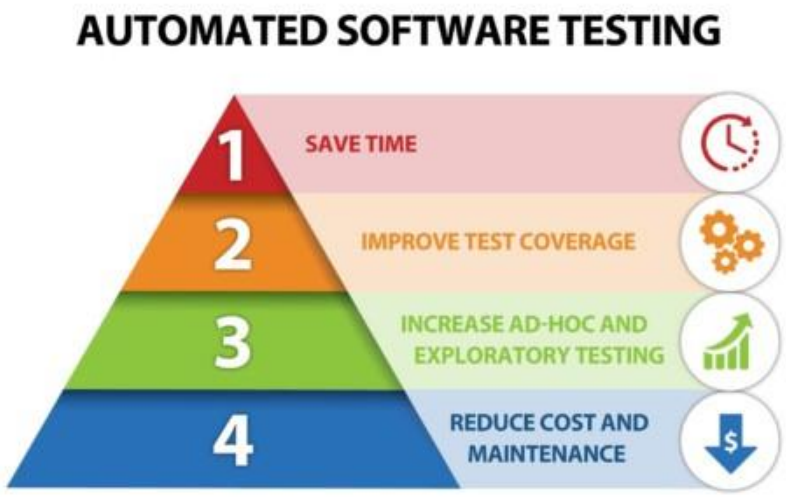
Manual Testing

Advantages of using manual testing tools :

- **Reduce time of testing**
- **Improve the bugs finding**
- **Deliver the quality software/product**
- **Allow to run tests many times with different data**
- **Getting more time for test planning**

- Save resources or reduce requirement
- It is never tired and expert person can work at a time many tools

Automation Testing



Benefits of using testing tools:

- Save Time
- Speed
- Repeatability
- Maintenance of the test suite
- Reusable
- Increase Coverage
- Cost Reduction

Static testing tool	Dynamic testing tool
These tools are used by developers as part of the development and component testing process	These tools require the code to be in a "running state"
code is not executed or run but tool itself is executed	They analyse rather than testing
It is extension of compiler technology	They also help to understand background processes
It also perform static analysis of requirement or analysis of website	These tool used by developed in component integration testing,, middle ware , testing robustness and security.
Helps to understand the structure of the code and can also be useful to enforce coding standards.	Also performs web site testing to check whether each link does actually link to something else, it can find dead links .
Features /characteristics of static testing tools are: • Checks cyclomatic complexity • Enforces coding standards • Analyse structures and dependencies • Helpful in understanding coding • Identify defects in code.	Features/characteristics of static testing tools are: • Detect memory leak • Identify pointer arithmetic errors , null pointer • Identify time dependence.
Examples. Flow analyzer, path tests, coverage analyzers, Interface analyzers	Examples. Test driver, Test beds, Emulators, Mutation analyzers