

# Software Test Metrics

Software test metrics is to monitor and control process and product. It helps to drive the project towards our planned goals without deviation.

Software test metrics are classified into two types

1. Process metrics
2. Product metrics

## Process Metrics:

Software Test Metrics used in the process of test preparation and test execution phase of STLC.

The following are generated during the Test Preparation phase of STLC:

## Test Case Preparation Productivity:

It is used to calculate the number of Test Cases prepared and the effort spent for the preparation of Test Cases.

### ***Formula:***

Test Case Preparation Productivity = (No of Test Case)/ (Effort spent for Test Case Preparation)

*E.g.:*

No. of Test cases = 240

Effort spent for Test case preparation (in hours) = 10

Test Case preparation productivity =  $240/10 = 24$  test cases/hour

## Test Design Coverage:

It helps to measure the percentage of test case coverage against the number of requirements

### **Formula:**

Test Design Coverage = ((Total number of requirements mapped to test cases) / (Total number of requirements) \*100

*E.g.:*

Total number of requirements: 100

Total number of requirements mapped to test cases: 98

Test Design Coverage =  $(98/100) * 100 = 98\%$

The following are generated during the Test Execution phase of STLC:

## Test Execution Productivity:

It determines the number of Test Cases that can be executed per hour

### **Formula:**

(No of Test cases executed)/ (Effort spent for execution of test cases)

*E.g.:*

No of Test cases executed = 180

Effort spent for execution of test cases = 10

Test Execution Productivity =  $180/10 = 18$  test cases/hour

## Test Execution Coverage:

It is to measure the number of test cases executed against the number of test cases planned.

### **Formula:**

Test Execution Coverage = (Total no. of test cases executed / Total no. of test cases planned to execute) \* 100

*E.g.:*

Total no. of test cases planned to execute = 240

Total no. of test cases executed = 160

Test Execution Coverage =  $(160/240) * 100 = 66\%$

## Test Cases Passed:

It is to measure the percentage no. of test cases passed

### **Formula:**

Test Cases Pass = (Total no. of test cases passed) / (Total no. of test cases executed) \* 100

*E.g.:*

Test Cases Pass =  $(80/90) * 100 = 88.8 = 89\%$

## Test Cases Failed:

It is to measure the percentage no. of test cases failed

### ***Formula:***

Test Cases Failed = (Total no. of test cases failed) / (Total no. of test cases executed) \* 100

*E.g.:*

Test Cases Failed = (10/90) \* 100 = 11.1 = 11%

### **Test Cases Blocked:**

It is to measure the percentage no. of test cases blocked

### ***Formula:***

Test Cases Blocked = (Total no. of test cases blocked) / (Total no. of test cases executed) \* 100

*E.g.:*

Test Cases Blocked = (5/90) \* 100 = 5.5 = 6%

Check below video to see "Test Metrics"

## **Product metric:**

Software Test Metrics used in the process of defect analysis phase of STLC.

### **Error Discovery Rate:**

It is to determine the effectiveness of the test cases.

### ***Formula:***

Error Discovery Rate = (Total number of defects found / Total no. of test cases executed) \* 100

*E.g.:*

Total no. of test cases executed = 240

Total number of defects found = 60

Error Discovery Rate =  $(60/240) * 100 = 25\%$

## **Defect Fix Rate:**

It helps to know the quality of a build in terms of defect fixing.

### ***Formula:***

Defect Fix Rate =  $(\text{Total no of Defects reported as fixed} - \text{Total no. of defects reopened}) / (\text{Total no of Defects reported as fixed} + \text{Total no. of new Bugs due to fix}) * 100$

*E.g.:*

Total no of defects reported as fixed = 10

Total no. of defects reopened = 2

Total no. of new Bugs due to fix = 1

Defect Fix Rate =  $((10 - 2) / (10 + 1)) * 100 = (8/11)100 = 72.7 = 73\%$

## **Defect Density:**

It is defined as the ratio of defects to requirements.

Defect density determines the stability of the application.

### ***Formula:***

Defect Density =  $\text{Total no. of defects identified} / \text{Actual Size (requirements)}$

*E.g.:*

Total no. of defects identified = 80

Actual Size= 10

Defect Density =  $80/10 = 8$

## **Defect Leakage:**

It is used to review the efficiency of the testing process before UAT.

### ***Formula:***

Defect Leakage =  $((\text{Total no. of defects found in UAT}) / (\text{Total no. of defects found before UAT})) * 100$

*E.g.:*

No. of defects found in UAT = 20

No. of Defects found before UAT = 120

Defect Leakage =  $(20 / 120) * 100 = 16.6 = 17\%$

## **Defect Removal Efficiency:**

It allows us to compare the overall (defects found pre- and post-delivery) defect removal efficiency

### ***Formula:***

Defect Removal Efficiency =  $((\text{Total no. of defects found pre-delivery}) / ((\text{Total no. of defects found pre-delivery}) + (\text{Total no. of defects found post-delivery}))) * 100$

*E.g.:*

Total no. of defects found pre-delivery = 80

Total no. of defects found post-delivery = 10

Defect Removal Efficiency =  $((80) / ((80) + (10))) * 100 = (80/90) * 100 = 88.8 = 89\%$

-----**END**-----

Muntasir Abdullah Mizan  
muntasir.abdullah01@gmail.com