

Software Testing and Quality Analysis

Module Supervisor: Mrs. M. Halassa

**Chapter 2:
Software Test Methods
&
Test Activity Planning**



Course Outline

1. The different levels of testing

- Unit testing
- Integration Test
- System testing.
- Acceptance testing

2. Classification of Testing methods

- Static Test Methods
- Dynamic Test Methods

3. Fundamental testing process

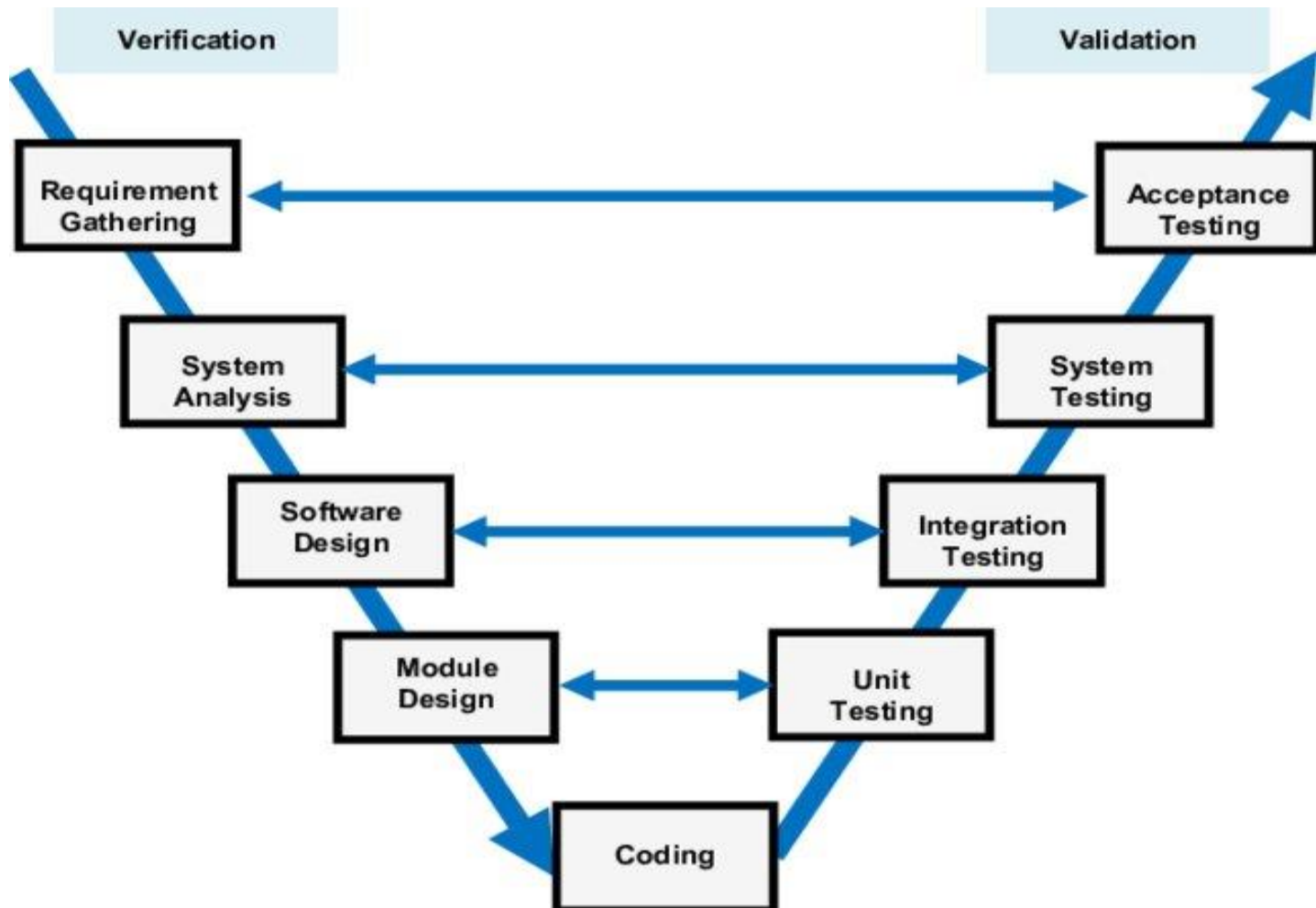
4. Conclusion

Introduction

- ❖ Testing is not limited to running the software in order to identify failures: it is also necessary to plan, set objectives, design test conditions, provide test data, test environments and of course to control all this.

I- The different levels of testing

V-Model life cycle for software testing



Unit testing

- In this type of testing, errors are detected individually from every component or unit by individually testing the **components** or **units** of software to ensure that they are fit for use by the developers.
- It is the smallest testable part of the software .
- Usually done by **developers**, in development environments.

2- Integration testing:

In this testing, two or more modules which are unit tested are integrated to test, It has 2 types of tests

- **Validity** testing – the new module responds to what has been requested when integrated into the overall system
- **Non-regression** tests – The integration of the new module does not induce functional regressions (modification of the behavior of existing modules) or non-functional regressions (*instability*) of the system as a whole

Target: Components and overall system.

Performed by the developer.

3- System testing:

Ensure that the application has the expected features,

- It consists of testing the system at its future operating site under operational conditions and beyond (overload, hardware failure, etc.).
- **Target:** the overall system on the client's IT architecture
- Performed by the group of testers.

3- System testing:

For the purpose of testing the different quality characteristics, there are several types of system testing among them:

- **Functionality testing: consists of** testing whether the system performs the functions for which it was designed.
- **Volume testing: consists of** testing whether the system is capable of handling a large volume of data specified in the requirements.

3- System testing:

- **Usage tests (or usability tests):** This type of tests focuses on the human factor. This will involve selecting a representative subset of potential users to test the system.
- **Security testing: consists of** testing the security properties that the system must ensure such as the availability, integrity, confidentiality of data and services. The purpose of these tests is to reveal security vulnerabilities.
- **Performance tests: consist of** testing the performance of the system by illustrating its failures in terms of response time or processing capacity under certain conditions.

3- System testing:

- **Robustness tests:** Aims to validate the fact that the system reacts correctly to non-compliant use (for example to the entry of invalid data).
- **Configuration tests: consist of** testing the system in order to verify whether it works in the different configurations (networks, operating systems, etc.) for which it was designed.
- **Reliability testing: consists of** testing the reliability properties of the system specified in the requirements.

4. Acceptance testing:

This is the last step before system acceptance.

- The system is tested under the conditions defined by the future user, rather than by the developer.
- Acceptance testing is done by the customers to check whether the delivered products perform the desired tasks or not
- This testing helps the project team to know the further requirements of the users directly
- It brings confidence and satisfaction to the clients as they are directly involved in the testing process. It is easier for the user to describe their requirement.

II- Classification of test methods

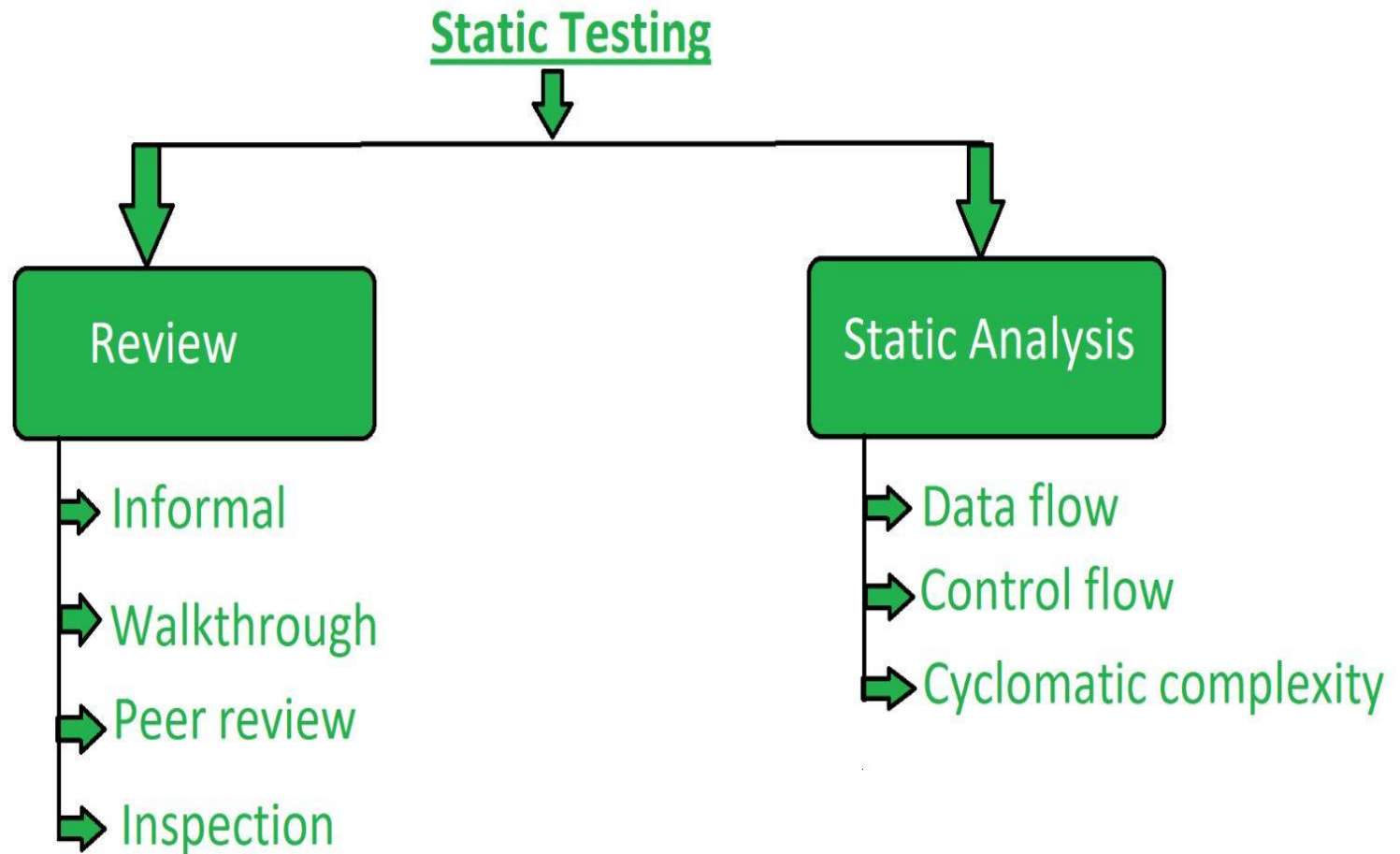
Classification of methods

- There are different classes of test methods depending on whether the program to be validated (under test) is executed during the test or not. Existing test methods are divided into two categories: **static tests** and **dynamic tests**.
- Static test methods are used to examine the software **without running** it; dynamic test methods require generation of test cases for **running the** software

A. Static test methods

- Static test methods consist of the textual analysis of the work product in order to detect errors, without the execution of the program. The ideal, during this test, is to bring together four speakers:
 - 1. A Moderator** (project managers)
 - 2. The Programmer,**
 - 3. The Designer**
 - 4. and an Inspector** (quality manager)

A. Static test methods



1. Review

- In static testing, the review is a process or technique that is performed to find potential defects in the design of the software. It is a process to detect and remove errors and defects in the different supporting documents like software requirements specifications. People examine the documents and sorted out errors, redundancies, and ambiguities. Review is of four types:

1. Review

- **Informal:** In an informal review the creator of the documents put the contents in front of an audience and everyone gives their opinion and thus defects are identified in the early stage.
- **Walkthrough:** It is basically performed by an experienced person or expert to check the defects so that there might not be problems further in the development or testing phase

1. Review

- **Peer review:** Peer review means checking documents of one another to detect and fix defects. It is basically done in a team of colleagues.
- **Inspection:** Inspection is basically the verification of documents by the higher authority like the verification of software requirement specifications

2. Static Analysis

- ❖ Static Analysis includes the evaluation of the code quality that is written by developers. Different tools are used to do the analysis of the code and comparison of the same with the standard. It also helps in following identification of the following defects:
 - Unused variables.
 - Dead code.
 - Infinite loops.
 - Variable with an undefined value.
 - Wrong syntax.

2. Static Analysis

- ❖ Static Analysis is of three types:
 - **Data Flow:** Data flow is related to the stream processing.
 - **Control Flow:** Control flow is basically how the statements or instructions are executed.
 - **Cyclomatic Complexity:** Cyclomatic complexity defines the number of independent paths in the control flow graph made from the code or flowchart so that a minimum number of test cases can be designed for each independent path.

B. Dynamic test methods

- Dynamic test methods consist of executing the program to be validated (under test) using a test set. They aim to detect errors by comparing the results obtained by the execution of the program with those expected by the specification of the application.
- Dynamic testing involves performing the following four **steps**:
 - **Selection of test sets:** This step consists of finding input data, scenarios to run the test as well as the expected result.

B. Dynamic test methods

- **Submission of the test set:** In order to execute the tests selected in the previous step, a test program is executed with the test data.
- This program can be the software that we want to test in the case of system testing or acceptance testing. For unit testing and integration testing, the test program is a test driver that must execute the part of the software that is to be tested.

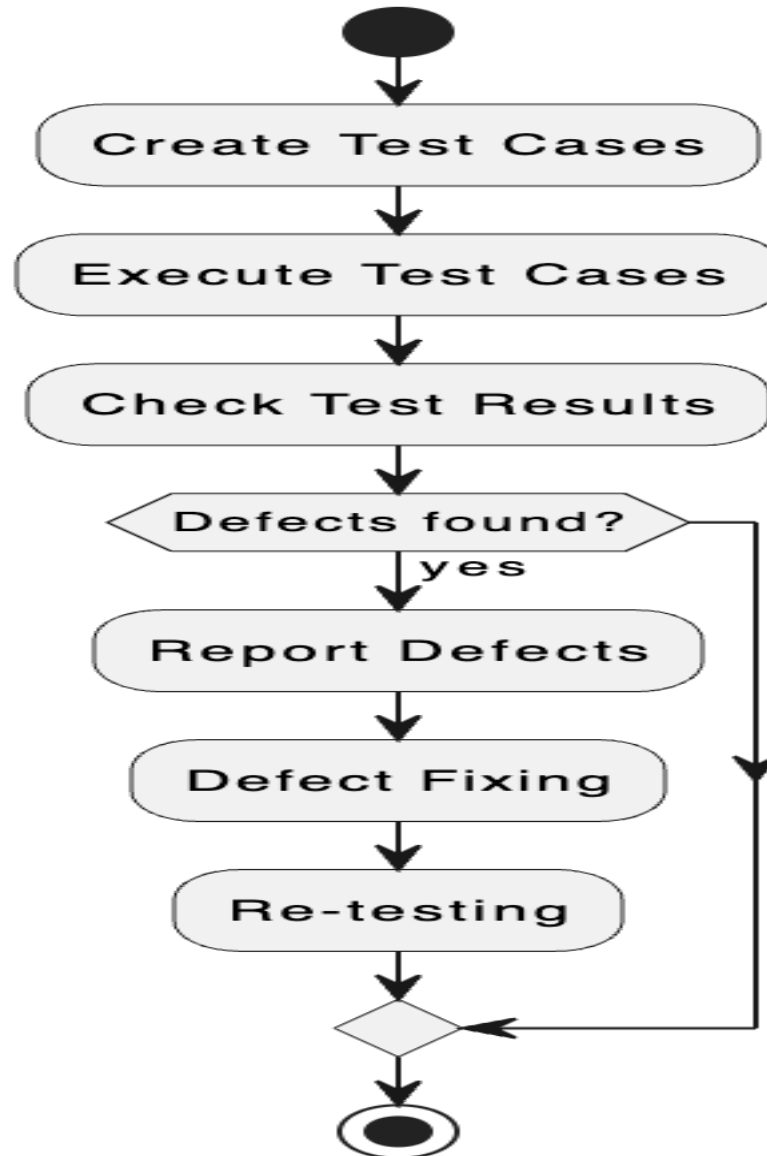
B. Dynamic test methods

- **Analysis of the results:** It consists of deciding on the success or failure of the test set. The problem then arises of the oracle because the decision is not always simple, or even possible. An oracle is a function that must make it possible to distinguish whether an output of the software is correct.
- If the oracle has detected a fault, it must be corrected and the same test case must be re-executed to verify that the correction has effectively eliminated the fault. If the oracle has not detected a failure, either the stop criterion is met and the software test phase is completed, or a new test case will be considered.

B. Dynamic test methods

- **Evaluation of the quality of the tests carried out (stopping criteria):** An exhaustive test can rarely be carried out. In most cases, you have to settle for a less expensive test and therefore you have to have a criterion to decide when the test is satisfactory.

Dynamic Testing Steps

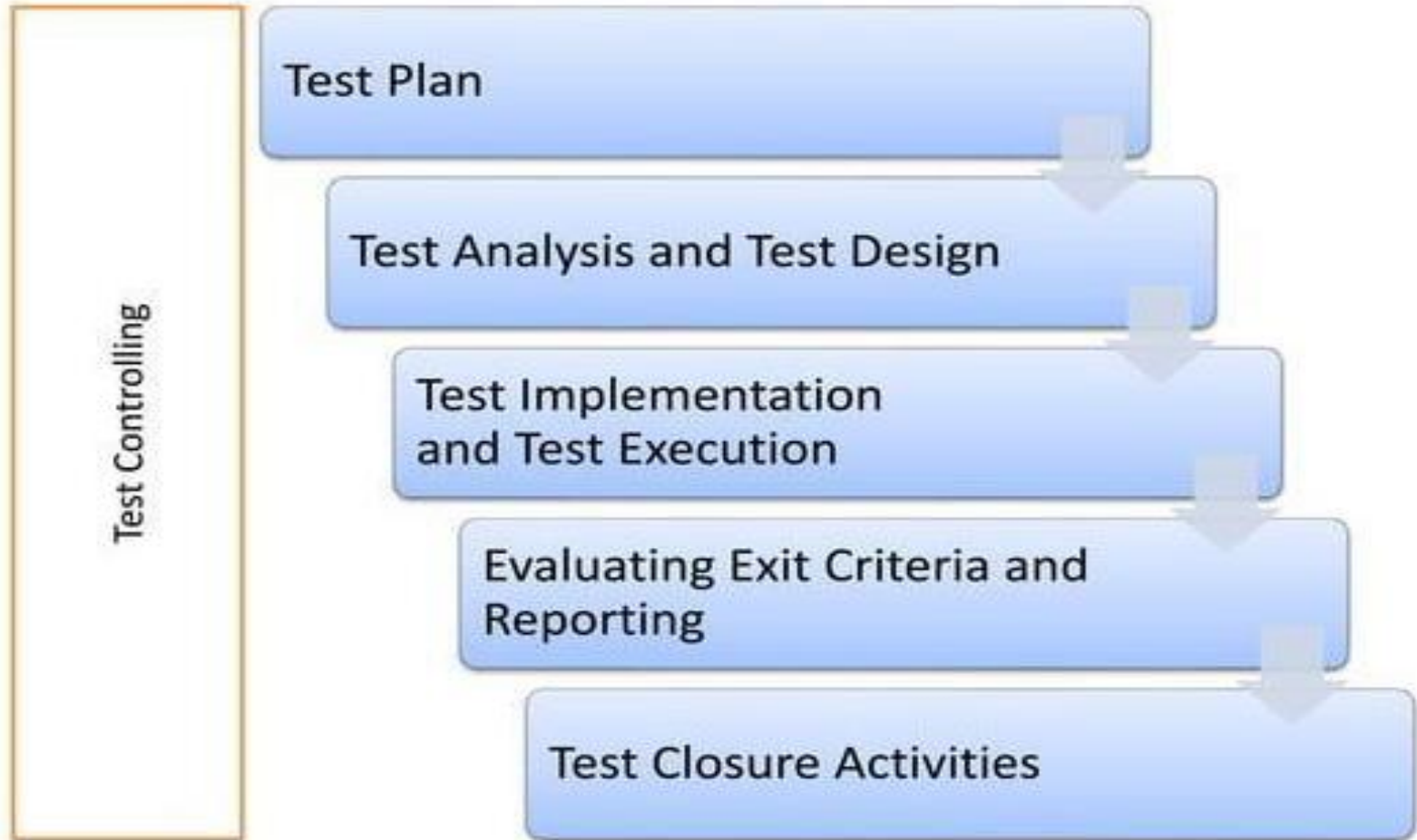


III. Fundamental testing process

Testing process

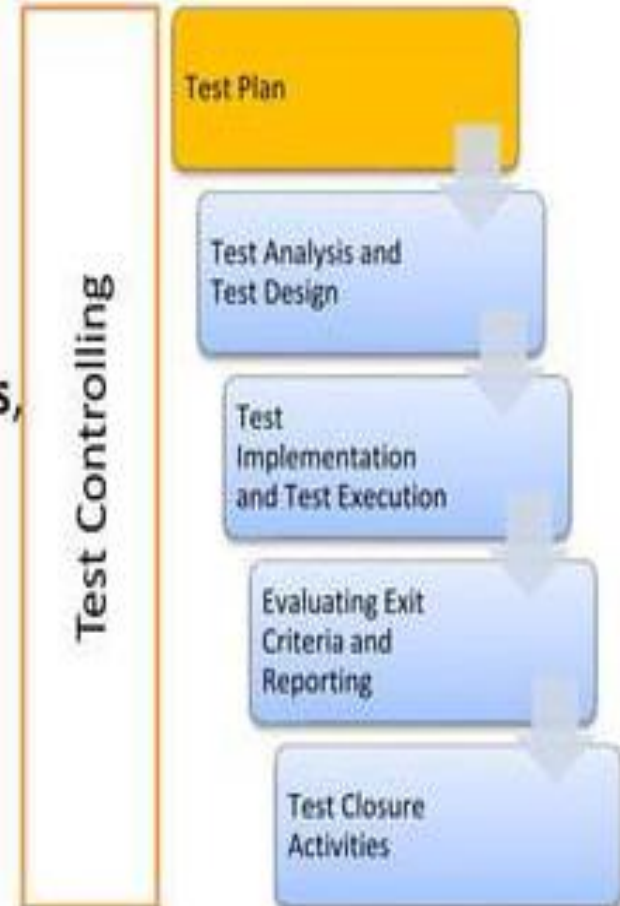
- ❖ Testing is a process itself
- ❖ The testing process is determined by the following phases
 - Test planning
 - Test analysis and test design
 - Test implementation and test execution
 - Evaluating Exit Criteria and reporting.
 - Test closure activities
 - Test Controlling (at all phases)

Testing process



1. Test planning

- Determining the **scope** and **risk**
- Identifying the objectives of testing and **exit criteria**
- Determining the approach: **test techniques**, test **coverage**, testing **Teams**
- Implement testing **method/test strategy**, plan **time** span for activities following
- Acquiring and scheduling test resources: **people**, test **environment**, test **budget**

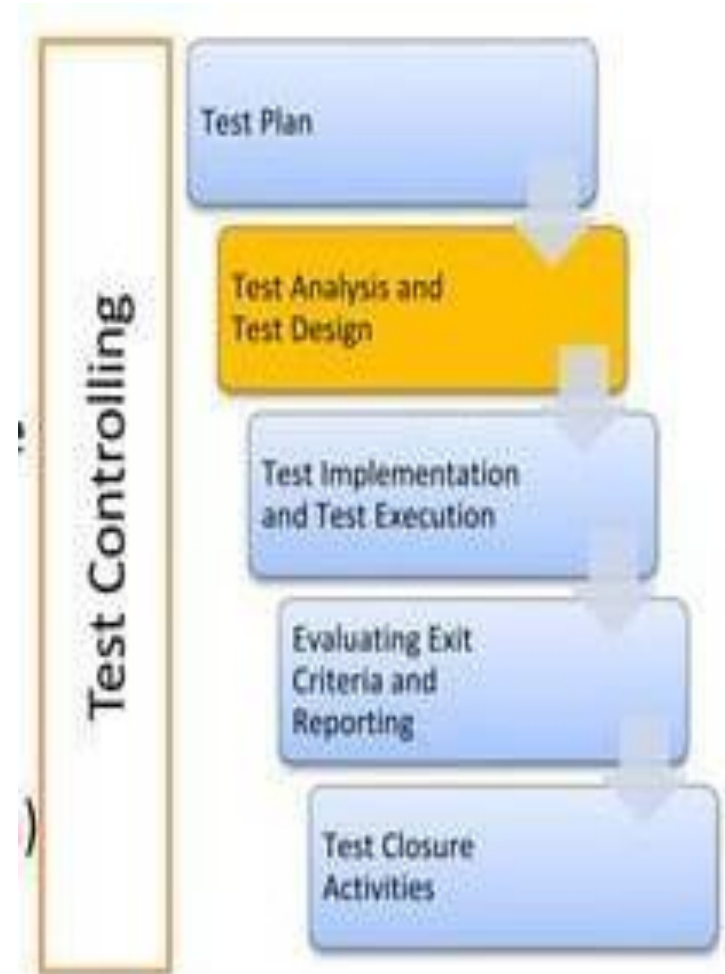


2. Test Analysis and Design

Test analysis and design represent activities where general test objectives are transformed into tangible test conditions and test designs.

This step consist of the following major tasks:

- ✓ Review the basics of the test (such as requirements, software integrity level (risk level), risk analysis reports, architecture, design and interfaces)
- ✓ Evaluate the testability of the requirements and the system



3. Test implementation and test execution

- ✓ creating test data , writing test procedure and creating test sequences
- ✓ creating test automation scripts if necessary.
- ✓ configuring the test environment
- ✓ executing test
- ✓ test result recording and analysis
- ✓ **Retest** (after defect correction)
- ✓ **Regression test** : ensure that changes (after installing a new release, or error fixing) did not uncover other or introduce new defects.



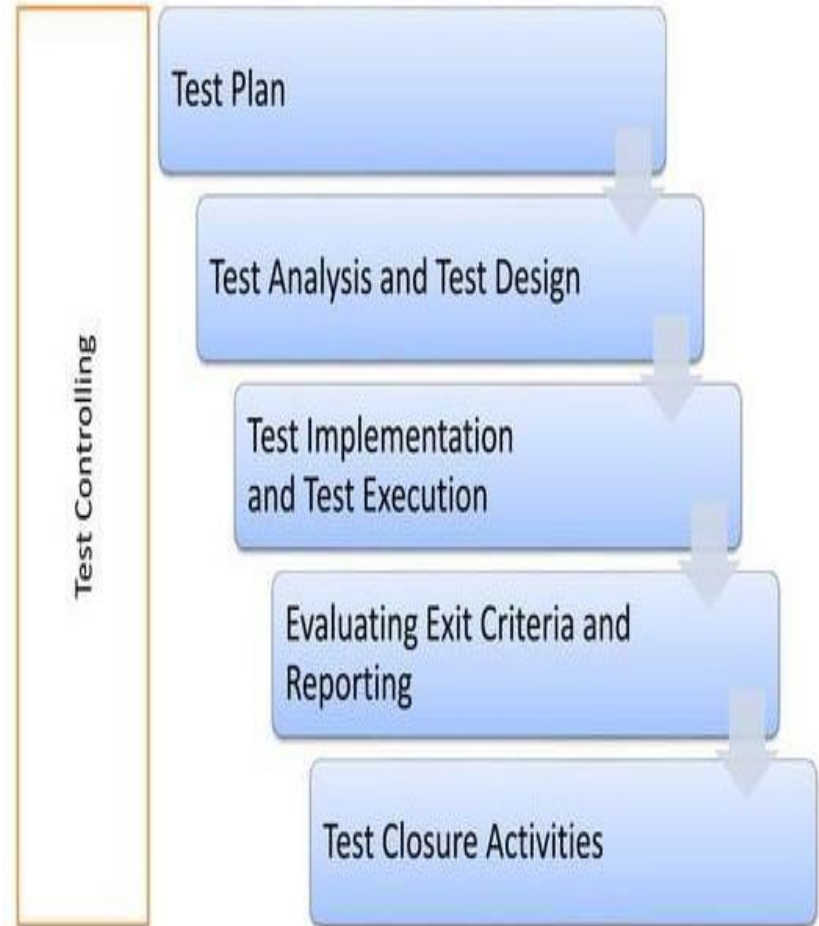
4. Evaluating Exit Criteria and reporting.

- ✓ Assessing test execution against the defined objectives (e.g. test and criteria)
- ✓ Evaluating **test logs** (summary Of test activities, test result, communicate exit criteria)
- ✓ Provide information to allow the decision, whether **more test** should take place



5. Test Closing Activities

- ✓ Collect data from completed test activities to consolidate experience.
- ✓ Closure of incident reports
- ✓ Documenting the acceptance of the system
- ✓ Finalizing and archiving test ware, for later reuse, hand over to operations.
- ✓ Analyzing "lessons learned" for future project



6. Test Controlling

This is the activity of comparing actual progress against the plan, and reporting the status, including deviations from the plan. It involves taking actions necessary to meet the mission and objectives of the project



Conclusion

- Testing is a creative activity:
 - Imagine possible scenarios that could cause software to fail
 - And build test banks to verify functionalities and constraints.