TESTING METRICS

What is Software testing Metrics ?

In **software testing**, Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. In other words, **metrics** helps estimating the progress, quality and health of a **software testing** effort

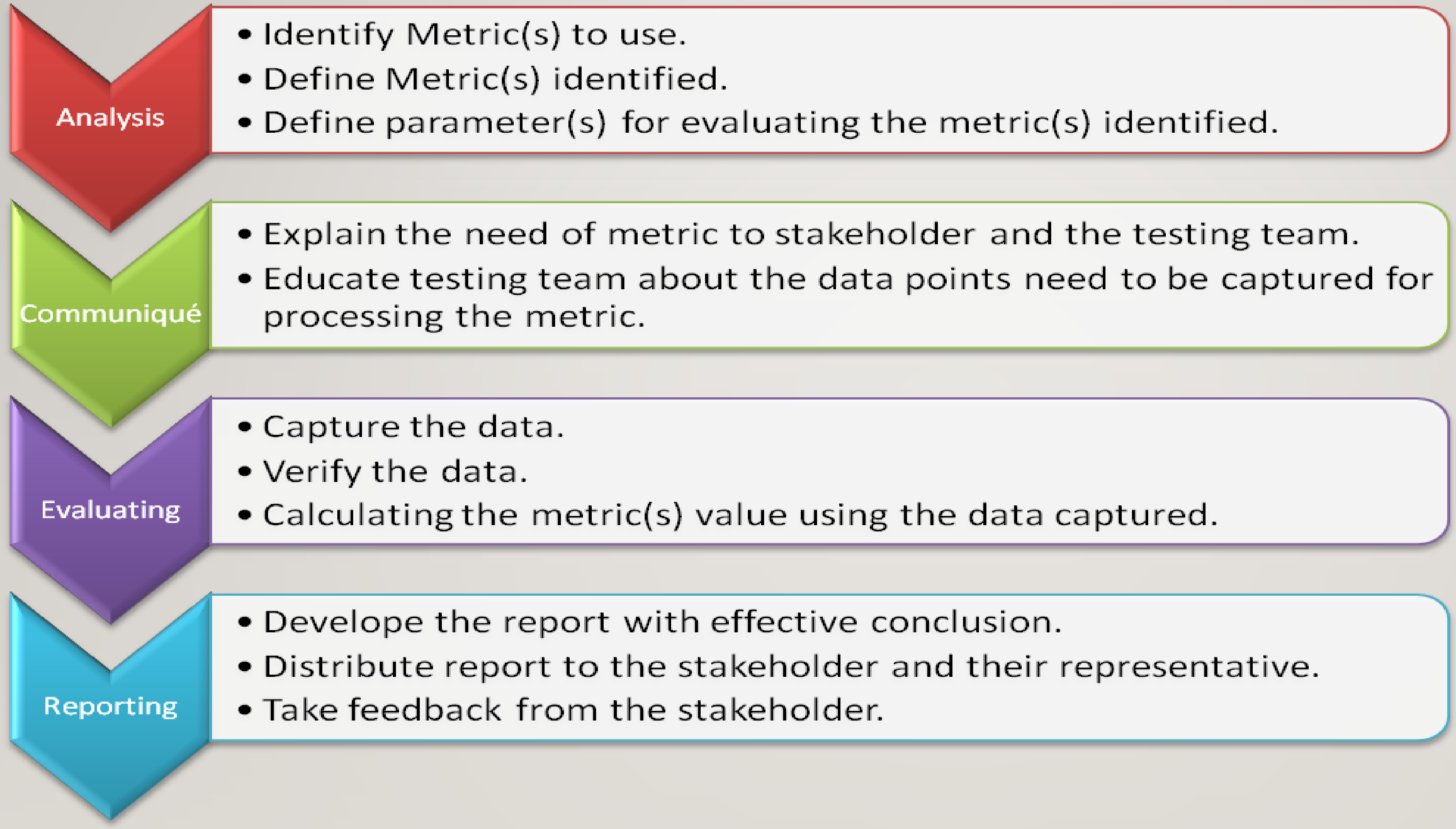
Metrics helps estimating the progress, quality and health of a software testing effort. The ideal example to understand metrics would be a weekly mileage of a car compared to its ideal mileage recommended by the manufacturer.

Why we need Software testing Metrics ?

"We cannot improve what we cannot measure" and Test Metrics helps us to do exactly the same.

- Take decision for next phase of activities
- Evidence of the claim or prediction
- Understand the type of improvement required
- Take decision or process or technology change

Testing Metrics life cycle



Analysis

- Identify Metric(s) to use.
- Define Metric(s) identified.
- Define parameter(s) for evaluating the metric(s) identified.

Communiqué

- Explain the need of metric to stakeholder and the testing team.
- Educate testing team about the data points need to be captured for processing the metric.

Evaluating

- Capture the data.
- Verify the data.
- Calculating the metric(s) value using the data captured.

Reporting

- Develope the report with effective conclusion.
- Distribute report to the stakeholder and their representative.
- Take feedback from the stakeholder.

Metrics & Formula

Metric Name	Objective	Formula
Defect density (LB)	Defect density metric not only indicates the quality of the product being developed, but it can also be used as a basis for estimating a number of defects in the next iteration or sprint. It can be defined as the number of defects per 1,000 lines of code or function points.	Defect Density = Total Number of defects/Total lines of code (OR) Defect Density = Total Number of defects/No of testcases
Sprint velocity	Measure of the amount of work a Team can tackle during a single Sprint	Velocity: Σ story points)
Testing effectiveness	used to measure the test effectiveness of test team and is calculated at the end of test execution phase.	(Total no. of application defects found by test team - Total no. of application defects rejected by the customer/developer) / (Total no. of application defects found by test team + Total no. of defects found by customer/test team during UAT) *10
Defect Removal Efficiency (DRE)	To identify the test effectiveness of the syste	(No. of Defects found during QA testing / (No. of Defects found during QA testing +No. of Defects found by End-user)) * 100
Defect Leakage	Defect Leakage is the Metric which is used to identify the efficiency of the QA testing	Defect Leakage = (No. of Defects found in UAT / No. of Defects found in QA testing.) * 100
Requirement Creep		(Total number of requirements added/No of initial requirements)X100
Schedule Variance	This metric helps to reduce the schedule variance or schedule variation by tracking it from start to end of the project	(Actual efforts – estimated efforts) / Estimated Efforts) X 100
Schedule slippage		(Actual end date – Estimated end date) / (Planned End Date – Planned Start Date) X 100
Test Case Productivity	used to determine the test case design productivity and is used as input parameter for future estimation.	(Number of Test case Prepared) / (Effort spent for Test Case Preparation)
Test execution Productivity	used to determine the test case execution productivity and is used as input parameter for future estimation.	(Number of Test case executed / (Effort spent for Test Case Preparation)
Requirements Leakage Index	gives indication on effectiveness of the requirements evocation process.	(No. of Missed Requirements)/ (Total No. of Initial Requirements) *100
Requirement Stability Index	used to measure the changes in the business requirement added or deleted compared to the original requirements decided at the start of the project.	(Total number of original business requirements + Number of requirements changed till date+ Number of requirements added + Number of requirements deleted) / (total number of original requirements)
Effort Variance	Effort Variance metric is used to study the distribution of workload. It is measured usually at overall project level	(Actual Effort – Estimated Effort) / (Estimated Effort)*100
Error Discovery Rate	Error Discovery Rate metric is used to determine the effectiveness of the test cases and measured at test execution phase.	Total no. of defects found in application /Total no. of test cases or scripts executed
Test Execution Coverage	used to determine the coverage of testing and measured at test execution phase.	Total no. of test cases or scripts executed / To Total no. of test cases or scripts planned to execute *100
Test Design Coverage	helps to measure the functional coverage of test cases designed and improve the test coverage.	(Total number of testable requirements mapped to test cases or Scripts) / (Total number of base lined testable requirements)*100

THANK YOU