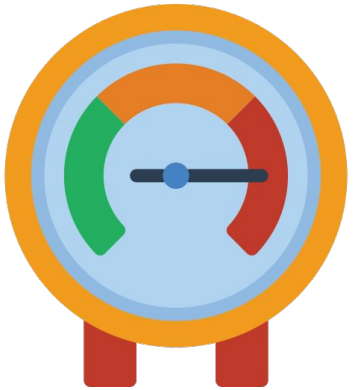
We are

Globant

Testing Metrics

Why do we need Test Metrics?

- Metrics let us know **where we are** and prioritize tasks.
- Testing Metrics is a **powerful risk management tool.**
- It's **never too late to start recording** key information on your project.
- This data can be used **to improve future work**, estimates and quality level.



Benefits of having good Metrics

- Metrics let us **predict** long term direction and scope.
- Provides a basis for estimation and facilitates planning.
- Provides a means for Control / Status Reporting.
- Identifies risks areas that require more testing.
- Provides meters **to flag actions for faster**, more informed decision making.
- Quickly identifies and helps resolve potential problems.
- Allows define the action plan to solve the issues

Test metrics provide an objective measure of the effectiveness and efficiency of testing.

Key Factors to bear in mind

- Collect only the data that you will actually need/use to make informed decisions.
- Do not base decisions solely on data that is variable and can be manipulated.
- Metrics should be decided on the basis of their importance to stakeholders rather than ease of data collection.
- Metrics that are not interested to the stakeholders should be avoided.

Metrics Lifecycle

The process of setting up metrics involves:

1. *Identifying the metric*
2. *Identify data required for the metric*
3. *Communicating to stakeholders*
4. *Capturing and verifying the data*
5. *Analyzing and processing data*
6. *Reporting*

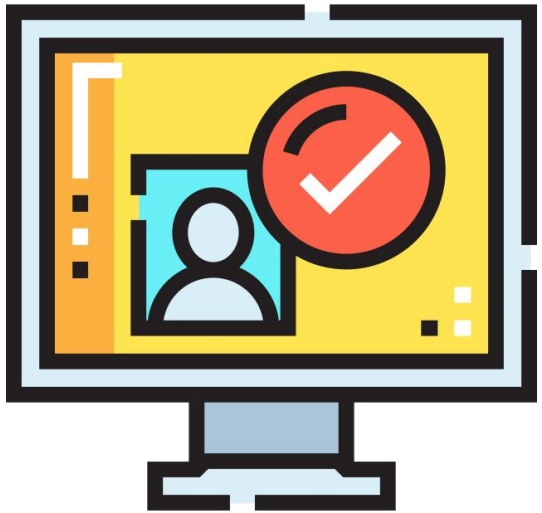
Let's go a bit deeper on each step, so that you can see what is this all about.

Metrics Lifecycle

Step 1: identify the right metrics

The right metrics can be identified only after:

- Thinking about what you want to measure and why.
- Analyzing the value/benefit of each metric.
- Identifying the audience.
- Identifying the goals you are trying to achieve.



Metrics Lifecycle

Step 2: Identify data required for the metric

There are two kinds of metrics:

1. **Base Metrics**: Metrics for which data can be captured directly.

Total TC = TC Passed + TC Failed + TC Blocked
T. TC = 5 + 3 + 2

1. **Derived Metrics**: Derived from base metrics.

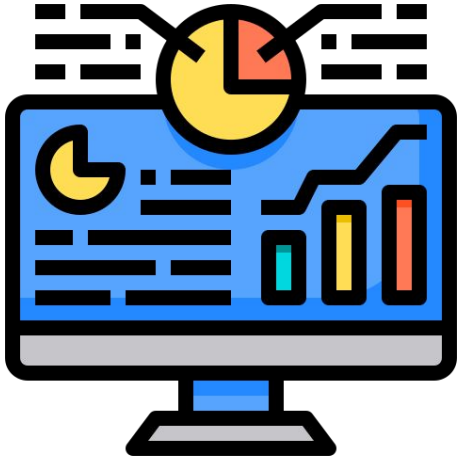
% of Testing Coverage = Total TC - TC Passed
% T. Cov. = 10 - 5

10 ---------- 100%
5 ---------- ? % T. Cov = 50%

Metrics Lifecycle

After identifying suitable metrics, data required must be analyzed by:

- Identifying the source of data for each base metric.
- Avoid ambiguous data.
- Defining a common template for capturing all base metrics.
- Obtaining feedback from the team which captures the data.



Metrics Lifecycle

Step 3: Communication

To ensure better results and commitment, metrics planning must involve all stakeholders:

- Communicate the need for metrics to all affected teams.
- Obtain feedback from stakeholders.
- Communicate expectations to stakeholders – how often the data needs to be collected, how often the reports will be generated.

Metrics Lifecycle

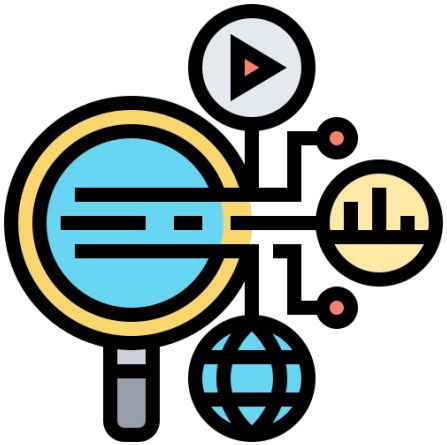
Step 4: Capture and Verify the Data:

- Ensure all the capturing mechanism is set up.
- Communicate and give proper guidelines to the team on the data that is required and why.
- Identify the sources of inaccurate data for each base metric and take corrective steps.

Metrics Lifecycle

Step 5: Analyze and Process the data:

- Verify whether the data filled is accurate and up to date.
- Define the template in which the derived data must be captured.
- Calculate derived metrics based on base metrics.
- Verify whether the metrics are conveying the correct information.



Metrics Lifecycle

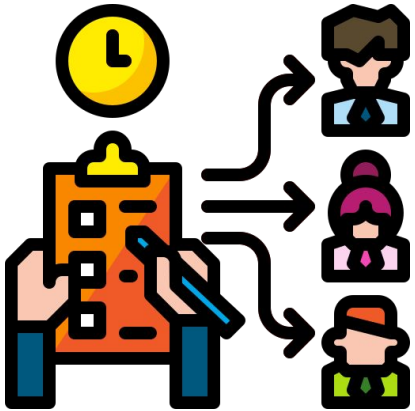
Step 6: Reporting Metrics:

- Reports should contain a summary.
- Reporting should be in a clearly understandable format, preferably graphs and charts with guidelines to understand the report.
- Reports should clearly point out all the issues or highlights.
- Reports should be presented in such a way that metrics are compared against benchmarks and trends shown.

Metric Tips

To avoid the common pitfalls in test metrics the following aspects need to be considered:

1.Management commitment.
2.Measuring too much, too soon.
3.Measuring too little, too late.
4.Wrong metrics.
5.Vague metrics definition.
6.Collecting data that is not used.
7.Lack of communication.
8.Share the results.



Most common Metrics used on a testing project

Test cases definition phase

1. Number of test cases planned Vs Ready for execution.

1. Total time spent on definition Vs Estimated time.

A - 100
B - ?

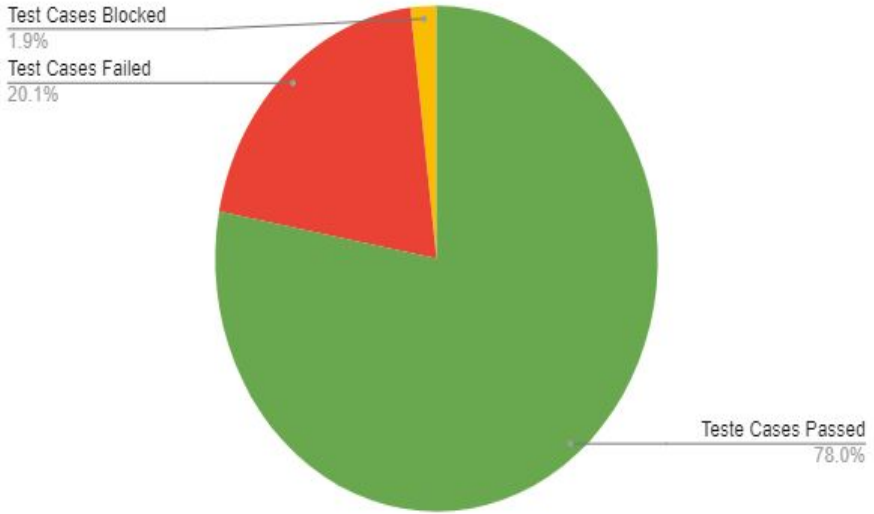
A = 15
B = 6

% Execution =(B*100)/A
% Maria Execution =(6*100)/15
% Maria Execution = 40%

Most common Metrics used on a testing project

Test Execution and Progress:

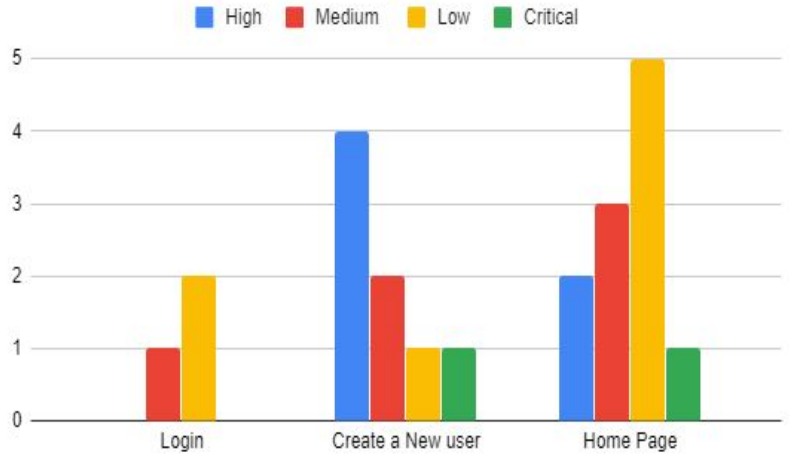
1. TCs executed Vs TCs planned.
2. Time spent on execution Vs Planned.
3. TCs Passed, Failed and Blocked.
4. TCs passed by functional area.



Most common Metrics used on a testing project

Test Product Metrics:

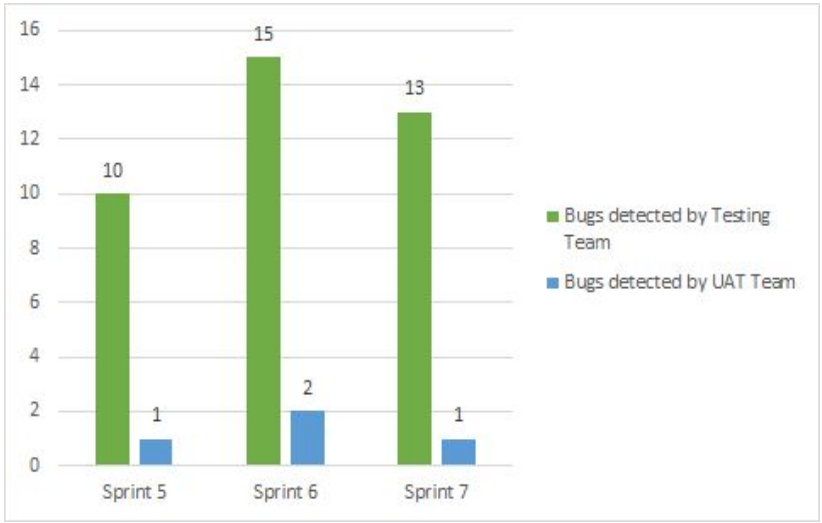
1. Bugs created and closed per cycle.
2. Bugs closed Vs bugs Reopened.
3. Bugs distribution by severity per cycle.
4. Bugs distribution by severity by functional area.
5. Bugs created by test cases executed.
6. Bugs created by status per cycle.



Most common Metrics used on a testing project

Testing Quality and effectiveness Metrics:

1. Bugs not detected by the team but the customer at the UAT.
2. Bugs reported by Testing team / Bugs reported by UAT team + Bugs reported by the Client on Production environment * 100 (severity Level)
3. Bugs not detected by functional area / requirement.
4. Defect Capture Rate.



Thank You!