
Testing Concepts
Lesson 5: Testing Metrics
Capgemini

Lesson Objectives

To understand the following topics

- Monitoring the Progress
- Metrics of Test Progress
- Reporting Test Status
- Test Control
- Configuration Management & Configuration Control
- Products for Configuration Management in Testing
- Definition of Metrics
- Need of Metrics
- Metrics for Testing
- Types of Metrics
- Types of Metrics – Project Metrics
- Types of Metrics – Process Metrics
- Types of Metrics – Productivity Metrics
- Types of Metrics – Closure Metrics

Monitoring the Progress

Why Test monitoring is necessary?

- To know the status of the testing project at any given point in time
- To provide visibility on the status of testing to other stake holders
- To be able to measure testing against defined exit criteria
- To be able to assess progress against Planned schedule & Budget

IEEE 829 Standard: Test Log Template Content

- Test Log identifier
- Description (items being tested, environment in which the testing is conducted)
- Activity and event entries (execution description, procedure results, environmental information, anomalous events, incident report identifiers)

Metrics of Test Progress

Metrics should be collected during and at the end of a test level. They are also valuable input into process improvement. Common metrics for test progress monitoring include:

- The extent of completion of test environment preparation
- The extent of test coverage achieved, measured against requirements, risks, code, configurations or other areas of interest
- The status of the testing compared to various test milestones
- The economics of testing, such as the costs and benefits of continuing test execution in terms of finding the next defect or running the next test.

Reporting Test Status

Reporting test status is about effectively communicating our findings to other project stakeholders. It is usually done through Test Summary Report

IEEE 829 Standard: Test Summary Report Template

- Test summary report identifier
- Summary
- Variances
- Comprehensive assessment
- Summary of results
- Evaluation
- Summary of activities
- Approvals

Test Control

Test control is the response to Test Monitoring and Test Reporting that allows us to be IN CONTROL of the project

Issues need to monitored and reported

The process of control is the corrective actions required to put a testing effort (project) back on track

For Example:

- Re-prioritize tests when an identified risk
- Change the test schedule based on availability of a test environment

Configuration Management & Configuration Control

Configuration Management:

- A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configuration item

Configuration Control or Version control:

- An element of configuration management, consisting of evaluation, coordination, approval or disapproval and implementation of changes to configuration items after formal establishment of their configuration identification

Products for Configuration Management in Testing

Test plans
Test designs
Test cases:
- Test input
- Test data
- Test scripts
- Expected results

Actual results
Test tools

What would not be under configuration management?
- Live data

Definition of Metrics

Metrics - definition

- A metric is the measurement of a particular characteristic of a program's performance or efficiency.
- A quantitative measure of the degree to which a system, component or process possesses a given attribute.

Process measurement and analysis, and utilization of quantitative methods for quality management are the two key process activities at Level 4 Maturity of CMM.

While the need for metrics has been recognized, implementation of structured measurement programs is lagging, especially in the software testing area.

Efficient test process measurement is essential for managing and evaluating the effectiveness of a test process.

Test metrics are an important indicator of the effectiveness of a software testing process.
There are various metrics that evaluate the effectiveness of the testing process and also the other metrics that are affected due to the testing process.

In general, we measure following things
Functionality
Schedule
Cost

Need of Metrics

Why Measure

- Tracking Projects against plan
- Take timely corrective actions
- Getting early warnings
- Basis for setting benchmarks
- Basis for driving process improvements
- Tracking process performance against business

Test Metrics data collection helps predict the long term direction and scope for an organization
Provides a basis for estimation and facilitates planning for closure of the performance gap
Provides a means for control/status reporting
Identify critical processes that will be monitored statistically
Identifies risk areas that require more testing
Provides meters to flag actions for faster and more informed decision making
Helps in identifying potential problems and areas of improvement
Provide an objective measure of the effectiveness and efficiency of testing

Metrics for Testing

Defect Density

- Total Defect density = (Total number of defects including both impact and non-impact, found in all the phases + Post delivery defects)/Size

Average Defect Age

- Average Defect age = (Sum of ((Defect detection phase number – defect injection phase number) * No of defects detected in the defect detection phase))/(Total Number of defects till date)
- Defect Removal Efficiency
- DRE = 100 * No. of pre-delivery defects / Total No. of Defects

Defect Density is the no. of defects found in a unit.

Defects are measured as found in
Reviews
Testing
Acceptance
Warranty
The defect that are hampering the functionality are 'impact' defects. Defect those are not affecting functionality like look n feel errors, displacement errors are non-impact errors.
The defect impacting the functionality are direct. Impact defect whereas the commenting standard. Defect found in code review is non-impact.

E.g. If Total number of direct impact defects found in all the phases = 20, Post delivery defects = 10, Size = 100, Direct Impact Defect density = 20+15/100 = 0.30
We should try n minimize it. We should minimize impact & non-impact errors.
Average Defect Age
The Average defect age tells us for how long the defect was in the system after it was injected.
For E.g. If Defect detection phase number= 4, defect injection phase number = 2, No of defects detected in the defect detection phase = 10, Total Number of defects till date = 40
Defect Age = 4-2*20/40 = 1 ---------- This is the defect age. Should be kept minimum. Difference should be kept minimum.
Average Defect age is the sum of defects detected at all the stages.
Defect Removal Efficiency
For E.g. If No. of pre-delivery defects = 5, Total No. of Defects = 20,
DRE = 100 *5/20 = 100*0.25=25
This has to be 100%. No of pre-delivery defects should be greater than total no. of defects. So
the errors should get find before delivery.
Low Defect Removal Efficiency means – More defects left undetected before delivery, Reviews and Testing failed to detect them.

Metrics for Testing (Cont.)

Review Effectiveness

- Review Effectiveness = 100 * Total no. of defects found in review / Total no. of defects

Cost of finding a defect in review(CFDR)

- Cost of finding a defect in reviews = (Total efforts spent on reviews / No. of defects found in reviews)

Cost of finding a defect in testing(CFDT)

- Cost of finding a defect in testing = (Total efforts spent on testing / defects found in testing)

Review Effectiveness :
The review effectiveness tells us how effective is review process. If all the defects are found during the review then the effectiveness % will be 100.
E.g. If Total no. of defects found in review = 20, Total no. of defects=40
RE = 100*20/40= 50. So defects should get find at review stage only to achieve 100 % effectiveness.
Low Review Effectiveness means : More defects detected in testing. Reviews failed to detect early

Cost of finding defect in review
This metric tells us the effort spent in finding a defect in reviews. Cost of reviews include all the efforts spent in review briefing, defect recording etc. This includes reviewer, recorder and creators time if creator is attending the review his/her time also should be recorded.
For E.g. If Total efforts spent on reviews = 40 hrs, No. of defects found in reviews = 20, CFDR= 40/20=2 Hrs

Cost of finding defect in testing

This metric computes the Cost of finding a defect in testing . Total time spent on testing includes time to create and review, run the test cases and recording the defects. This should not include the time spent in fixing the defects.

CFDT = Total efforts spent on testing / defects found in testing = 60/30=2 hrs

Metrics for Testing (Cont.)

Components of CoQ – Prevention Cost, Appraisal Cost, Failure Cost

Prevention Cost: (Green Money)

- Cost of time spent in DP meetings
- Cost of time spent by DPR/PM/TL on analysis of defect entries/discussions with team members
- Cost of time spent by the team in implementing the preventive actions identified from project start date to till date

Appraisal Cost: (Blue Money)

- Cost of time spent on review and testing activities from the project start date to till date

Failure Cost: (Red Money)

- Failure costs include internal and external failure costs
- Cost of time taken to fix the pre and post delivery defects
- Expenses incurred in rework – Customer does not pay for this

Prevention - Money required preventing errors and to do the job right the first time is considered prevention cost. This category includes money spent on establishing methods and procedures, training workers and planning for quality. Prevention money is all spent before the product is actually built.

Appraisal – Appraisal costs cover money spent to review completed products against requirements. Appraisal includes the cost of inspections, testing and reviews. This money is spent after the product or subcomponents are built but before it is shipped to the user.

Failure – Failure costs are all costs associated with defective products. Some failure costs involve repairing products to make them meet requirements. Others are costs generated by failures, such as the cost of operating faulty products, damage incurred by using them and the costs incurred because the product is not available. The user or customer of the organization may also experience failure costs.

Metrics for Testing (Cont.)

Money spent beyond what it would cost to build a product right first time

Cost of Quality

- % Cost of Quality = (Total efforts spent on Prevention + Total efforts spent on Appraisal + Total efforts spent on failure or rework)*100/(Total efforts spent on project)
- Failure cost = Efforts spent on fixing or reworking the pre-delivery defects + (3 * efforts spent on fixing or reworking the post-delivery defects)

Cost of Quality - Cost of Quality consists of Prevention cost, Appraisal cost & Failure (or Rework) cost. Here, cost is the efforts measured in terms of person days.

Prevention cost consists of efforts spent on preventing defects such as:

1. Time spent in various Defect Prevention meetings
2. Time spent by Defect Prevention Reviewer/Project Leader on analysis of defect entries/discussions with team members/SQA
3. Time spent by the team in implementing the preventive actions identified from project start date to till date.

For E.g. If Total efforts spent on Prevention = 20, Total efforts spent on Appraisal =30, Total efforts spent on failure or rework = 40, Total efforts spent on project = 140

Cost of quality = (20+30+40)*100/140= 64. This has to decrease. Total efforts of all the activities should match with efforts spent on time.

Failure cost = Efforts spent on fixing or reworking the pre-delivery defects + (3 * efforts spent on fixing or reworking the post-delivery defects)

(As the impact of post delivery defects will be high, weightage of “3” has been attached to it)

For E.g. Efforts spent on fixing or reworking the pre-delivery defects = 40, efforts spent on fixing or reworking the post-delivery defects = 20 Failure cost = 40+(3*20) = 40+60 = 100.

We need to keep it minimum. So, efforts for post-delivery defects should be 0 to minimize failure cost.

Metrics for Testing (Cont.)

Test Case Effectiveness

- Test Case Effectiveness = # of defects detected using the test cases * 100/ total # of defects detected in testing
- This metrics defines the effectives of the test cases which is measured in terms of the number of defects found in testing with using the test cases
- Source of Data
  - Defect data and number of test cases from test execution report

P.S.: - These metrics are mainly applicable to V&V projects

Test Case Effectiveness (TCE) : This metrics defines the effectives of the test cases which is measured in terms of the number of defects found in testing without using the test cases.

Test Case Effectiveness = # of defects detected using the test cases * 100/ total # Of defects detected in testing

E.g. If # of defects detected using the test cases = 30, total # of defects detected in testing = 50

Test Case Effectiveness = 30*100/50 = 60
Effectiveness needs to be high. # of defects detected using the test cases should match total # of defects detected in testing to have effectiveness.

Metrics for Testing (Cont.)

Test Case Adequacy

- Test Case Adequacy = No. of actual Test cases * 100 / No. of test cases estimated
- This metrics defines the number of actual test cases created vs. the estimated test cases at the end of the test case preparation phase
- The estimated No. of the test cases are based baseline figures and then added to test plan
- Number of Actual Test cases is also derived from project plan

P.S.: - These metrics are mainly applicable to V&V projects

Test Case Adequacy
This metrics defines the number of actual test cases created vs. the estimated test cases at the end of the test case preparation phase. The estimation of the planned test cases are based upon the baseline figures.

For E.g.

If No. of actual Test cases = 30, No. of test cases estimated = 40
Test Case Adequacy = 30*100/40 = 75

This has to be 100%. For which, No. of actual Test cases should match No. of test cases estimated. There should not be vast difference between them to achieve higher adequacy.

Metrics for Testing (Cont.)

Defect Detection Index

- Defect Detection Index = # of defects detected in each phase / total # of defects planned to be detected in each phase
- This is a measure of actual vs. planned defects at the end of each phase
  - Defect data from execution report

P.S.: - These metrics are mainly applicable to V&V projects

Defect Detection Index (DDI) - This is a measure of actual vs. planned defects at the end of each phase.

For E.g. If # of defects detected in each phase = 20, total # of defects planned to be detected in each phase = 100

Defect Detection Index = 20/100= 0.50

Metrics for Testing (Cont.)

Test Coverage: The following are the test coverage metrics:

Test Design:

- # Of Requirements or # Of Use Cases covered / # Of Requirements or # Of Use Cases Planned

Test Execution:

- # Of Test scripts or Test cases executed/# Of Test scripts or Test cases Planned

Test Automation:

- # Of Test cases automated/# Of Test cases

Metrics for Testing (Cont.)

Test Effectiveness
- # Of Test Cases failed (found defects)/# Of Test Cases executed

Delivered Defect Rate (Per 1000 Person Hours)
- (# Of Defects * 1000)/ Actual Effort

Defect Injection Rate (No of Defects / 100 Person Hours)
- No of Defects[phase wise] * 100/ Actual Effort[phase wise]

Defect Removal efficiency
- (# of Defects found internally / Total # Of(internal + external) Defects found) * 100

Test Effectiveness
Of Test Cases failed (found defects)/# Of Test Cases executed
This metric indicates the effectiveness of the Test Cases in finding the defects in the product

Delivered Defect Rate (Per 1000 Person Hours)
(# Of Defects * 1000)/ Actual Effort
The purpose of this parameter is to measure the defect slippage to our customer vis-à-vis total effort. This parameter will be used to predict the residual defects in the delivered product with our current capability.

Defect Injection Rate (No of Defects / 100 Person Hours)
No of Defects [phase wise] * 100/ Actual Effort [phase wise]
This is used to detect the defects injected during STLC Phases.

Defect Removal efficiency
(# of Defects found internally / Total # Of(internal + external) Defects found) * 100

It indicates the number of defects leaked after several levels of review and these defects are slipped to the customer.

This is same as review effectiveness. Need to discuss before removing.

Types of Metrics

There are several types of metrics

- Project Metrics
- Process Metrics
- Productivity Metrics
- Closure Metrics

Types of Metrics – Project Metrics

The following are the Project Metrics

- Test Coverage
- Defect Density
- Defect arrival rate

Defect arrival rate:
Of Defects * 100 / # of Test Cases planned for Execution
This metric indicates the quality of the application/product under test.
Lower the value of this parameter is better.

Types of Metrics – Process Metrics

The following are the Process Metrics

- Test Effectiveness
- Effort Variance
- Schedule Variance
- Cost of Quality
- OTD
- Delivered Defect Rate
- Defect Slippage or Test escape

Effort Variance
(Overall and Variance at each milestone)
((Actual effort - Planned effort) / Planned effort) * 100
The purpose of this parameter is to check the accuracy of the Effort estimation process to improve the estimation process.

Schedule variance
(Actual end date - Planned end date) / (Planned end date - Plan start date + 1) * 100
It depicts the ± buffer that can be used for optimum use of resource deployment And monitor the dates committed to the client and helps in better planning of future tasks.

OTD
# of deliveries on schedule / total # of deliverables made

This parameter is to measure the timely delivery of the Deliverables.

Defect slippage or Test escape
(Total # Of External Defects / Total # Of Defects detected (Internal+External)) * 100
This measure helps us to know how effectively we are detecting the defects Various stages of internal testing

Types of Metrics – Process Metrics (Cont.)

The following are the Process Metrics

- Defect Injection Rate
- Rejection Index
- Resource Utilization
- Review Effectiveness
- Test Case Design Rework Index
- Defect Removal Efficiency

Rejection index
Of Defects rejected/# Of Defects raised
The purpose of this parameter is to measure the Quality of the defects raised

Resource Utilization
Actual effort utilized in the month for project activities /Total available Effort in the month

Review Effectiveness
(No of Internal Review Defects / [No of Internal Defects+No of External Defects])*100
The purpose of this parameter is to measure how effective are our reviews in capturing all the phase injected defects.

Test Case Design Rework Index
(Test Cases with Review Comments for rework/Total Test Cases)*100

Types of Metrics – Productivity Metrics

The following are the Productivity Metrics

- Test case design productivity
- Test case execution productivity

Test case design productivity
Of Test cases (scripts) designed/ Total Test case design effort in hours
This metric indicates the productivity of the team in designing the test cases.

Test case execution productivity
Of Test cases executed/ Total Test case executed effort in hours
Effort shall include the set up time, execution time. This metric indicates the productivity of the team in executing the test cases.

Types of Metrics – Closure Metrics

The following are the Closure Metrics Effort distribution

- Test Design Review Effort
- Test Design Rework effort
- KM Effort

Effort Distribution
(Effort spent on a STLC Phase / Total STLC effort) * 100
This would indicate the effort distribution across STLC phases. Given the type of an engagement and application, we can re-use this in estimation & planning for future projects

Test Design Review effort
(Effort spent on Test Case design reviews / Total effort for spent on Test Case design) * 100

This can be used with other process metrics like "Review Effectiveness", "Defect removal Efficiency", "Defect Injection Ratio" to plan for an adequate review effort for future projects.

Test Design Rework effort
(Effort spent on Test Case design review rework / Total effort for spent on Test Case design) * 100

This can be used with other process metrics like "Effort Variance", "Schedule Variance" to plan for an adequate rework effort for future projects.

KM Effort
(Total Effort spent on preparation of the KM artifacts / Total actual effort for the project) * 100

This indicates the effort spent on KM.This along with other process and product metrics can be used to plan for KM activities for future projects.

Summary

In this lesson, you have learnt:

- Various testing metrics like
  - Defect Density
  - Average Defect Age
  - Defect Removal Efficiency
  - Review Effectiveness
  - Cost of finding a defect in review
  - Cost of finding a defect in testing
  - Cost of Quality
  - Test Case Effectiveness
  - Test Case Adequacy
  - Defect Detection Index

Review - Questions

Question 1: The defect impacting the functionality are _____.(Indirect/Direct/Standard)

Question 2: CFDR metric tells us the effort spent in finding a defect in testing

- Option: True / False

Question 3: Metrics are used to evaluate the effectiveness of the testing process

- Option: True / False

Review – Match the Following

1. Minimum	A. Average Defect Age
2. Maximum	B. Defect Removal Efficiency
	C. Cost of finding a defect in testing
	D. Review Effectiveness
	E. Cost of Quality
	F. Test Case Adequacy