NAME : G.NITHISH

REG. No. : 19BCS0012

COURSE TITLE : Software Quality Assurance / Testing - TH

COURSE CODE : CSC4001

FACULTY : Prof. UMA K

1.) Software Testing Principles:

A) Software testing is a Procedure of implementing Software or the application to identify the defects or bugs. For testing an application or software, we need to follow some Prinicples to make our Product defects free, and that also help the test engineers to test the software with their effort and time. Here in this section, we are going to learn about the seven essential Priniciples

Let us see the seven different testing Prinicples:-

- Testing shows the Presence of defects
- Exhaustive Testing is not possible
- Early Testing.
- Defect Clustering
- Pesticide Paradox
- Testing is Context-dependent
- Absence of errors fallacy

Principle 1:- Testing shows the Presence of defects.

- The test engineer will test the application to make sure that the application is bug or befects free.

- While doing testing we can only identify that the application or software has any errors. The Primary purpose of doing testing is identify the numbers of unknown bug with the help of Various methods and testing techniques because the entire test Should be traceable to the customer requirement. Which means that to find any defects that might cause the Product failure to meet the client's needs.

- By doing testing on any application, we can decrease the number of bugs, which doesn't mean that the application is defect-free because sometimes the software seems to be bug-free while performing multiple types of testing on it, But at the time of deployment in the production Server, if the end-user encounters those bugs which are not found in the testing Process.

Principle 2:- Exhaustive testing is impossible.

- Sometimes it seems to be very hard to test all the modules and their features with effective and non-effective combinations of the inputs data throughout the actual testing process.

- If you were to test all the possible combination Projects execution time and Costs will rise exponentially.

- Hence, instead of performing the exhaustive testing baundless determinations and most of the hard work is unsuccessful. So we can complete this type of variations according to the importance of modules because the product timelines. Will not permit us to perform such type of testing scenarios.

Principle 3:- Early Testing:-

- Here early testing means that all the testing activities should start in the early stages of the software development life cycle's requirement's analysis stage to identify the defects because if we find the bugs

At an early stage, it will be fixed in the inital stage itselfen, which may cost us very less as compared to those which are identified in the future phase of testing Process

*) To Perform testing we will require the requirement Specification documents, therefore, if the requirements are defined incorrectly, then it can be fixed directly rather than fixing them in another stage, which could be the development Phase.

Principles 4:- Defect clustering:-

*) Defects are not evenly distributed in a system.

*) They are clustered.

*) The defect clustering defined that throughout the testing Process, we can detect the number of bugs which are correlated to a small number of modules. We have Various reasons for this such as the modules could be complicated; the coding Part may be complex, and so on.

★ These type of software or the application will follow the Pareto Principle which states that we can identify that approx. Eighty percent of the complication is present in 20 percent of the modules. with the help of this, we can find the uncertain modules, but this method has its difficulties if the same test are performing reguarly, hence the same test will not able to identify the new defects.

Principles 5:- Pesticide paradox

★ This Principle defined that if we are executing the same set of test cases again and again over a particular time, then these kinds of the test will not be able to find the new bugs in the software or the application,

★ To get over these pesticide paradoxes. it is very significant to review all the test cases frequently. And the new and different tests are necessary to be written for the implementation of multiple parts of the application or the software which helps us to find more bugs.

Principle 6:- Test is context-dependent:-

- Testing is a context-dependent principle States that we have multiple fields such as e-commerce websites, commercial websites, and so on are available in the market.
- There is a definite way to test the commercial site as well as the e-commerce website because every application has its own needs, features and functionality.
- To check this type of application we will take the help of various kinds of testing. different technique, appraaches and multiple methods.

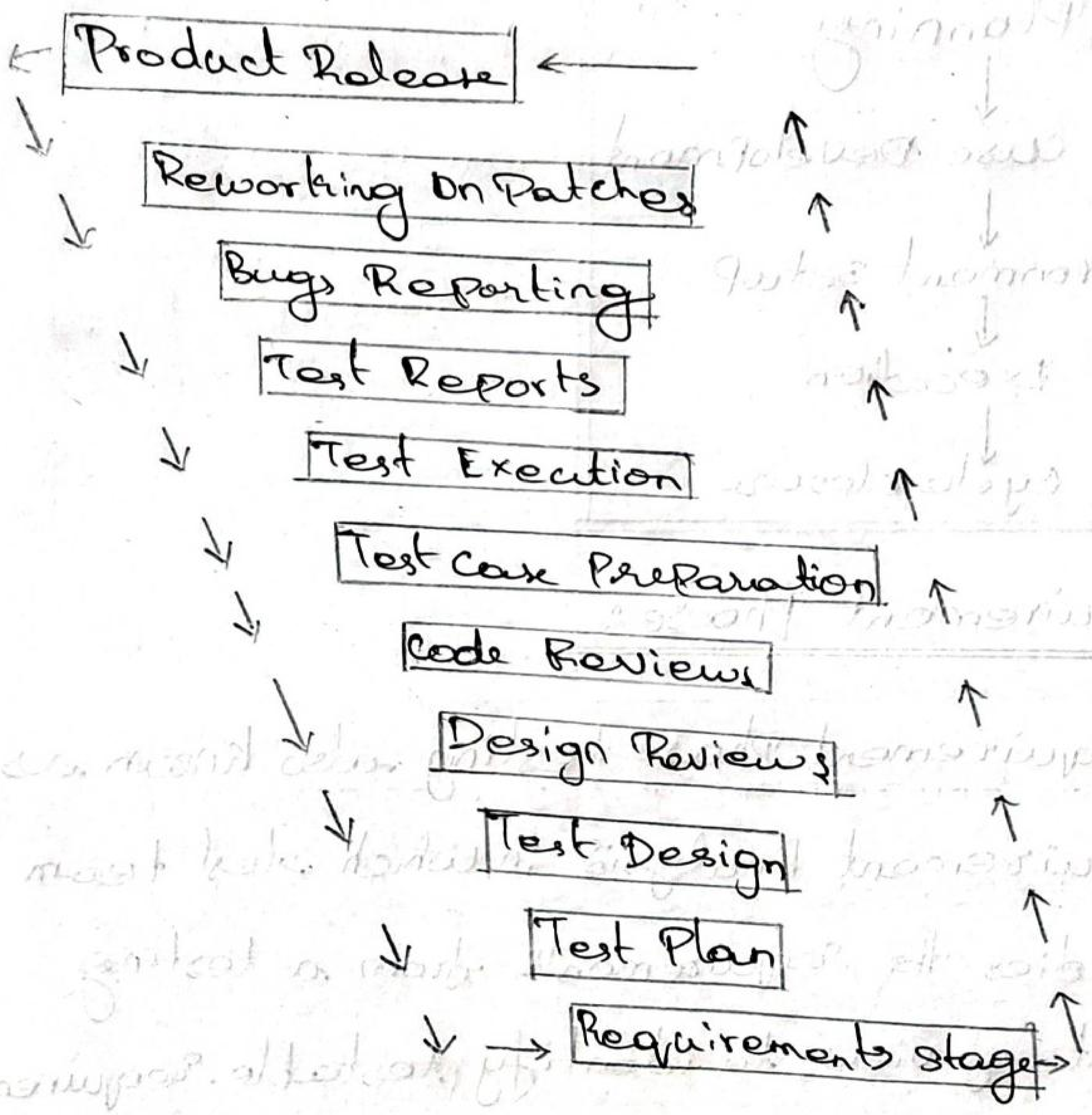
Principle 7:- Absence of error fallacy:-

- Once the application is completely testest and there no bugs identified before the release, so we can say that the application is 99 percentage bug-free, But there is the chance when the application is tested beside the incorrect requirment, identified the flaws fixed them on a given period would not help as testing is done on the wrong specification which doesn't apply to the client's requirements.

2) Software Testing Life cycle models:-

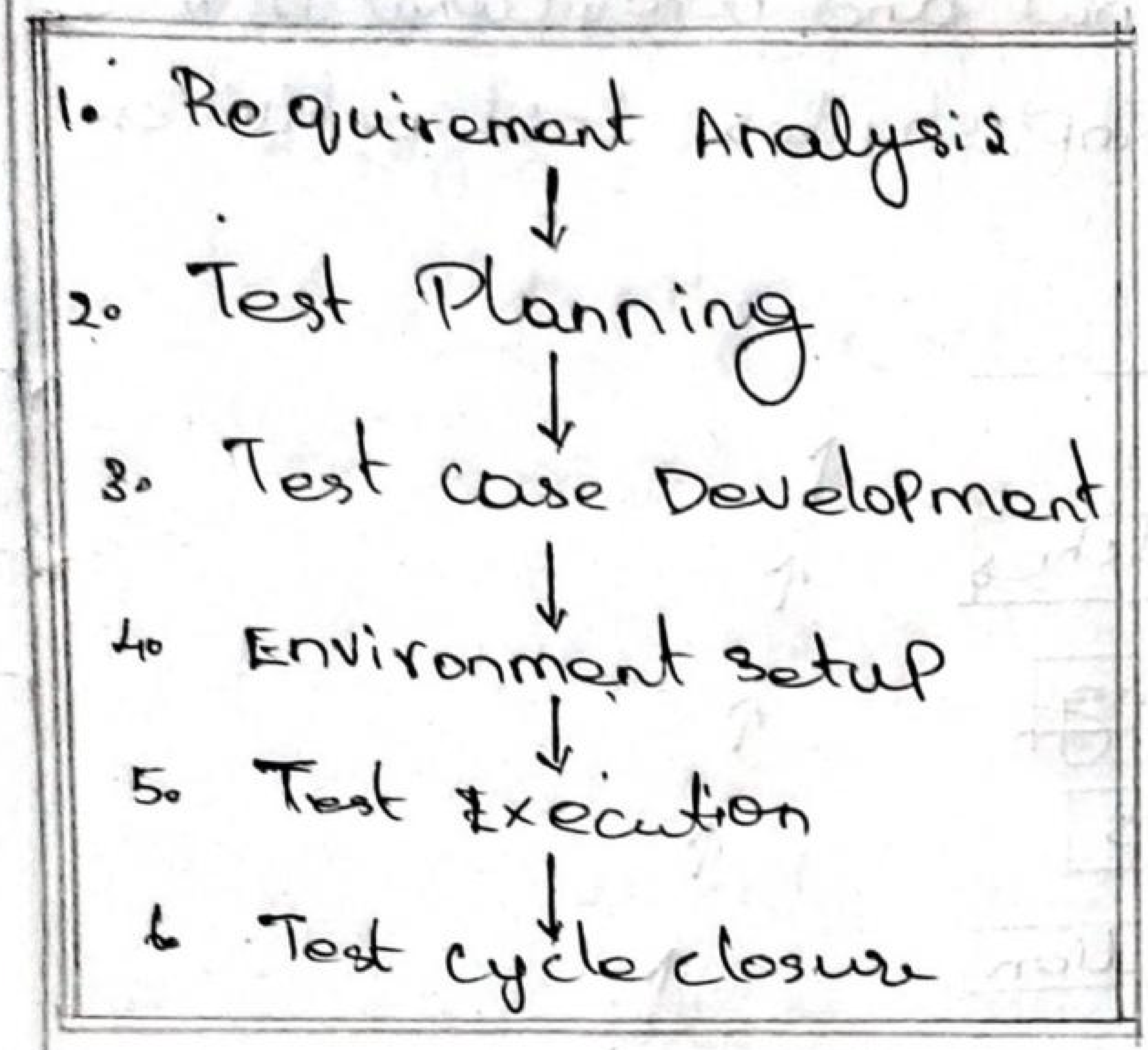
STLC:

Software Testing life cycle identifies what test activites to carry out and when (what is the best time) to accomplish those test activities.



STLC Phases:-

Six major Phase in every Software Testing Life cycle Model



1. Requirement Analysis
2. Test Planning
3. Test case Development
4. Environment setup
5. Test Execution
6. Test cycle closure

1.) Requirement Phase:

Requirement Phase testing also known as Requirement Analysis in which test team Studies the requirements from a testing Point of view to identify testable requirement and the QA team may interact with Various stakeholders to Understand ; requirments in detail. Requirements Could be either functional or non-functional Automation feasibility for the testing Project is also done in this stag.

Activities in Requirement Phase Testing:-

*) Identify types of tests to be Performed.

*) Grather details about testing Priorites

*) Prepare Requirment Traceability Matrix

*) Identify test environment details whein testing is supposed to be carried out.

Deliverables of Requirement Phase.

•) RTM

•) Automation feasibility report

2.) Test Planning:-

*) Test Planning in STLC is a phase in which a senior QA manager determines the test Plan strategy along with efforts and cost estimates for the Project. More over, the resources, test envioonment, test limitations & the testing schedule are also determined.

Deliverables of Test Planning:-

*) Test Plan/strategy document

*) Effort estimation document.

3.) Test Case Development phase:

*) the test case Development Phase involves the Creation, Verification and rework of test cases & test scripts after the test plan is ready. Initially, the test data is identified then created & reviewed and then reworked based on preconditions.

Deliverables of Test case Development Phase:-

*) Test cases/scripts
*) Test data.

4.) Test Environment Setup:-

*) Test environment setup decides the software and hardware conditions under which a work product is tested. It is one of the Critical aspects of the testing process and can be done in Parallel with the Test case Development phase.

*) Test team may not be involved in this activity if the development teams provides the test environment.

Test Execution Phase:-

- Test Execution Phase is carried out by the tester in which testing of the software build is done based on test plans and test cases Prepared.

- The Process consists of test script exection, test script maintenance and bug reporting. If bugs are reported then it is reverted back to development team for correction and retesting will be performed.

Test cycle Closure:-

- Test cycle closure Phase is completion of test execution which involves several activities like test completion reporting, Collection of test completion matrices and test results.

- Testing team member meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from current test cycle. The idea is to remove process bottlenecks for future test cycles.