

**Paper Name: Software Engineering**

**Topic: Software Metrics**

**Paper Code: BCA CC409**

**Semester IV**

**By Ms. Amrita Prakash**

**Assistant Professor,**

**Department of Computer Science**

**Patna Women's College**

**E-mail: [amrita.bca@patnawomenscollege.in](mailto:amrita.bca@patnawomenscollege.in)**

## Software Metrics

- A metric is a derived unit of measurement that cannot be directly observed, but is created by combining two or more measures.
- An ideal metrics should be simple and precisely defined, easily obtainable, valid and robust.

### **Software Measurement**

Software measurement is used to assess the quality of the end product or engineered product.

These measures are collected throughout the Software Development Lifecycle with an intention to improve the software process on a continuous basis.

Measurement helps in estimation, quality control, productivity assessment and project control throughout a Software project.

Software measurement are of two categories:-

1. Direct Measures and
2. Indirect Measures

1. **Direct Measures** include software processes like cost and effort applied and products like lines of code produced, execution speed and other defects that have been reported.
2. **Indirect Measures** include products like functionality, quality, complexity, reliability, maintainability and much more.

### **Measurement process is Characterized by a set of five activities:-**

1. Formulation:- This performs measurement and develops appropriate metrics for software under consideration.
2. Collection:- This collects data to derive the formulated metrics.
3. Analysis:- This calculates metrics and the use of mathematical tools.
4. Interpretation:- This analyzes the metrics to attain insight into the quality of representation.

5. Feedback:- This communicates recommendation derived from product metrics to the software team.

In order to perform these activities effectively, it is recommended to automate data collection and analysis, establish guidelines and recommendations for each metric, and use statistical techniques to interrelate external quality features and internal product attributes.

## Software Metrics

Once measures are collected they are converted for metrics for use. IEEE defines metric as “a quantitative measure of the degree which a system, component or process possesses a given attribute’.

The goal of software metrics is to identify and control essential parameters that affect software development.

### **Objectives of using Software Metrics:**

- Measuring the size of the software quantitatively.
- Assessing the level of complexity involved.
- Assessing the strength of the module by measuring coupling.
- Assessing the testing techniques.
- Specifying when to stop testing.
- Determining the data of release of the software.
- Estimating cost of resources and project schedule.

To achieve these objectives, software metrics are applied to different projects for long period of time to obtain indicators.

For example, schedule – based resource allocation can be effectively enhanced with the help of metrics.

Metrics is often used interchangeably with measure and measurement.

**Measure** can be defined as quantitative indication of amount, dimension, capacity or size of product and process attributes.

**Measurement** can be defined as the process of ascertaining the measure.

**Metrics** can be defined as quantitative measures that allow software engineers to identify the efficacy and improve the quality of software process, project and product.

Once measures and metrics have been developed, indicators are obtained. These indicators provide a detailed insight into the software process, software project or intermediate product.

How are metrics used?

- Measures are often collected by SW engineers and used by SW managers.
- Measurements are analyzed and compared to past measurements, for similar projects, to look for trends (good and bad) and to make better estimates.

Three domains of SW metrics

- Product – – These measurements relate to SW product and all related artifacts.
- Examples: code, design docs, test plan, user manual ...LOC,

# of objects, # of pages, # of files. – Measures can also be used to evaluate the SW quality:

- Cyclomatic complexity: a way to measure the complexity of a module.
  - It assigns a value  $V(G)$  to a module based on the control flow of the module. Some companies place a cap on  $V(G)$ . If too high, the module must be redesigned.
- Process: - – These measures used to quantify characteristics of the SW process. – Usually related to events or things that occur. – Examples: # defects found in test, # requirements changes, # days to complete task ...
- Project – – used to manage the SW project “Tactic”. – Estimating cost is the first application of Project Metrics – Examples: estimates of SW development time based on past projects.

## Advantage of Software Metrics

- Comparative study of various design methodology of software systems.
- For analysis, comparison, and critical study of different programming language concerning their characteristics.
- In comparing and evaluating the capabilities and productivity of people involved in software development.
- In the preparation of software quality specifications.
- In the verification of compliance of software systems requirements and specifications.
- In making inference about the effort to be put in the design and development of the software systems.
- In getting an idea about the complexity of the code.
- In taking decisions regarding further division of a complex module is to be done or not.
- In guiding resource manager for their proper utilization.
- In comparison and making design tradeoffs between software development and maintenance cost.
- In providing feedback to software managers about the progress and quality during various phases of the software development life cycle.
- In the allocation of testing resources for testing the code.

## Disadvantage of Software Metrics

- The application of software metrics is not always easy, and in some cases, it is difficult and costly.
- The verification and justification of software metrics are based on historical/empirical data whose validity is difficult to verify.
- These are useful for managing software products but not for evaluating the performance of the technical staff.
- The definition and derivation of Software metrics are usually based on assuming which are not standardized and may depend upon tools available and working environment.
- Most of the predictive models rely on estimates of certain variables which are often not known precisely.