

# SOFTWARE METRICS

P. Nityasree  
17MIS1007

## DIGITAL ASSIGNMENT

① Suppose you want to assess the productivity of a software tester on your project. How would you define the tester's productivity, and what data would you collect? At what point in the development process would you collect the data?

A) The evaluation of the software tester can be done by validity of bugs that you reported. The quality of your writing as when the report issues and the no. of bugs that you have report and contributions to other testers bugs. For evaluating the tester's productivity the following parameters are included such as:

- 1) Number of scripts run,
- 2) Number of defects found,
- 3) Number of defects closed,
- 4) Number of defects reopened (severity)
- 5) Number of reported defects replicated
- 6) Number of hard to replicate yet critical/severe defects that can now be replicated
- 7) Number of issues/defects than can fixed.

And it also includes such as

- 1) What's exactly you are trying to accomplish in project
- 2) What are the ways to make testing



team effective, and more efficient, and also to spend less time on testing.

The productivity in metrics is calculated as

$$\text{Productivity (For test case execution)} = \frac{\text{Actual number of test cases}}{\text{actual effort expended in testing}}$$

$$\text{Productivity (defect detection)} = \frac{\text{Actual no. of defects (review + testing)}}{\text{actual effort spent on (review + testing)}}$$

The testers don't collect any data from the existing sources, but also they generate huge volumes of data to ensure the quality. As testers continuously explore, learn, and apply the most efficient approaches for data collection, generation, maintenance, automation and comprehensive data management.

The software tester productivity has measuring ongoing effectiveness/ performance such as bug reports,

→ designing, running, modifying the test cases

→ developing test strategies/plans

→ Editing technical documentation

→ writing support materials and facilitate the inspections

→ personal attributes as they effect the performance



## Editing bugs:

- Summary short and checking whether it is obvious where to start / replicate bug
- Is the replication sequence provided as a numbered set of steps
- Does the report include unnecessary info.
- Replicate the report and the summary accurately describe the failure
- Follow up tests as report relies on other specific values, with no indication about program
- Testers Evaluation as does the description include non-factual information.

## Quality measurement: process

- Risk analysis:  
What are the key risks, Are you testing risks,  
As optimizing your testing against risks

## Qualitative analysis:

- reviewing the specific artifacts of tester and performances as test cases, risk analysis  
Impacts scheduling

Data includes such as the No data, valid dataset, illegal data format, and also boundary condition format, dataset and the dataset for performance, load and stress testing  
the dataset is a Evaluation and Equivalence partition and dataset, decision table set, state transition dataset, usecase test dataset. and also the absent data.

End