Software Testing Methodologies

Types of Testing

Types Of Testing

Black box testing:

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

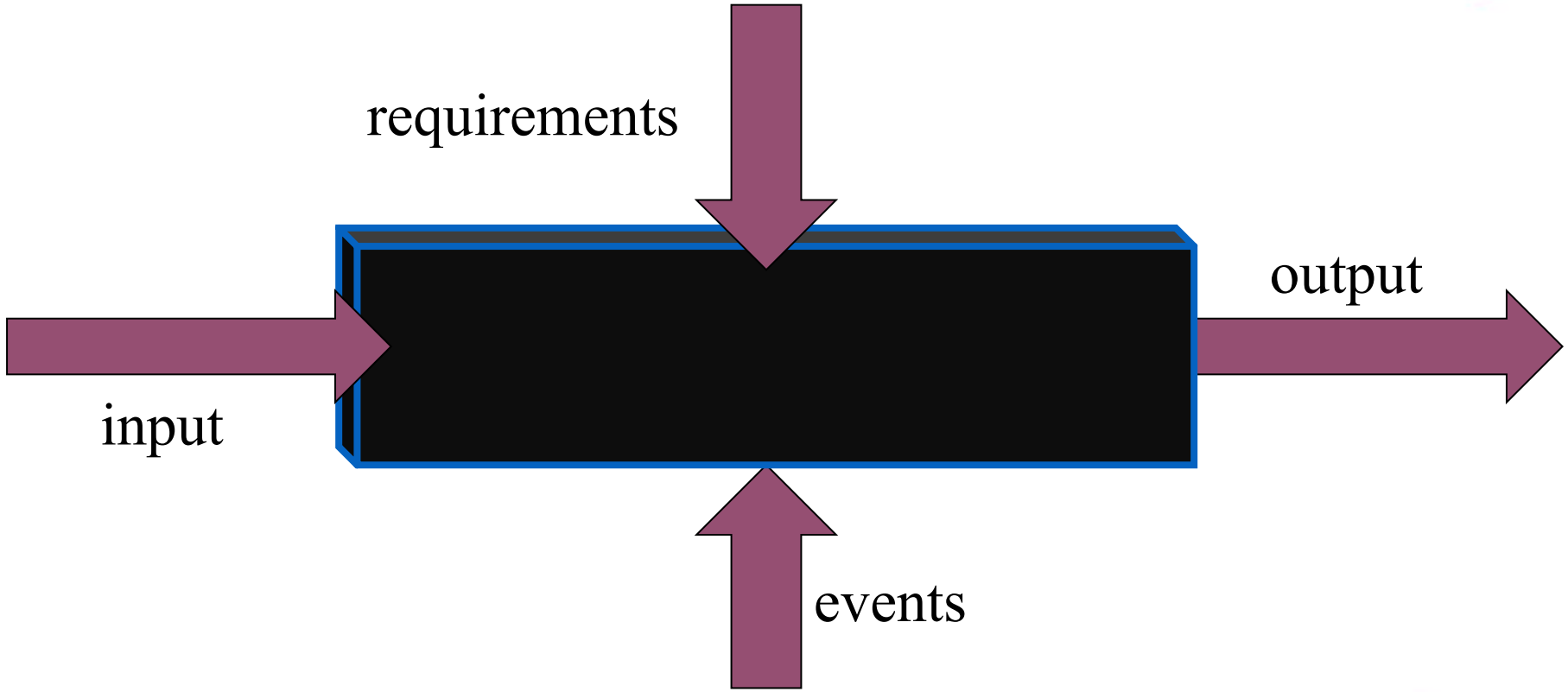
White box testing:

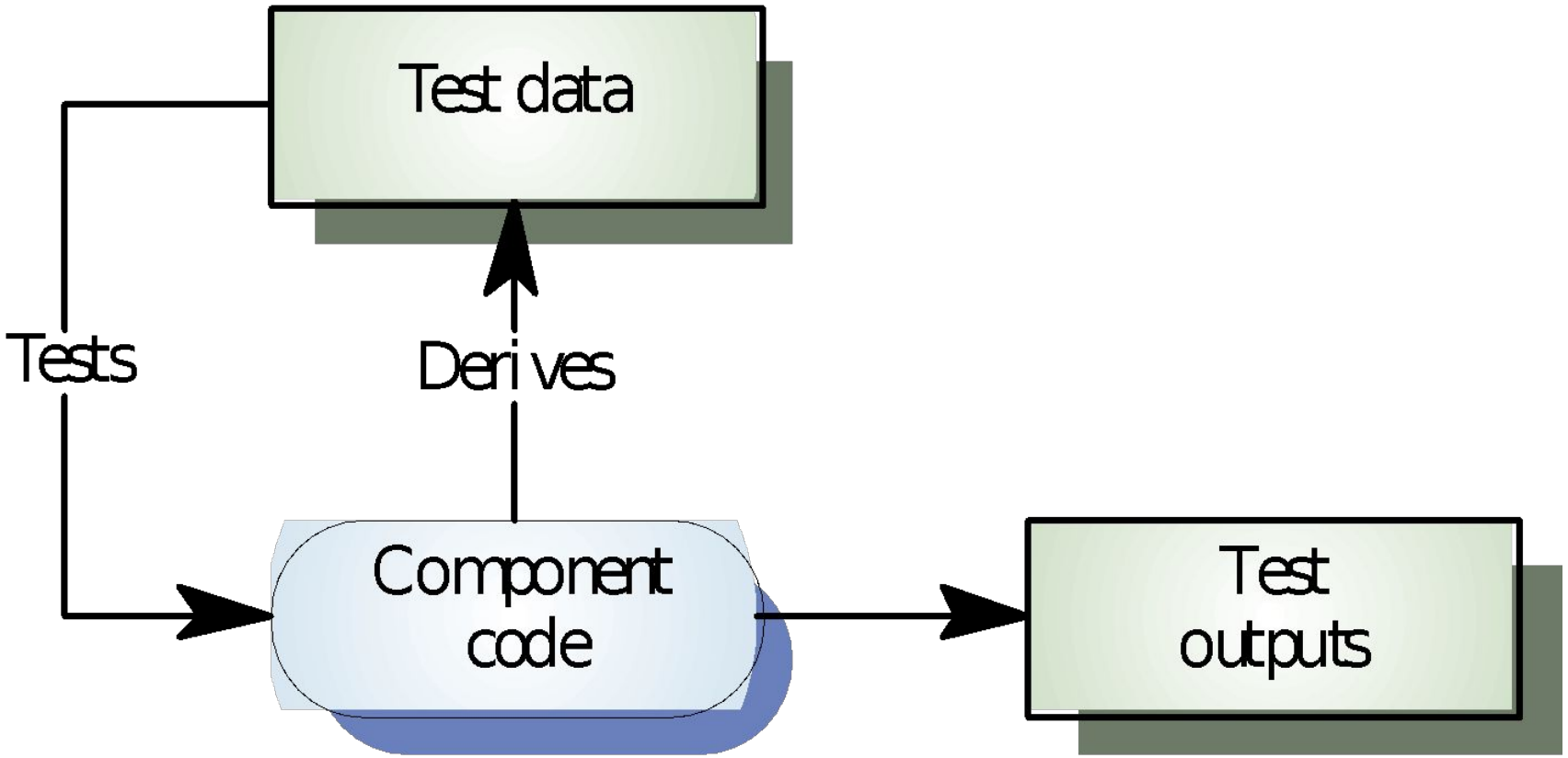
- Knowledge of the internal program design and code required
- Tests are based on coverage of code statements, branches, paths and, conditions

Black Box Testing Types

- Functional Testing
- Non-functional Testing
- Others

Black box testing




Test data
Tests
Derives
Component code
Test outputs

Functional Testing

- The objective of functional testing is to make sure that the software application meets the intended requirements
- The tester knows only the inputs and the expected outputs of the functions
- Functional testing can be done both manually and using automated tools

Functional Testing

Functional Testing Types:

- Smoke /Sanity Testing
- Retesting
- Regression Testing
- Usability Testing

Smoke/Sanity Testing

- It’s a quick test that ensures, whether the major functions of the software work without bothering with finer details

Smoke/Sanity Testing difference

Smoke Testing	Sanity testing
In software industry, smoke testing is a shallow and wide approach whereby all areas of the application without getting into too deep, is tested.	A sanity test is a narrow regression test that focuses on one or a few areas of functionality. Sanity testing is usually narrow and deep.
A smoke test is scripted, either using a written set of tests or an automated test	A sanity test is usually unscripted.
A Smoke test is designed to touch every part of the application in a cursory way. It's shallow and wide.	A Sanity test is used to determine a small section of the application is still working after a minor change.
Smoke testing is conducted to ensure whether the most crucial functions of a program are working, but not bothering with finer details. (Such as build verification).	Sanity testing is a cursory testing; it is performed whenever a cursory testing is sufficient to prove the application is functioning according to specifications. This level of testing is a subset of regression testing.
Smoke testing is normal health check up to a build of an application before taking it to testing in depth.	Sanity testing is to verify whether requirements are met or not, checking all features breadth-first.

Retesting

- Re-execution of test cases on same application build with different inputs or test data

Regression Testing

- Regression testing is the re-execution of some subset of tests that have already been conducted in a functional or system test to ensure that changes to design or code have not generated any new defects or broken any existing functionality
- Regression may be conducted manually, by executing a subset of all test cases or using automated testing tools

Usability Testing

- Usability testing is done to test whether end users can use the application with minimum stress and maximum efficiency.
- Ensures that the application is easy to work with, limits keystrokes and is easy to understand
- Usability testing has to be preferably done early in the lifecycle so that user interface and workflow changes do not cause substantial rework later.

Performance Testing

Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or device

This process can involve

- Measuring the response time or Qualitative attributes such as reliability, scalability and interoperability may also be evaluated
- The process can compare two or more devices or programs in terms of parameters such as speed, data transfer rate, bandwidth, throughput, efficiency or reliability

Load Runner

- We use Load Runner as a tool for Load Testing

Security Testing

Security testing attempts to check that access control and protection mechanisms are built properly into a system

In Security testing, password cracking, unauthorized entry into the software, network security are all taken into consideration

Other Testing Types

Other Testing Types:

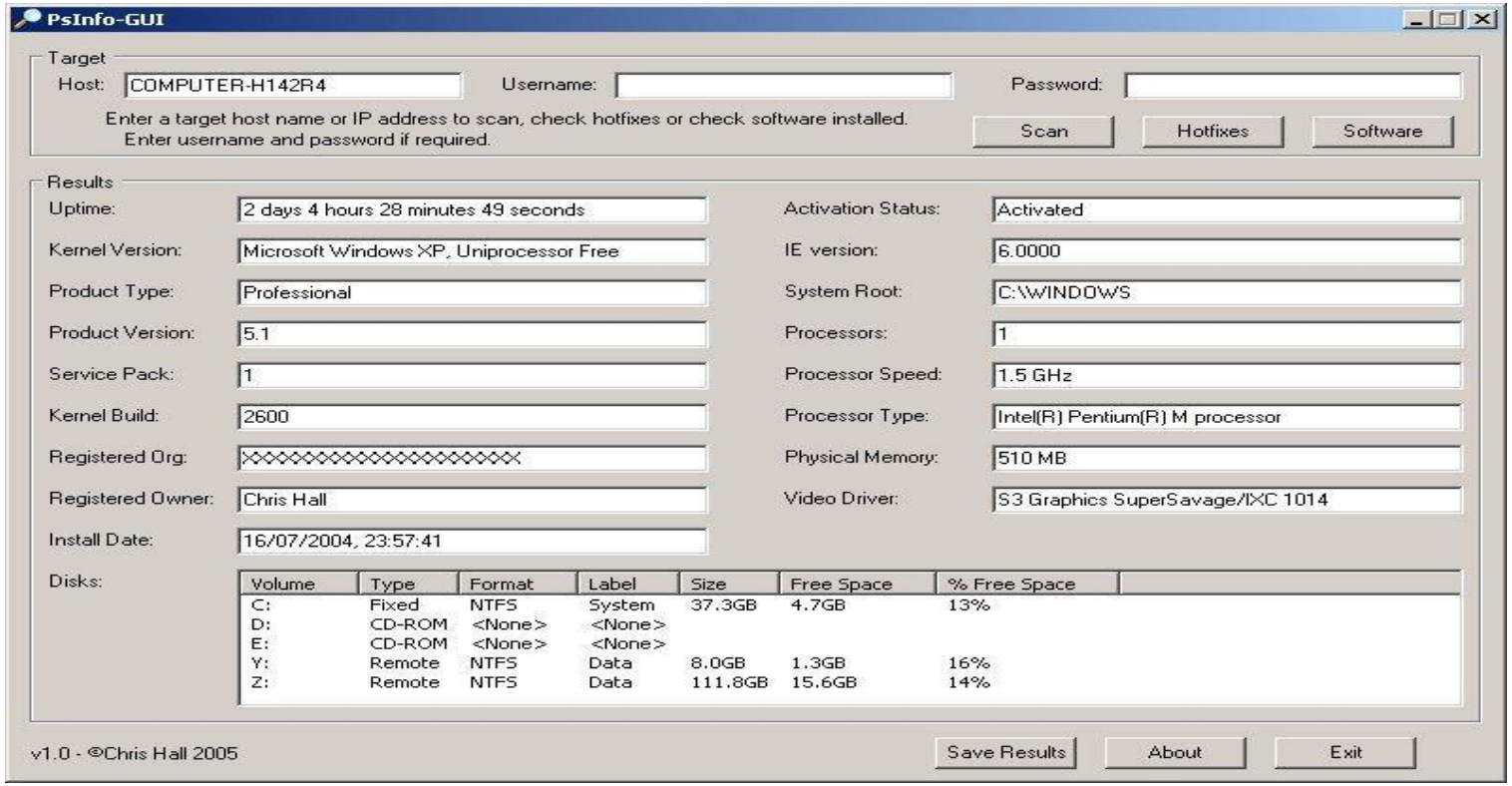
- GUI Testing
- Adhoc Testing
- Database Testing

GUI Testing

- GUI software testing is the process of testing a product that uses a graphical user interface, to ensure it meets its written specifications.
- That is, testing how the application and the user interact. how it displays screen text, images, buttons, menus, dialog boxes, icons, toolbars and more.
- Most clients in client/server and web-based systems deliver system functionality using a GUI.

GUI Testing

- GUI testing is used to validate every screen in terms of ease of use (understandability of screen)
- Look and feel (attractiveness of screen)
- Less no. of events to
- complete a task
- i.e. short navigation



Good GUI

List of seven important traits to a good GUI

Following standards and guidelines

- Consistent
- Flexible
- Comfortable
- Correct
- Useful

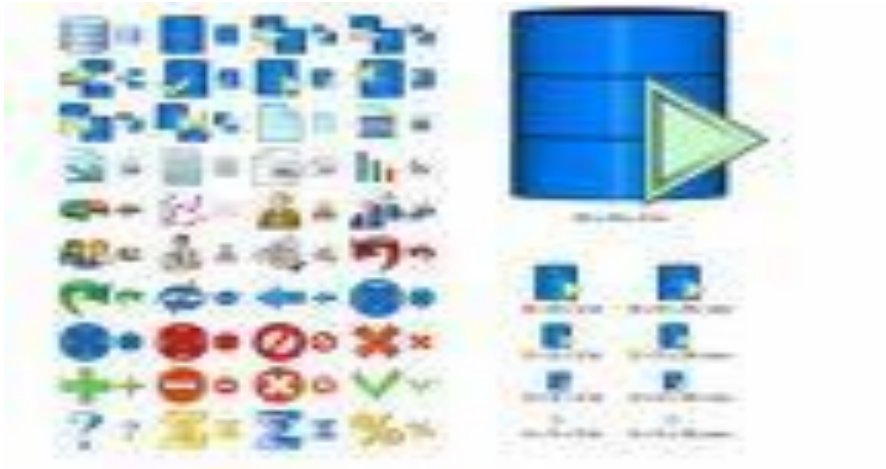
Adhoc Testing

- It's a random testing It means testing a application without proper test plan and documentation.
- Without preparing and executing test cases
- For Ad hoc testing one should have strong knowledge about the application
- Adhoc is unstructured and inconsistent level of testing.
- Adhoc testing is a part of exploratory testing
- Adhoc testing is most often used as a complement to other types of testing.

Database Testing

Database testing is nothing but testing the front end field effect on backend database tables and comparing whether it is changing pertaining to it or not

- Check the Fields are defined Correctly in the Respective tables.
- Check the Field length
- Check the Relationship between tables
- Check the Dependencies between tables



Stress testing

- Is running the software under less than ideal condition
- Goal is starve the software
- This is done by exercising the product close to or beyond its limits .
- Low memory, low disk space, slow CPUs, slow modems and so on
- Does this sound like boundary condition? (it is).

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

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

Compatibility testing

- Testing how well software performs in a particular HW/SW/OS/NW environment

Comparison testing

- Comparing SW strengths and weakness to competing products.

Mutation testing

- To determining if a set of test data or test cases is useful, by deliberately introducing various bugs
- Re-testing with the original test data/cases to determine if the bugs are detected

Recovery Test

Confirms that the system recovers from expected or unexpected events without loss of data or functionality.
Example-

- Shortage of disk space
- Unexpected loss of communication
- Power out conditions

Grey Box Testing

Gray/grey box testing is a software testing technique that uses a combination of black box testing and white box testing

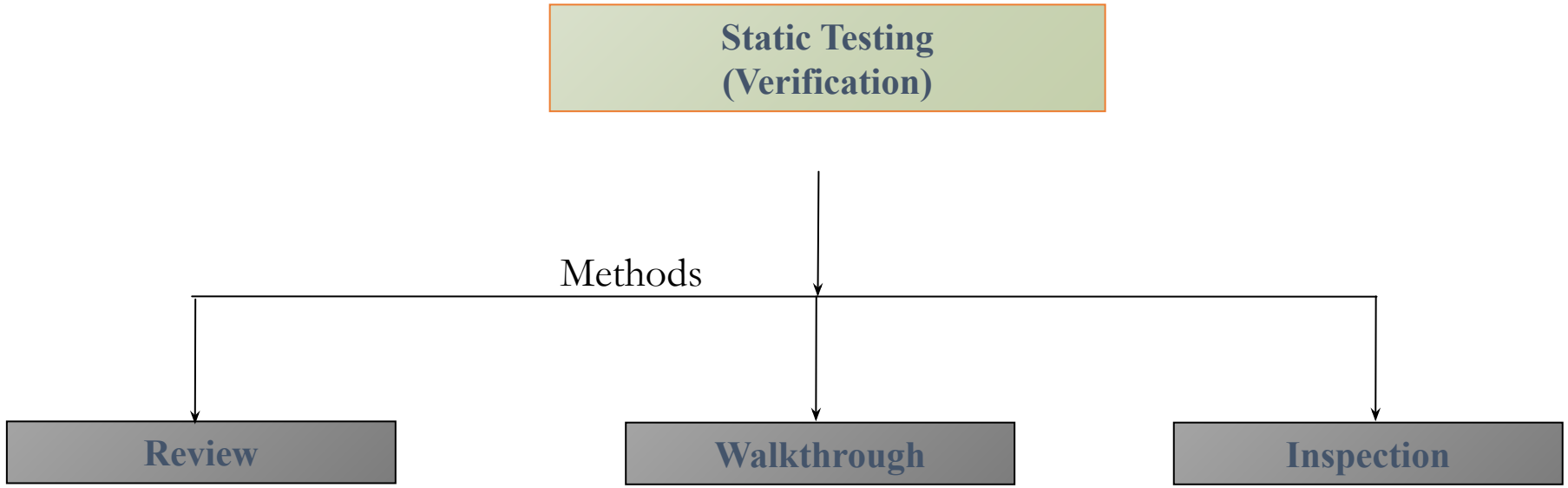
Testing Category

Testing can be categorized as follow:

- Static Testing (Verification)

- Dynamic Testing (Validation)

Static Testing (Verification)



Static Testing (Verification)

Static Testing: Testing of a component or system at specification or implementation level without execution of that software

Static Testing methods:

- **Review:** An evaluation of a product or project status to ascertain discrepancies from planned results and to recommend improvements.
- **Inspection:** A type of peer review that relies on visual examination of documents to detect defects.
- **Walkthrough:** A step-by-step presentation by the author of a document in order to gather information and to establish a common understanding of its content.

Review

Process or meeting during which a work product, or its set is presented to Project personnel, managers, users, customers, or other interested parties for comment or approval

- Inprocess Review
- Decision Point or Phase End Review
- Software requirement Review
- Critical design Review
- Test readiness Review
- Post Implementation Review
- Management Review
- Requirement Review
- Design Review
- Code Review
- Adhoc Review

Inspection

Inspection

- Inspections can be used for verifying the products of any development process for detection of defects

Types of Inspection

- **Design Inspection:** in detailed designs before coding
- **Code Inspection:** in code before testing,
- **Test ware Inspection:** in test designs, test cases and test procedures

Inspection

Preparation

- Inspection team gets code, docs
- Focus on specific component
- Several days prior to inspection

Inspection

- Moderator runs meeting
- Inspectors paraphrase code
- Scribe notes defects
- Developer stays quiet
- May be asked for clarification

Inspection

Rework:

- Developer makes changes
- Address all defects identified by inspection

Follow-Up:

- Moderator makes sure changes in place
- May bring in other team members if required
- May call for re-inspection

Walkthrough

Walkthrough

Less formal than Inspection

- Identify defects, make suggestions

Team “walks through” the code

- Basically a manual simulation of code

Driven by (simple) test data

- Not an exhaustive test of program
- Some specificity to spark discussion

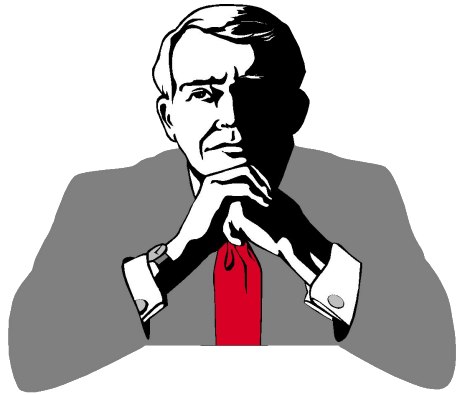
Developer present

- May be questioned about rationale

Walk-through

Objectives of Walk-through:

- Detect errors early
- Ensure (re) established standards are followed.
- Train and exchange technical information among project teams which participate in the walkthrough
- Increase the quality of the project, thereby improving morale of the team members



Walk-through

Exit criteria for the Walk-through:

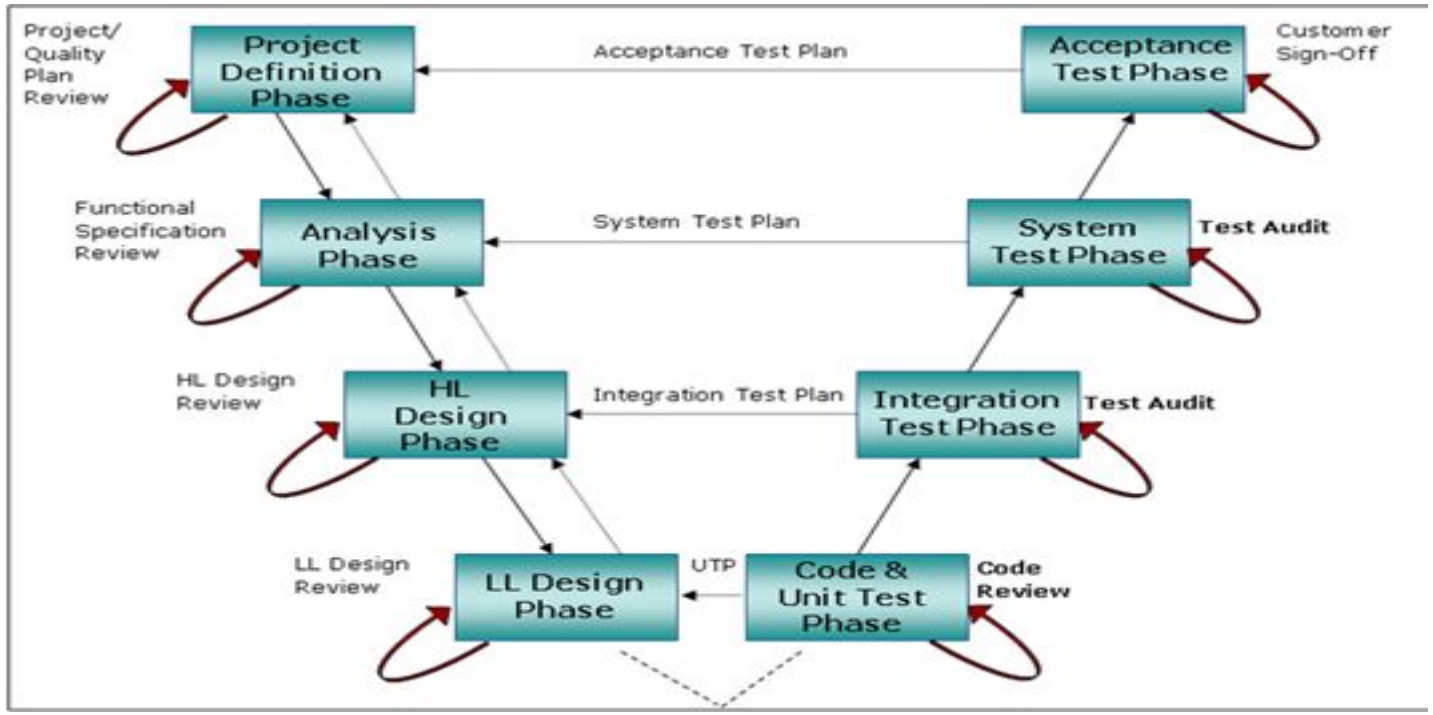
- The entire software product has been examined
- Recommendations and required actions have been recorded
- The walk-through output has been completed

Audit

Audit - Audits are independent reviews that assess compliance with software requirements, specifications, baselines, standards, procedures, instructions, codes and contractual and licensing requirements

Objectives - The objective of an audit is to verify that software products and processes comply with standards, guidelines, specifications and procedures

Static Testing (Verification)



Dynamic Testing

- Dynamic Testing (Validation)
- Levels of dynamic testing
- Unit testing
- Integration testing & their types
- System testing
- Acceptance testing & their types

Dynamic Testing (Validation)

Dynamic Testing (Validation):

Testing that involves the execution of the software of a component or system

Levels of Dynamic Testing (Validation):

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

Methods of Dynamic Testing (Validation):

- White box testing
- Black box testing

Thankyou!