Software Testing Methodologies

Test Metrics

Metrics

Definition: A metric quantifies or measures a characteristic of a process or product.

- Software metrics are numerical data related to software development. Metrics strongly support software project management activities.
- Software testing metrics basically aims at the testing coverage and defect trends.

For example:

- Number of test cases prepared/Man hour.
- Number of Test cases executed/man hour.

Why We Need Metrics?

“You cannot improve what you cannot measure,

You cannot control what you cannot measure.”

Test metrics helps in:

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

Metrics

Metrics strongly support software project management activities:

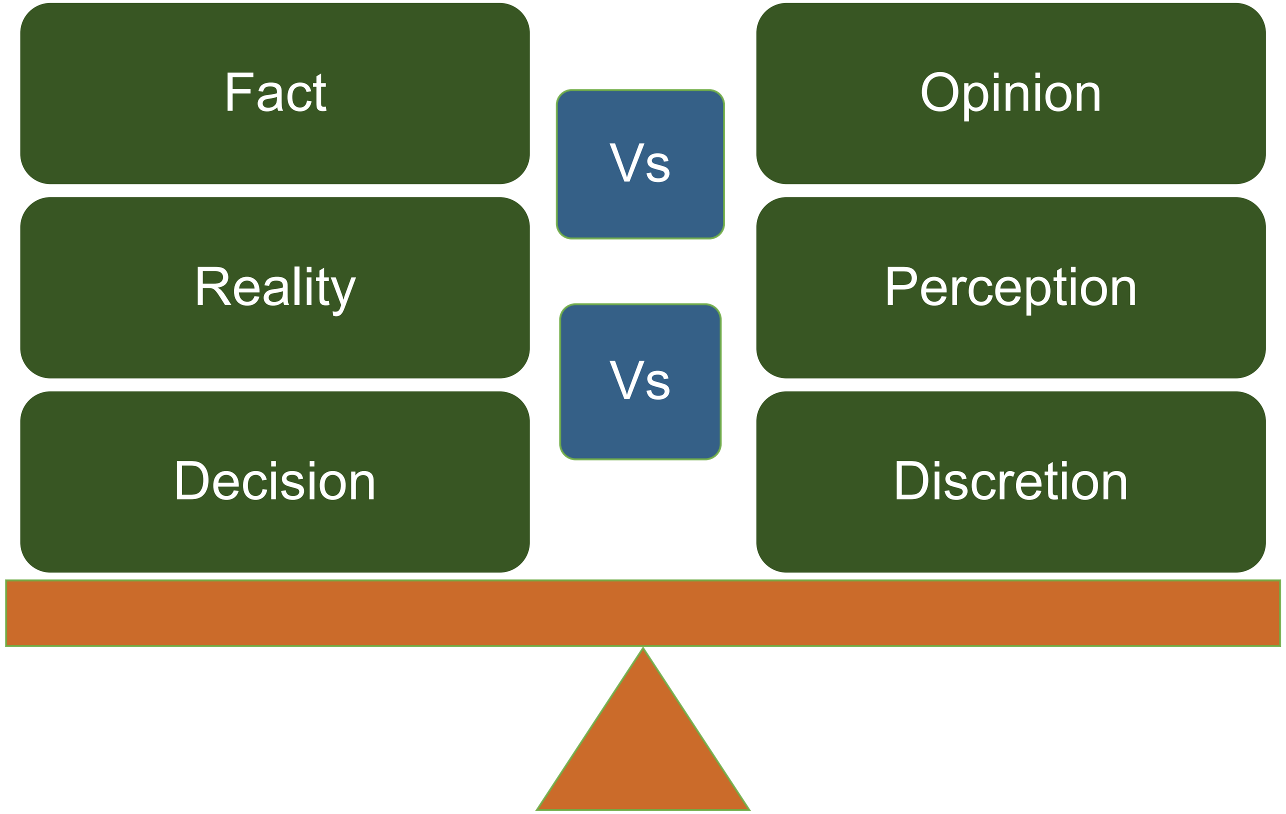
- **Planning** - Metrics serve as a basis of cost estimating, training planning, resource planning, scheduling, and budgeting.

- **Organizing** - Size and schedule metrics influence a project's organization.

- **Controlling** - Metrics are used to status and track software testing activities for compliance to plans.

- **Improving** - Metrics are used as a tool for process improvement and to identify where improvement efforts should be concentrated and measure the effects of process improvement efforts.

Why Data?



Fact

Vs

Opinion

Reality

Vs

Perception

Decision

Discretion

Type of Metrics

- **Base Metrics (Direct Measure)** - Base metrics constitute the raw data gathered by a Test Analyst throughout the testing effort. These metrics are used to provide project status reports to the Test Lead and Project Manager; they also feed into the formulas used to derive Calculated Metrics

  Ex: # of Test Cases, # of Test Cases Executed

- **Calculated Metrics (Indirect Measure)** - Calculated Metrics convert the Base Metrics data into more useful information. These types of metrics are generally the responsibility of the Test Lead and can be tracked at many different levels (by module, tester, or project).

  Ex: % Complete, % Test Coverage

Metrics for Testing Projects

- Effort Variance
- Schedule Slippage
- Productivity
- Review Effectiveness
- Review Efficiency
- Defect Density
- Pre Delivery Defect Density
- Post Delivery Defect Density
- Defect Seepage

Metrics for Testing Projects

- Cost of Quality
- Resource Utilization
- Test Pass Index (%)
- Automation Coverage (%)
- Test Case Creation Rate
- Test Case Execution Rate
- Defect Detection Rate

Metrics for Testing Productivity

- **Name:** Test Case Creation Productivity/Test Case Creation Rate
- **Defi.:** Total number of test cases created per person day
- **Formula:**

$$\textbf{Test Case Creation Productivity/Test Case Creation Rate} = \frac{\textbf{Total \# Of Test Cases Created}}{\textbf{Total Effort Spent On Test Cases Creation}} * 100$$

Metrics Related to Defects

- **Name:** Defect Rejection Ratio
- **Defi.:** Total number of defects rejected vis-a-vis total number of defects logged.
- **Formula:**

$$\text{\% Rejected Defects/ Defect Rejection Ratio} = \frac{\text{\# of Rejected Defects}}{\text{Total number of Defects logged}} * 100$$

Metrics Related to Defects

- **Name:** Defect Slippage
- **Defi.:** Total number of external testing defects vis-à-vis total number of internal testing defects.
- **Formula:**

$$\textbf{Defect Slippage} = \frac{\textbf{\# Of External Testing Defects}}{\textbf{\# Of Internal Testing Defects}} * 100$$

Metrics Related to Defects

- **Name:** Percentage Design Rework
- **Defi.:** Percentage of Test cases changed due to requirement changes.
- **Formula:** $\text{\% Design Rework} = \dfrac{\text{\# Of Test Case Changed Due To Reqmt Changes}}{\text{Total \# Of Test Cases}} * 10$

Metrics Related to Review/Rework

- **Name:** Test Design Effectiveness
- **Defi.:** Percentage of effort spent in doing rework on Test cases due to review comments.
- **Formula:**

$$\textbf{Test Design Effectiveness} = \frac{\textbf{\# Of Hours Reworked Due To Review Comments}}{\textbf{Total Hours Spent On Test Designing}} * \mathbf{10}$$

Metrics Related to Review/Rework

- **Name:** Review Efficiency
- **Defi.:** Total number of review defects logged per the review hours spent.
- **Formula: Review Efficiency =** $\dfrac{\textbf{\# Of Review Defects}}{\textbf{\# Of Review Hours Spent}}$ *** 100**

Other Metrics

- **Name:** Percentage Query Resolution
- **Defi.:** Total Number of queries resolved vis-à-vis total number of queries raised by the testing team.
- **Formula:**

$$\% \text{ Query Resolution} = \frac{\text{\# Of Queries Resolved}}{\text{\# Of Queries Raised By The Testing Team}} * 100$$

Other Metrics

- **Name:** Environment Stability
- **Defi.:** Total Number of hours environment uptime vis-à-vis total execution time.
- **Formula: Environment Stability =** $$\frac{\textbf{\# Of Hours Environment Uptime}}{\textbf{Total Execution Time}} * 100$$

Metrics for Automation

- **Name:** Percentage Automation Coverage
- **Defi.:** Percentage of test cases automated.
- **Formula:**

$$\text{Automation Coverage (\%)} = \frac{\text{Number of Automated Test Cases}}{\text{Total Number Of Test Cases}} * 100$$

Thankyou!