**CHAVA SAI TEJA  
13BCE1032**

` **PROJECT METRICS**

Software process and project metrics are quantitative measures that enable software engineers to gain insight into the efficiency of the software process and the projects conducted using the process framework. In software project management, we are primarily concerned with productivity and quality metrics. There are four reasons for measuring software processes, products, and resources (to characterize, to evaluate, to predict, and to improve).

**What are they used for?**

* A software team can use software project metrics to adapt project workflow and technical activities.
* Project metrics are used to avoid development schedule delays, to mitigate potential risks, and to assess product quality on an on-going basis.
* Every project should measure its inputs (resources), outputs (deliverables), and results (effectiveness of deliverables).
* *Project metrics* enable project manager to
* Assess status of ongoing project
* Track potential risks
* Uncover problem are before they go critical
* Adjust work flow or tasks
* Evaluate the project team’s ability to control quality of software wrok products

**Project Metrics:-**

These are metrics that pertain to Project Quality. They are used to quantify defects, cost, schedule, productivity and estimation of various project resources and deliverables.

1. Schedule Variance  : Any difference between the scheduled completion of an activity and the actual completion is known as Schedule Variance.  
   
Schedule variance = ((Actual calendar days – Planned calendar days) + Start variance)/ Planned calendar days x 100  
  
2. Effort Variance: Difference between the planned outlined effort and the effort required to actually undertake the task is called Effort variance.  
   
Effort variance = (Actual Effort – Planned Effort)/ Planned Effort x 100  
  
3. Size Variance: Difference between the estimated size of the project and the actual size of the project (normally in KLOC or FP).  
   
Size variance = (Actual size – Estimated size)/ Estimated size x 100  
  
4. Requirement Stability Index: Provides visibility to the magnitude and impact of requirements changes.  
   
RSI = 1- ((No of changed + No of deleted + No of added) / Total no of Initial requirements) x100.  
  
5. Productivity (Project): It is a measure of output from a related process for a unit of input.  
   
Project Productivity = Actual Project Size / Actual effort expended in the project  
  
6. Productivity (for test case preparation) = Actual no of test cases/ Actual effort expended in test case preparation.  
  
7. Productivity (for test case execution) = Actual number of test cases / actual effort expended in testing.  
  
8. Productivity (defect detection) = Actual number of defects (review + testing) / actual effort spent on (review + testing).

9. Productivity (defect fixation) = actual no of defects fixed/ actual effort spent on defect fixation.  
  
10. Schedule variance for a phase: The deviation between planned and actual schedules for the phases within a project.  
   
Schedule variance for a phase = (Actual Calendar days for a phase – Planned calendar days for a phase + Start variance for a phase)/ (Planned calendar days for a phase) x 100  
  
11. Effort variance for a phase: The deviation between planned and actual effort for various phases within the project.  
   
Effort variance for a phase = (Actual effort for a phase – planned effort for a phase)/ (planned effort for a phase) x 100.

**WebApp Project Metrics**

* Number of static Web pages (Nsp)
* Number of dynamic Web pages (Ndp)
* Customization index: C = Nsp / (Ndp + Nsp)
* Number of internal page links
* Number of persistent data objects
* Number of external systems interfaced
* Number of static content objects
* Number of dynamic content objects
* Number of executable functions

**Process Metrics:-**

1. Cost of quality: It is a measure of the performance of quality initiatives in an organization. It’s expressed in monetary terms.  
   
Cost of quality = (review + testing + verification review + verification testing + QA + configuration management + measurement + training + rework review + rework testing)/ total effort x 100.  
  
2. Cost of poor quality: It is the cost of implementing imperfect processes and products.  
Cost of poor quality = rework effort/ total effort x 100.  
  
3. Defect density: It is the number of defects detected in the software during development divided by the size of the software (typically in KLOC or FP)  
   
Defect density for a project = Total number of defects/ project size in KLOC or FP  
  
4. Review efficiency: defined as the efficiency in harnessing/ detecting review defects in the verification stage.  
   
Review efficiency = (number of defects caught in review)/ total number of defects caught) x 100  
  
5. Testing Efficiency: Testing efficiency = 1 – ((defects found in acceptance)/ total no. of testing defects) x 100  
  
6. Defect removal efficiency: Quantifies the efficiency with which defects were detected and prevented from reaching the customer.  
   
Defect removal efficiency = (1 – (total defects caught by customer/ total no of defects)) x 100  
  
7. Residual defect density = (total number of defects found by customer)/ (Total number of defects including customer found defects) x 100