Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

WHITE PAPER

TEST METRICS AND KPI'S

UST Global®, 20 Enterprise, Aliso Viejo CA 92656
Phone: (949) 716-8757 Fax: (949) 716-8396

www.ust-global.com

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Abstract

This document serves as a guideline for understanding metrics and the Key performance indicators for a testing project.

Metrics are parameters or measures of quantitative assessment used for measurement, comparison or to track performance or production. Software metric is a measure of some property of a piece of software or its specifications. Defining and capturing a range of metrics allows them to be measured.

Key Performance Indicators (KPIs) are measurements defined to show the state of critical strategic goals or targets the organization has set for itself. They are calculated by using sets or combinations of metrics. It's a measure of performance, commonly used to help an organization define and evaluate how successful it is in terms of making progress towards its long-term organizational goals.

As a test team, one of the key deliverable is information that conveys the progress of test work and the testing status of the project's development items. Data of this type is vital to understand the current testing effort and review how it was conducted previously and consider how to make improvements in the future. Various groups and individuals within the organization and clients require this information in support of their own activities and decision making. It enables organizations to improve the quality of management decision making by ensuring that reliable and secure information and data is available throughout the Software lifecycle.

What is Metrics?

Metrics - Combination of two or more measures used to compare s/w processes, projects, and products. Set of metrics derived from the data collected from past projects at the organization level. These baselines are used as a basis to arrive at project specific quality goals or KPI's.

Need for metrics in Testing and Quality analysis:

- To determine the quality and progress of testing.
- To calculate how much more time is required for the release
- To estimate the time needed to fix defects.
- Metrics help in deciding the scope of the product that can be released by considering the defect density across modules, their importance to customers and impact analysis of those defects

Benefits of Metrics:

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Key Metrics for a Testing Project:

Project Management Metrics	Metrics in Requirement & Test Design Phase	Metrics in Test Execution Phase
Effort Variation	Test Case Preparation Productivity	Test Effectiveness %
Schedule Variation	Test Design Coverage	Test Execution Coverage
Duration Variation	Review Efficiency	Test Execution productivity
Size Variation	Requirement Stability Index	Test Execution Status
Load Factor	Requirement leakage Index	Error Discovery Rate
Work Efficiency Index		Application Defect Density
Cost of Quality		Quality of fixes
Rework Effort %		Defect Leakage %
% Effort for Project Management		Defect detection efficiency
Risk Identification Efficiency %		Average defect age
Risk Mitigation Efficiency %		

Project Management Metrics:

Effort Variation

This metric is the difference between Estimated and Actual effort as compared against the Estimated Effort.

Objective: The objective of this metric is to study the distribution of workload by Stage and to reduce the deviation of the actual effort expended as against the estimated effort.

When it should be measured: It should be measured at overall project level, Stage level and Task level (Process level and Sub process level for SDLC stages)

Input/Measure	Formula	Example
Actual Effort Estimated Effort	(Actual Effort - Estimated Effort)/ (Estimated Effort) *100	Estimated Effort (in person days) = 5 Actual Effort (in person days) = 7 Effort Variation% = =(7-5)/5 * 100 = 40%

Benefits:

- To determine the efficiency of effort planning.
- To determine the nature and extend of variances and related impact analysis on cost and schedule.

Schedule Variation

This metric is the ratio of difference between the Actual End Date and Planned End Date Vs difference between Planned End Date and Planned Start Date for the project.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Objective: The objective of this metric is to reduce the schedule variation by tracking it from beginning stage of the project through the end of the project, thereby reducing time overruns. Schedule Variation metric is mainly used as an indicator for capability to meet milestones

When it should be measured: It should be measured at overall project level, Stage level and Task level (Process level and Sub process level for SDLC stages). Schedule variation need to be calculated only when the stage is completed.

Input/Measure	Formula	Example
Actual End Date Planned Start Date Planned End Date	((Actual End date – Planned End date) / (Planned End date - Planned Start date)) * 100	Planned Start Date = 1-Jan-13 Planned End Date = 31-Jan-13 Actual Start Date = 2-Jan-13 Actual End Date = 1-Feb-13 Schedule Variation % = (1 / 31) * 100 = 3.22%

Benefits:

- Schedule variation metrics is mainly used as an indicator for capability to meet milestones.

Duration Variation

This metric is the difference between Total Planned Vs Actual duration for the project

Objective: The objective of this metric is same as schedule variation metrics i.e. to reduce the duration variation by tracking it from beginning stage of the project through the end of the project, thereby reducing time overruns. Why we may need both Duration Variation and Schedule Variation is that at times the task/stage may be finished within Planned Duration (Delayed start compensated with delayed finish to the same extent as days it started late.) whereas it might have exceeded the committed deadline, which is ultimately the schedule slippage.

When it should be measured: It should be measured at overall project level, Stage level and Task level (Process level and Sub process level for SDLC stages). Duration variation need to be calculated only when the stage is completed.

Input/Measure	Formula	Example
Actual End Date Actual Start Date Planned Start Date Planned End Date	((Actual End Date – Actual Start Date) – (Planned End Date – Planned Start Date)) / (Planned End Date – Planned Start Date) * 100	Planned Start Date - 1-Jan-13 Planned End Date - 31-Jan-13 Actual Start Date - 2-Jan-13 Actual End Date - 1-Feb-13 Duration Variation % = ((31 –31)/ 31) *100 = 0 %

Benefits:

- Schedule variation metrics is mainly used as an indicator for capability to meet milestones.

Size Variation

This metric is the difference between Estimated Size and Actual size as compared against Estimated Size.

Objective: This metric is used for Comparison of estimated size and the actual size of the project. Results of size variation can be used to analyze reasons for schedule and duration variations and also can be considered for estimate preparations in future.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

When it should be measured: It should be measured at overall project level and Task level.

Input/Measure	Formula	Example
Actual Size Estimated Size	((Actual Size – Estimated Size) / Estimated Size) * 100	Actual Size - 90 FP Estimated Size - 100 FP Size Variation % = ((90 – 100)/100) *100 = -10%

Benefits:

- Schedule variation metrics is mainly used as an indicator for reasons of effort and schedule variations.

Load Factor

Load Factor is computed as ratio of Actual Effort expended in the project to Total Effort available to the project in terms of number of resources allocated to the project for any given period.

Objective: The objective of this metric is to verify the resource allocation.

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Actual Effort Allocated Effort	(Actual Effort / Allocated Effort)	Actual Effort (in person days) = 100 No: Of Resources = 6 No: of days allocated = 20 Allocated Effort (Available effort * No of Resources) in Person Days = 120 i.e (20*6) Load Factor = 100/120 = 0.83

Benefits:

- To know resource utilization in the project
- Helps to utilize the resources optimally.

Work Efficiency Index

Load Work Efficiency Index is the ratio of percentage of work completed to percentage of effort expended.

Objective: This metric is particularly important and mandatory for fixed bid/fixed duration projects, and close monitoring is required to prevent effort overruns.

When it should be measured: It should be measured at overall project level.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Input/Measure	Formula	Example
Percentage of Work completed - (Actual Effort) / (Actual Effort + Effort required to complete remaining work) * 100 Percentage of effort expended - (Actual Effort/ Estimated Effort) *100	(Percentage of Work completed / Percentage of effort expended)	Estimated Effort (in person days) = 10 Actual Effort (in person days) = 5 Effort Required to complete remaining work (in person days) = 7 %Work Completed = (5/(5+7))*100 = 41.66% %Effort Expended = (5/10)*100 = 50% Work Efficiency Index = (41.66 / 50) = 0.83

Benefits:

- This metric is particularly important and mandatory for fixed bid/fixed duration projects, and close monitoring is required to prevent effort overruns.

Cost of Quality

COQ basically consist of 3 types of cost - Prevention Cost, Appraisal Cost and Failure Cost. Failure Cost can be further divided into Internal Failure and External Failure. Here the cost measure is in terms of Effort.

Objective: The objective of this metric is to measure the effort spent on reviews, testing and rework against the production effort.

Definition of Cost of Quality Categories:

Category	Definition	Typical Costs for Software
Prevention	Efforts to ensure product quality	Defect Prevention. Metrics collection and Analysis, Training, process improvements.
Failures Cost	Internal Failure: Quality failures detected prior to product shipment External Failures: Quality failures detected after product shipment	Rework and retesting arising both due to internal and external failures
Appraisal	Discovering the condition of the product	Reviews, Testing and product quality audits

Input/Measure	Formula
Effort spent on Prevention Effort spent on Appraisal Effort spent on Failure Effort spent on Production	(Effort spent on Prevention + Effort spent on Appraisal + Effort spent on Failure) / (Effort spent on Prevention + Effort spent on Appraisal + Effort spent on Failure + Effort spent on Production) *100

Benefits:

- This metric is useful to calculate the total effort spent for QA in a project.

Rework Effort %

Percentage of Effort expended on Rework activities in the project to overall Effort

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Objective: Expending effort on Rework activities is inevitable in any project. Objective of this metrics is to monitor that effort spent on Rework is kept as minimum as possible

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Total Effort for Rework Total Effort for Project	(Total Effort expended on Rework across all stages)/ (Actual Overall Project Effort) *100	Total Effort for Rework = 10 PD Total Effort for Project = 200 PD Rework Effort % =(10/200)*100= 5 %

Benefits:

- Using this metrics actual effort for rework can be found and by analyzing this metrics actions can be taken to reduce rework effort %

% Effort for Project Management

Project Management is an umbrella activity that is carried out throughout the project. Effort expended for Project Management activities to overall Project effort is computed as percentage figure.

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Effort for Project Management Total Effort for Project	(Effort expended for Project Management activities)/ (Overall Project Effort) *100	Total Effort for Project Management = 10 PD Total Effort for Project = 200 PD Rework Effort % =(10/200)*100= 5 %

Benefits:

- This metrics is used to identify & of test management activities for a testing project.

Risk Identification Efficiency %

This metric determines the efficiency of identifying risk in a project. This helps in planning for the mitigation and contingency to be carried out in a project. This metric is computed as Number of Risk identified to Number of Risk occurred, expressed as percentage figure

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Total no of Risks Identified Total no of Unanticipated Risk Occurred	(Total no of Risks Identified /(Total no of risks identified + Total no of Unanticipated Risk Occurred))*100	Total no of Risks Identified = 6 Total no of Unanticipated Risk Occurred = 2 Risk Identification Efficiency % = (6/8)*100 = 75%

Benefits:

- This metric determines the efficiency of identifying risk in a project
- This helps in planning for the mitigation and contingency to be carried out in a project

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Risk Mitigation Efficiency %

This metric is computed as Number of Risk mitigated to Number of Risk identified for mitigation. This gives insight into efficiency to mitigate risk.

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Total no of Risks Identified for mitigation No. of Risk mitigated but occurred	((No. of Risk identified for mitigation - No. of Risk mitigated but occurred) / (No. of Risk identified for mitigation)) * 100	Total no of Risks Identified for mitigation = 10 No. of Risk mitigated but occurred = 1 Risk Mitigation Efficiency % = (9/10)*100 = 90 %

Benefits:

- Gives insight into efficiency to mitigate risk and take further actions.

Metrics in Requirement & Test Design Phase:

Requirement Stability Index:

Requirements Stability Index gives indication on effectiveness of the Requirements gathering process. It's a comparison of change to requirements (added/ deleted/ modified) and the original requirements.

Objective: This is used to measure the changes that are coming in (compared to the original requirements decided at the start of the project) during the course of the project. This measures the dimension of changes in terms of number of requests.

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Total no of Original Requirements Cumulative no of requirements changed (till date) Cumulative no of requirements added (till date) Cumulative no of requirements deleted (till date)	(Total no of Original Requirements + Cumulative no of requirements changed (till date) + Cumulative no of requirements added (till date) + Cumulative no of requirements deleted (till date)) / (Total no of Original Requirements	Total number of Original Requirements = 50 Cumulative number of requirements changed (till date) = 1 Cumulative number of requirements added (till date) = 2 Cumulative number of requirements deleted (till date) = 1 RSI = (50+1+2+1)/50 = 1.08

Benefits:

- This can be factored into plan and rework effort and schedule for remaining phases/activities/modules
- This helps to know how stable the requirements are.

Requirement leakage Index

Requirements Stability Index gives indication on ratio of Number of Missed requirements with Total No. of requirements

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Objective: Requirements Leakage Index gives indication on effectiveness of the Requirements elicitation process. Requirements Leakage index is given by Number of Missed Requirements to Number of Total Requirements, expressed as percentage figure. Here the missed requirements are those that were missed by the Project team during Requirements Elicitation process

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula	Example
Total No. of Initial Requirements Total No. of Missed Requirements	(No. of Missed Requirements)/ (Total No. of Initial Requirements) *100	Total No. of Initial Requirements = 50 Total No. of Missed Requirements = 2 RLI = (2/50)*100 = 4

Benefits:

- This helps to know effectiveness of the Requirements elicitation process.

Review Efficiency

It's the ratio of number of review defects to total defects in software (review and testing)

Objective: This metric shows the efficiency of the review process. A higher ratio of Review Defects to Total Defects indicates a very efficient Review Process. However, a lower ratio need not necessarily mean that the review is inadequate. This may be due to frequent changes in scope, requirements etc. Typically, the review defects include all reviews, starting from the Requirements Stage. All review comments that arise from the review of Code/unit / integration / system test plans / procedures / cases shall be counted under Review defects. The total number of Defects includes all the defects found in reviews and testing.

When it should be measured: It should be measured at overall project level and stage level.

Input/Measure	Formula	Example
Number of Review defects Total number of Testing Defects including customer reported test defects	(Number of Review defects) / (Total number of Review + Testing Defects [including customer reported test defects])*100	Number of Review defects = 10 Total number of Testing Defects including customer reported test defects = 100 Review Efficiency% = (10/110)*100 = 9.09

Benefits:

- This helps to know effectiveness of the reviews performed and to improve the review process.
- Increase in review efficiency decrease the defect leakage to next stage.

Test Case Preparation Productivity

It determines the number of Test cases / Scripts that can be prepared per person days of effort.

Objective: To determine the test design productivity based on which future estimation can be done for the similar projects

When it should be measured: It should be measured during test design phase

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Input/Measure	Formula	Example
No of Test Cases or Scripts Prepared Effort spent for Test Case/ Script preparation	(No of Test Cases or scripts)/ (Effort spent for Test Case/script Preparation)	No of Test Cases or Scripts = 1000 Effort spent for Test Case preparation in PD = 100 Test Case Preparation Productivity =1000/100

Benefits:

- This helps to determine number of test cases prepared per person days of effort.
- If productivity is low due to reasons like too many changes to requirements, it helps us to factor the effort into the project plan and re estimate the remaining activities/stages.

Test Design Coverage

This is to measure the percentage of test case coverage against the number of test requirements.

Objective: The objective of this metrics to measure the functional coverage of test cases designed.

When it should be measured: It should be measured during test design phase.

Input/Measure	Formula	Example
Total number of baselined testable requirements mapped to test cases Total number of baselined testable requirements	((Total number of testable requirements mapped to test cases or Scripts) /(Total number of baselined testable requirements)*100	Total number of baselined testable requirements mapped to test cases = 1100 Total number of baselined testable requirements = 1080 Design Coverage % =(1080/1100)*100 =98.18

Benefits:

- This metrics is used as an indicator of quality of test design and used to improve test coverage.

Metrics in Test Execution Phase:

Test Effectiveness %

This metrics shows the efficiency of removing defects by internal Testing before delivering to customer. It determines quality of defects logged.

Objective: The objective of this metrics is to determine the testing efficiency.

When it should be measured: It should be measured at end of every build or test execution phase.

Input/Measure	Formula	Example
Total Number of defects found by test team Total Number of defects Rejected by Customer Total number of defects found by customer during UAT	((Total no of application defects found by test team -Total number of application defects rejected by the customer)/(Total no of application defects found by test team + Total number of defects found by customer during UAT))*100	Total Number of defects found by test team = 100 Total Number of defects Rejected by Customer = 10 Total number of defects found by customer during UAT = 2 Test Effectiveness % = ((100 -10)/(100 + 2))*100 = 88.23 %

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Benefits:

- This metrics is used as an indicator of testing effectiveness of test team.

Test Execution Coverage

This metrics is useful to measure the number of test case executed against the plan.

Objective: The objective of this metrics is to determine the coverage of testing.

When it should be measured: It should be measured at test execution phase.

Input/Measure	Formula	Example
Total Number of test cases or scripts executed Total Number of test cases or scripts planned to execute	(Total Number of test cases or scripts executed / Total Number of test cases or scripts planned to execute)*100	Total Number of test cases executed = 1100 Total Number of test cases planned to execute = 1080 Test Execution Coverage = (1080/1100)*100 = 98.18

Benefits:

- This metrics is used as an indicator of coverage of testing.

Test Execution productivity

It determines the number of Test cases / Scripts that can be executed per person days of effort.

Objective: To determine the Execution productivity based on which future estimation can be done for the similar projects

When it should be measured: It should be measured during test execution phase.

Input/Measure	Formula	Example
No of Test Cases or Scripts Executed Effort spent for Test Case/ Script preparation	(No of Test Cases or scripts executed)/ (Effort spent for Test Case or script Execution)	No of Test Cases or Scripts Executed = 1000 Effort spent for Test Case/Script preparation = 100 PD Test Case Execution Productivity = 1000/100 = 10

Benefits:

- Helps to determine number of test cases executed per person days of effort.

Test Execution Status

This metrics is useful to measure the percentage of test cases executed.

Objective: The aim of this metric is to determine the status of Test Execution and overall project execution.

When it should be measured: It should be measured during test execution phase.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Input/Measure	Formula	Example
Total Number of test cases or scripts executed for each status (Passed /Failed/ Blocked/Not Completed) Total Number of test cases.	(Total Number of test cases or scripts executed for each status/ Total Number of test cases)*100	Total Number of test cases or scripts executed for each status (Passed - 70/ Failed -10/ Blocked -10 /Not Completed-10) = 50 Total Number of test cases = 100 % Test Cases Passed = (70/100)*100 = 70% % Test Cases Failed = (10/100)*100 = 10% % Test Cases Blocked = (10/100)*100 = 10% % Test Cases Not Completed = (10/100)*100 = 10%

Benefits:

- Helps to determine status of execution.

Error Discovery Rate

It is defined as ratio of defects per test cases

Objective: The aim of this metrics is to determine the effectiveness of the test cases.

When it should be measured: It should be measured during test execution phase.

Input/Measure	Formula	Example
Total no of Defects found Total no of test cases or scripts executed	Total number of defects found in application /Number of test cases or scripts executed	Total no of Defects found = 100 Total no of test cases or scripts executed = 800 Error Discovery rate = 100/800 = 0.125 Defects/Test cases

Benefits:

- Helps to determine effectiveness of the test cases.
- It is used to analyze build stability and support release decision, progress metric reported on a daily basis

Quality of fixes

It determines the quality of fix done by development team. This will also show how often previously working functionality was adversely affected by software fixes.

Objective: The aim of this metrics is to know the quality of a build in terms of defect fixing.

When it should be measured: It should be measured during each build and end of test execution phase

Input/Measure	Formula	Example
Total no of Fixed Defects Total no of reopened bugs Total no of new Bugs due to fix	(Total no of Defects reported as fixed - Total no. of reopened bugs) / (Total no of Defects reported as fixed + Total no. of new Bugs due to fix)*100	Total no of Fixed Defects = 100 Total no of reopened bugs = 10 Total no of new Bugs due to fix = 5 Quality Of Fix % =((100-10)/(100+5))*100 = 85.7 %

Benefits:

- Quantitatively calculates the quality of bug fixing.
- Lower the percentage indicates the poor quality of defect fixes.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Application Defect Density

Defined as the ratio of defects to test case or the ratio of defects to effort

Objective: The aim of this metrics is to determine the application stability.

When it should be measured: It should be measured build wise and end of test execution phase.

Input/Measure	Formula	Example
Total number of defects found in the application Actual Size (TCP/No of test cases executed for the release)	(Total number of defects found in the application /Actual Size (TCP/No of test cases executed for the release)	Total number of defects found in the application = 80 Actual Size (TCP/No of test cases executed for the release) = 1000 (Test cases) Application Defect Density = 80/1000 =.08

Benefits:

- The module in which the defect density is high can be identified.
- It helps to know density of defects spread across the modules.

Defect Leakage %

This metric gives a very good indication of the review / testing process within a stage. It determines the % of defect leaked to the subsequent phases

Objective: To determine the Defect Leakage

When it should be measured: It should be measured at overall project level and stage level.

Input/Measure	Formula	Example
Total number of defects captured in that stage Number of defects attributed to a stage but only captured in subsequent stages (UAT)	(Number of defects attributed to a stage but only captured in subsequent stages) / (Total number of defects captured in that stage + Total Number of defects attributed to a stage but only captured in subsequent stages) *100	Total number of defects captured in that stage = 90 Number of defects attributed to a stage but only captured in subsequent stages (UAT) = 10 Defect Leakage % = (10/(90+10))*100 =10%

Benefits:

- This metrics gives a good indication of testing process within a stage. Any defect leaked to next stage indicates that test mechanism is not adequate for that stage.
- A high leakage rate indicates that the process of testing carried out needs to be looked upon.

Defect Removal Efficiency %

Comparison of internally reported defects with total defects (including customer reported).

Objective: This metrics shows the efficiency of removing defects by Internal Review and Testing process before shipment of the product to customer. The metric involves pre-ship defects and post-ship defects.

When it should be measured: It should be measured at overall project level and stage level.

Authors:
Gregory Jose Gregory.jose@ust-global.com
Jusha Joseph Jusha.Joseph@ust-global.com

Input/Measure	Formula	Example
Total number of Pre-shipment Defects. Total number of post-shipment Defects	(Total number of Pre-shipment Defects)/ (Total number of Pre-shipment Defects + Total number of post-shipment Defects) *100	Total number of Pre-shipment Defects.= 100 Total number of post-shipment Defects = 2 Defect Removal Efficiency % = (100/102)*100 = 98.04

Benefits:

- This is a very good metrics for all operational models, as it is indicative of quality of the product delivered

Average defect age

Age of a defect that remains in the system in terms of number of stages on an average it passes without detection.

Objective: This metric gives an indication as to on an average how long it takes for the defects to get detected, in terms of number of stages.

When it should be measured: It should be measured at overall project level.

Input/Measure	Formula
Date of fix of each defect Date open for each defect Total no of defects	Sum (Date of Fix for each defect - Date of open for each defect)/Total no of defects

Benefits:

- If the average defect age is less than 1 that means the project, on an average, is able to find the defects within the same stage itself
- If it is greater than or equal to 1 then the project finds more defects only in the subsequent stages and not in the same stage.

References:

1) Kaner, Cem and Walter P. Bond
Software Engineering Metrics: What Do They Measure and How Do We Know?

2) Weinberg, Gerald M.
Quality Software Management, Volume 2: First-Order Measurement.

Biography

Jusha Joseph is a test professional with 5 years of experience and is currently working with UST Global Trivandrum .She has testing experience in different domains like SAP ,Web Client server and Mobile. She is an ISTQB Advance level certified QA professional and a Certified Scrum Master.

Gregory Jose is a test professional with 7 years of experience and is currently working with UST Global Trivandrum. He is an ISTQB certified professional and currently working as Test Analyst.