Module: Capstone Project using Java
Qualification: Advanced Certificate In Web Development

LITHAN

IU 1. Introduction to Testing

By the end of this session, you will be able to get an Introduction to Testing

Contents

LITHAN

S. No.	Topic Description	Required / Optional
01	Levels of Testing	Required
02	Test Metrics	Required

Types of Tests in Software Testing

LITHAN

- Each type of testing has its own features, advantages, and disadvantages as well.
- Given below is the list of some common types of Software Testing:

Functional testing types	Non-functional testing types
▪ Unit testing ▪ Integration testing ▪ System testing ▪ Sanity testing ▪ Smoke testing ▪ Interface testing ▪ Regression testing ▪ Beta/Acceptance testing	▪ Performance Testing ▪ Load testing ▪ Stress testing ▪ Volume testing ▪ Security testing ▪ Compatibility testing ▪ Install testing ▪ Recovery testing ▪ Reliability testing ▪ Usability testing ▪ Compliance testing ▪ Localization testing

Levels of Testing in Software Testing

LITHAN

- Level of software testing
  - A level of software testing is a process where every unit or component of a software/system is tested.
  - There are many different testing levels which help to check behavior and performance for software testing.
  - These testing levels are designed to recognize missing areas, reconciliation between the development lifecycle states.

- Software Development Life Cycle (SDLC) models
  - In SDLC models there are characterized phases such as requirement gathering, analysis, design, coding or execution, testing, and deployment.
  - All these phases go through the process of software testing levels

Levels of Testing in Software Testing

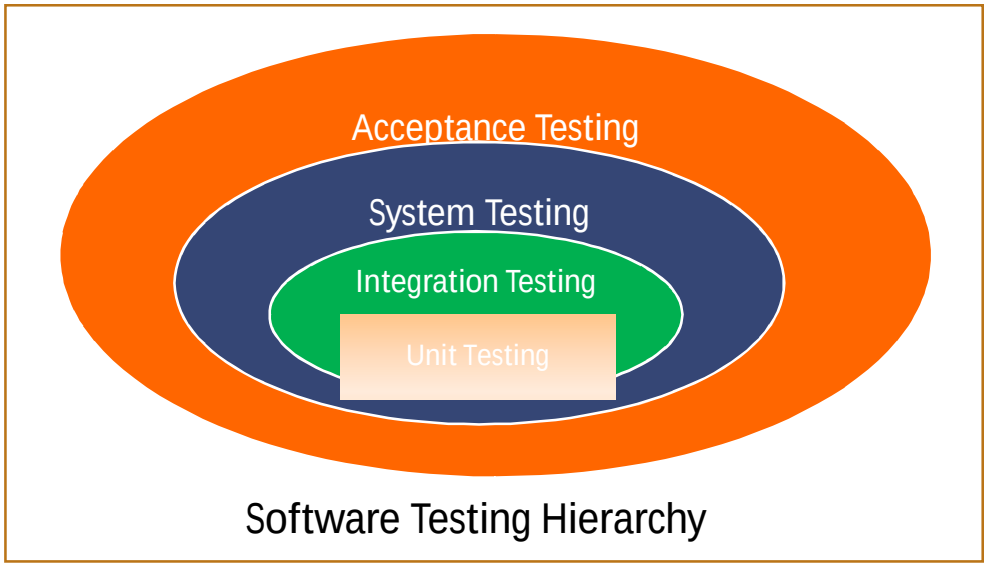
LITHAN

- Unit Testing (Module testing)
  - Debugging or tracing.
- Interface Testing
  - Interfaces between system components.
- Integration Testing
  - Communication between modules.
- Stress Testing
  - Stability and reliability of the system.
- Load Testing
  - Systems performance under real-life load conditions.
- System Testing.
  - Behavior of a system as per software requirement specification
- Acceptance Testing.
  - Verification Testing, Validation testing, Audit Testing

Levels of Testing - Software Testing Hierarchy

LITHAN

- Unit testing
  - Each Module/Component is tested.
- Integration testing
  - Testing Interconnectivity between the different components in a system.
- System testing
  - Testing of a complete and fully integrated software product.
- Acceptance testing
  - Validate the end to end business flow.



Software Testing Hierarchy

Unit Testing

LITHAN

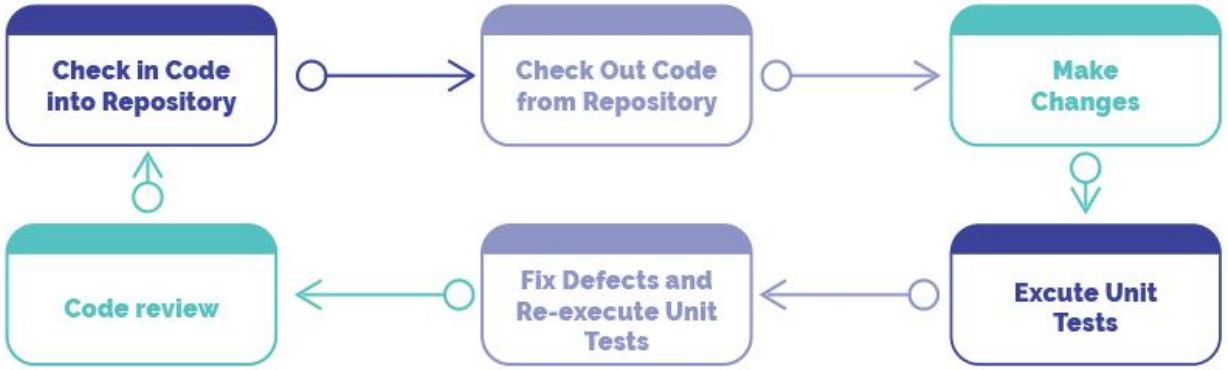
- Each Module/Component is tested in the system or software
- Unit testing is usually done by developers instead of testers
- Developers execute the build intentionally to find the defects
- Concerned with functional correctness and completeness of individual program units
- Written and run by software developers to meet designs
- Isolates each part of the program showing individual parts corrected.
- All components are tested at least once.
- In unit testing, a function which is in the form of a section of code is tested to verify its accuracy using drivers, unit testing frameworks, mock objects, and stubs
- Scope is smaller, easier to fix errors.

How to Perform Unit Testing ?

LITHAN

- Unit testing is usually automated, but sometimes it can be done manually.
- In automated unit testing, a developer writes code in the app to test the function or procedure.
- Once the app is deployed, that code can be removed.
- The function can be isolated to test the app rigorously and it reveals the dependencies between the code being tested and other units. Then the dependencies can be eliminated.
- Most developers use unit test automated framework to log the failing test cases.

UNIT TEST LIFE CYCLE



Advantages and Disadvantages of Unit Testing

LITHAN

- Advantages
  - With unit testing, the speed of development will be faster. If you perform developer tests instead of unit tests, then you need to set breakpoints, fire up the GUI, and provide inputs. But if you do unit test, you write the code, write the test, and then run the test. You don't need to provide the inputs or fire up the GUI
  - In the end, you have a more reliable code. It takes less time to find and fix the bugs during unit testing than in system or Acceptance testing
  - Unit tests are easier to debug as only the latest changes need to be debugged if the test fails, whereas if you test on a higher level, then you will have to scan the changes made within weeks or months
  - In addition, we can test a part of the project without waiting for others to be completed due to the modular nature of the unit test

- Disadvantage
  - The only disadvantage with unit testing is that it's not possible to check all the execution paths and it cannot catch broad system errors or integration errors

Interface Testing

LITHAN

- Interfaces between system components are tested.

- Incorrect mapping of data causes bugs :
  - Misinterpretation of the information due to Data inconsistency.
  - Software that interfaces between the two systems fails.
  - No data is transferred resulting entire interface failing.

- Performed in two phases:
  - When the interfaces are tested individually during system testing.
    - using a “dummy” system or stub to mimic the closed-loop system.
  - When the two systems are tested together with the systems communicating with one another during integration testing.

Why do Interface Testing?

LITHAN

- To ensure that end-users or customer should not encounter any problem when using a particular software product.
- To identify which application areas are usually accessed by end-users and to check its user-friendliness as well.
- To verify security requirements while communication propagates between the systems.
- To check if a solution is capable to handle network failures between an application server and website.

Integration Testing

LITHAN

- Integration testing is a level of software testing where individual units are combined and tested as a group.
- Interconnectivity between the different components in a system.
- The purpose of this level of testing is to expose faults in the interaction between integrated units.
- Test drivers and test stubs are used to assist in Integration Testing.
- Focuses on checking da,ta communication between modules.
- Testing is performed by developers.
- Integration Testing strategies:
  - Incremental Integration
    - Top-down
    - Bottom-Up
  - Big Bang
  - Sandwich/Hybrid
- Incremental Integration testing reduces the difficulty is localizing errors.

Integration Testing

LITHAN

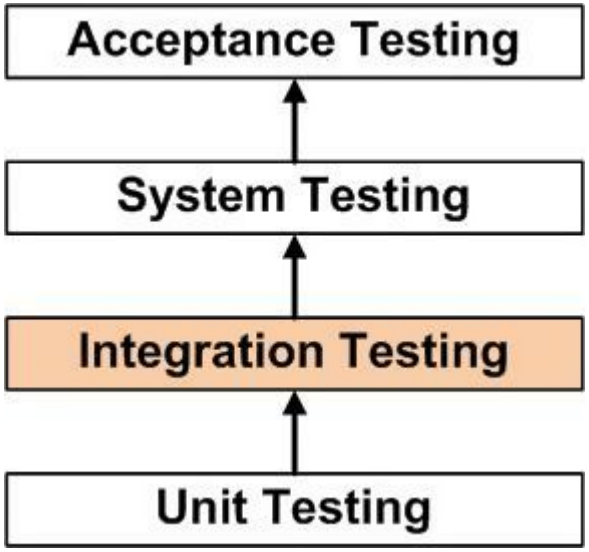
- Entry Criteria:
  - Unit Tested Components/Modules
  - High prioritized bugs are fixed and closed
  - Modules to be coded are completed and integrated successfully.
  - Integration tests Plan, test case, scenarios to be signed off and documented.
  - Required Test Environment to be set up for Integration testing.

- Exit Criteria:
  - Successful Testing of Integrated Application.
  - Executed Test Cases are documented.
  - All High prioritized bugs are fixed and closed.

Methods of Integration Testing

LITHAN

- Any of Black Box Testing, White Box Testing and Gray Box Testing methods can be used.
- Normally, the method depends on your definition of ‘unit’.
- When is Integration Testing performed?
  - Integration Testing is the second level of testing performed after Unit Testing and before System Testing.



- Who performs Integration Testing?
  - Developers themselves or independent testers perform Integration Testing.

Stress Testing

LITHAN

- Used to test the stability & reliability of the system.
- Determines the system robustness and error handling in heavy load conditions..
- Avoids system crash during extreme situations.
- Determines the limit where the system software or hardware breaks.
- Provides effective error management under extreme conditions.
- Goals of Stress Testing:
  - Checks if the system works under abnormal conditions.
  - Analyze the behavior of the system after failure.
  - Displaying appropriate error message when system is under stress.
  - Ensure that system recovers after failure(recoverability).

Load Testing

LITHAN

- Determines a systems performance under real-life load conditions.
- Provides application behavior when multiple users access it simultaneously.
- Load testing identifies:
  - Maximum operating capacity of an application.
  - Determines if current infrastructure is sufficient to run the application.
  - Sustainability of application with respect to peak user load.
  - Number of users an application supports.
  - Scalability to allow more users to access.
- Used for the Client/Server, Web based applications for Intranet and Internet.

System Testing

LITHAN

- Testing of a complete and fully integrated software product.
- Performed after system performs Integration test.
- Enables to test, verify and validate the business requirements and applications.
- Involves the external workings of the software from the user's perspective.
- Application is tested to verify the technical and functional specifications.
- System Testing involves:
- Testing the fully integrated applications including external peripherals.
- Testing of every input to check for desired outputs.
- Testing of the user's experience with the application.

System Testing

LITHAN

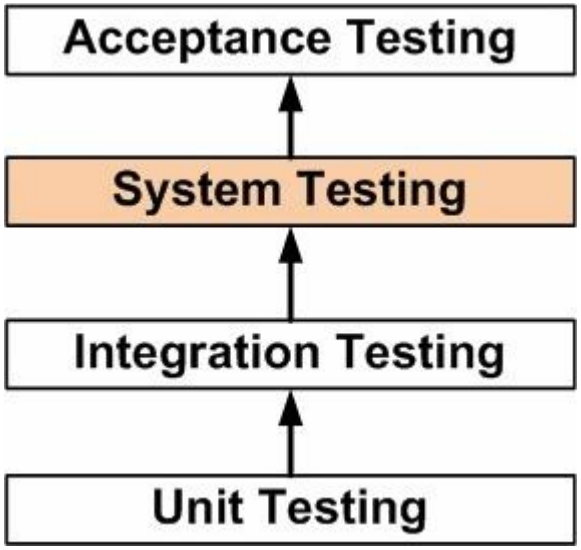
- Entry Criteria for System testing:
  - Complete software system should be developed.
  - Unit testing is performed.
  - Integration testing must be completed.
  - Specification for the product is completed.
  - Test Scripts are ready.
- Exit Criteria for System testing:
  - Application meets all requirements and functionalities.
  - Defects found during testing are fixed and closed.
  - All the test cases of system are executed.
  - No critical defects are opened.

Methods of System Testing

LITHAN

- Usually, Black Box Testing method is used.

- When is it performed?
  - System Testing is the third level of software testing performed after Integration Testing and before Acceptance Testing.



- Who performs it?
  - Normally, independent Testers perform System Testing.

System Testing-Types

LITHAN

- Usability testing
  - Focuses on the users ease of use of the application.
  - Flexibility in handling controls.
  - Ability of the system to meet its objectives.
- Regression testing
  - Retesting previously tested components to ensure proper functionality after modification in a part of system.
- Functional testing
  - Finds any possible missing functions.
- Recovery Testing
  - Demonstrates a software solution is reliable and trustful.
- Stress Testing
  - Tests the stability & reliability of the system.
- Load testing
  - Determines systems performance under real-life load conditions.

Alpha Testing

LITHAN

- It is the most common type of testing used in the Software industry.
- The objective of this testing is to identify all possible issues or defects before releasing it into the market or to the user.
- Alpha testing is carried out at the end of the software development phase but before the Beta Testing.
- Still, minor design changes may be made as a result of such testing.
- Alpha testing is conducted at the developer's site. In-house virtual user environment can be created for this type of testing.

Beta Testing

LITHAN

- Beta Testing is a formal type of software testing which is carried out by the customer.
- It is performed in **the Real Environment** before releasing the product to the market for the actual end users.
- Beta testing is carried out to ensure that there are no major failures in the software or product and it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software.
- Usually, this testing is typically done by end-users or others. It is the final testing done before releasing an application for commercial purpose. Usually, the Beta version of the software or product released is limited to a certain number of users in a specific area.
- So end user actually uses the software and shares the feedback to the company. Company then takes necessary action before releasing the software to the worldwide.

Acceptance Testing

LITHAN

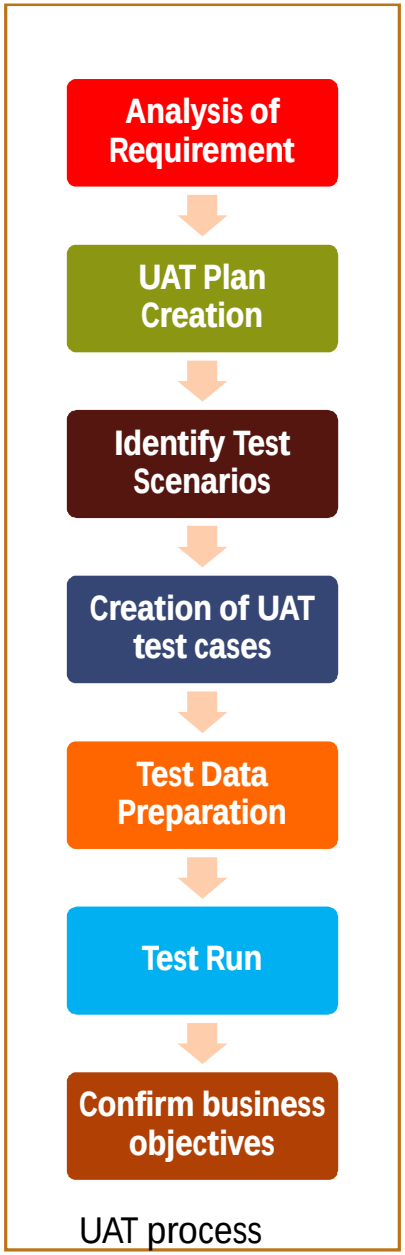
- Validate the end to end business flow.
- Last phase of the Software testing process.
- Performed by the Client to certify the system with respect to the requirements.
- Prerequisites of User Acceptance Testing:
  - Business Requirements are available.
  - Application Code is fully developed.
  - Unit Testing, Integration Testing & System Testing should be completed.
  - No defects in System Integration Test Phase.
  - Only Cosmetic error are acceptable before UAT.
  - Reported defects should be fixed and tested before UAT.
  - Traceability matrix for all testing is completed.
  - UAT Environment must be ready

Acceptance Testing

LITHAN

- User Acceptance Testing Process:
  - UAT is performed by the intended users of the system or software.
  - Performed at client location known as BETA testing.
  - Entry criteria for UAT should be satisfied.
  - Figure shows tasks performed by the testers.

- Exit criteria for UAT:
  - No critical defects open.
  - Business process works satisfactorily.
  - UAT Sign off meeting with all stakeholders.



UAT process

What is Acceptance Criteria ?

LITHAN

- UAT is performed by the intended users of the system or software.
- Performed at client location known as BETA testing.
- Validates the end to end business flow.
- Performed by the Client to certify the system with respect to the requirements.
- Entry criteria for UAT should be satisfied.
- Acceptance Criteria for UAT:
  - Business Requirements are available
  - Application Code is fully developed
  - All the Testing should be completed priorly
  - No defects in any Test Phase
  - Reported defects should be fixed and tested before UAT
  - Traceability matrix for all testing is completed
  - UAT Environment must be ready

Who performs the Acceptance testing?

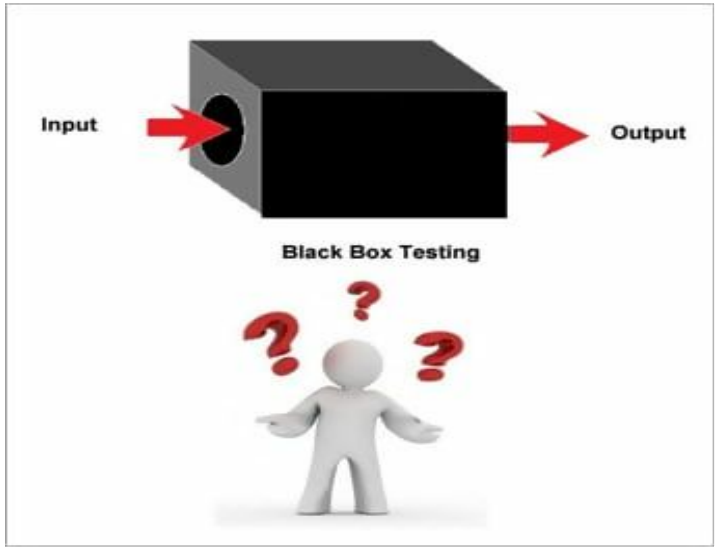
LITHAN

- Internal Acceptance Testing (Also known as Alpha Testing) is performed by members of the organization that developed the software but who are not directly involved in the project (Development or Testing). Usually, it is the members of Product Management, Sales and/or Customer Support.
- External Acceptance Testing is performed by people who are not employees of the organization that developed the software:
  - Customer Acceptance Testing is performed by the customers of the organization that developed the software. They are the ones who asked the organization to develop the software. [This is in the case of the software not being owned by the organization that developed it.]
  - User Acceptance Testing (Also known as Beta Testing) is performed by the end users of the software. They can be the customers themselves or the customers' customers.

Black Box Testing

LITHAN

- Internal system design is not considered in this type of testing. Tests are based on the requirements and functionality.
- Black box testing, which is also known as behavioral, opaque-box, closed-box, specification-based or eye-to-eye testing, is a Software Testing method that analyses the functionality of a software/application without knowing much about the internal structure/design of the item that is being tested and compares the input value with the output value.
- The main focus in black box testing is on the functionality of the system as a whole.



White Box Testing

LITHAN

- White Box testing is based on the knowledge about the internal logic of an application's code.
- It is also known as Glass box Testing. Internal software and code working should be known for performing this type of testing. Under these tests are based on the coverage of code statements, branches, paths, conditions.

Conclusion about test levels and types

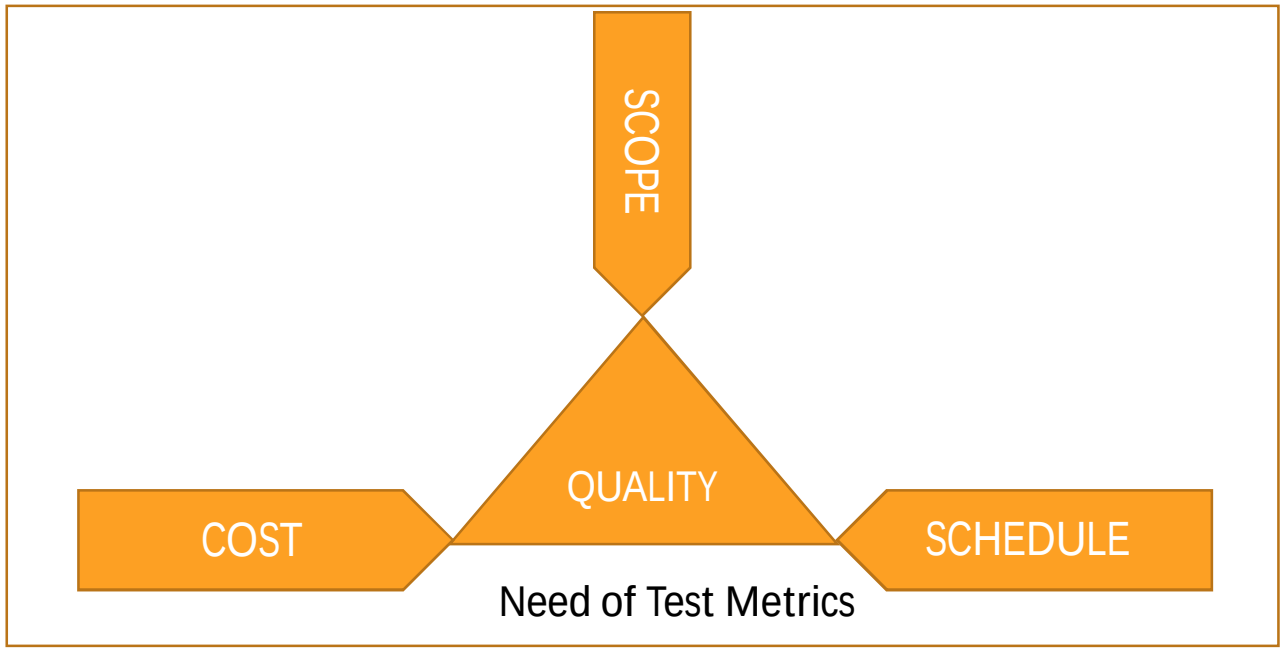
LITHAN

- There are more than 100+ types of testing
- Not all testing types are used in all types of projects.
- Use Testing methods suitable for your organization.

Why we need Test Metrics ?

LITHAN

- It is impossible to specify if the process implemented is improving or not without measurement.
- Improves the efficiency and effectiveness of a software testing process.
- Helps taking decisions for next phase of activities.
- Understands the type of improvement required.
- Makes decisions on process or technology change.
- Helps estimating the progress, quality and health of a software testing effort.

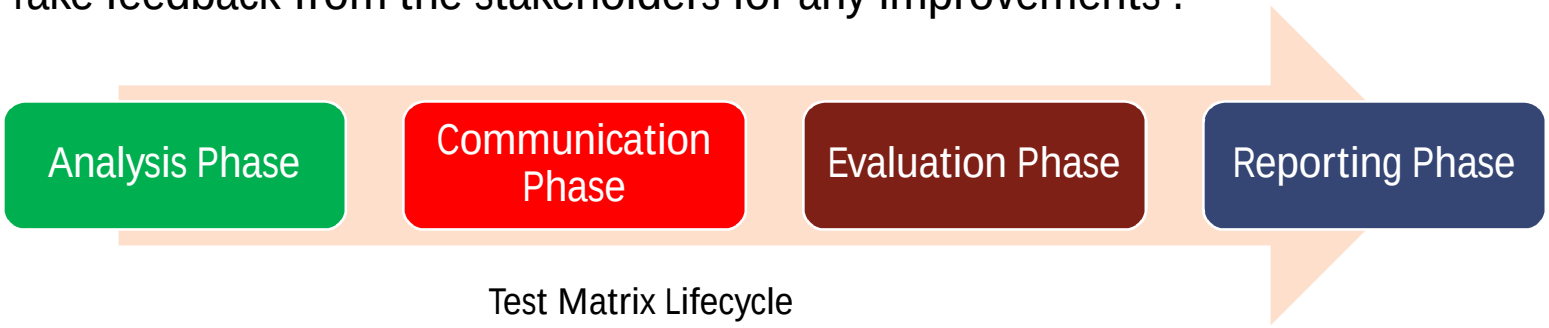


Need of Test Metrics

Testing Metrics Lifecycle

LITHAN

- Analysis Phase:
  - Identify the Metrics which has to be generated .
  - Define the identified Metrics.
- Communication Phase:
  - Explain the need of the Metrics to the stakeholders.
  - Educate the testing team about the data points for generating the Metrics.
- Evaluation Phase:
  - Capture and verify the data used for generating Metrics.
  - Calculate the Metrics based on the data captured.
- Reporting Phase :
  - Develop the Metrics report with effective conclusion.
  - Distribute to the stakeholders .
  - Take feedback from the stakeholders for any improvements .



Test Matrix Lifecycle

Types of Metrics

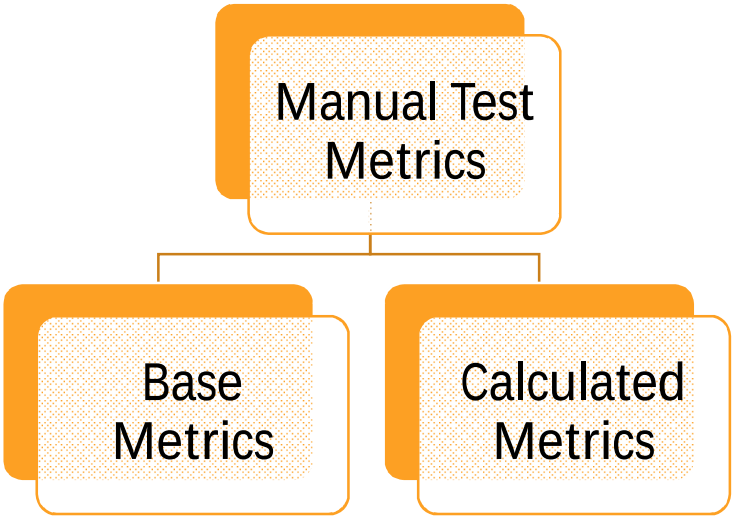
LITHAN

- Process Metrics
  - Improve the process efficiency of the SDLC (Software Development Life Cycle).
- Product Metrics
  - Deals with the quality of the software product.
- Project Metrics
  - Measures the efficiency of a project team or testing tools used by the team members.

Types of Metrics

LITHAN

- Manual Test Metrics
  - Base Metrics
    - Constitutes Raw data gathered by test engineer throughout the testing
    - Known as Direct measure metrics
  - Calculated Metrics
    - Converts the Base Metric data into useful information
    - Known as Indirect measure metrics



Base Metrics & Testing Phase

LITHAN

- Collects the raw data by the test analyst during test case development and execution.
- Used to provide project status reports to the Test lead and project manager .
- Provides the input data to feed into the formulas used to derive Calculated metrics .
- Examples of Base metrics are:
  - # of test cases
  - # of test cases executed

Base Metrics & Testing Phase

LITHAN

TEST METRIC	TESTING PHASE
Number of test cases	Test Development Phase
Number of Test cases Passed	Test Execution Phase
Number of Test cases Executed	Test Execution Phase
Number of Test cases Failed	Test Execution Phase
Number of Test cases under Investigation	Test Development Phase
Number of Test cases Blocked	Test Development Phase / Execution Phase
Number of Test cases Re- executed	Regression Phase
Number of First run Failures	Test Execution Phase
Total Executions	Test Reporting Phase
Total Passes and Failures	Test Reporting Phase
Test case Execution time	Test Reporting Phase
Test Execution time	Test Reporting Phase

Calculated Metrics & Phases

LITHAN

- Derived from the data collected in base metrics
- Converts the Base Metric data into useful information
- Prepared by the Test lead for test reporting purpose (% Complete, % Test Coverage).
- Test lead tracks the progress of project at levels:
  - Module Level
  - Tester Level
  - Project Level
- Provides valuable information leading improvements in Overall SDLC

Calculated Metrics & Phases

LITHAN

- Following Calculated metrics are created at Test Reporting Phase or Post Test Analysis Phase:
  - % of Test cases Passed
  - % of Test Coverage
  - % of Defects corrected
  - % of Test cases Blocked
  - % of Rework
  - % of Test Effectiveness
  - 1st Run Fail Rate
  - Defect discovery rate
  - Overall Fail rate

Other Common Metrics

LITHAN

- Test case execution productivity metrics
- Test case preparation productivity metrics
- Defect rectification time metrics
- Defect metrics
  - Defects by priority
  - Defects by severity
  - Defect slippage ratio


THANK YOU