

# Metrics and their scope

In today's world, virtually everything has a measurement unit, at least “everything” that can be measured. And so does a software. The software can be measured on the basis of its quality, and the thing used for this “software measurement” are called “metrics”.

Some wise people, in the computer world and elsewhere, have defined measurement as follows:

- "Measurement is the assignment of numbers to objects or events according to rule. The rule of assignment can be any consistent rule. The only rule not allowed would be random assignment, for randomness amounts in effect to a nonrule."
- "Measurement is the process of empirical, objective, assignment of numbers to properties of objects or events of the real world in such a way as to describe them."
- "Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to characterize them according to clearly defined rules."
- Measurement is "the act or process of assigning a number or category to an entity to describe an attribute of that entity."

Now, I'll try to define a metric in the best possible way:

IEEE Standard 1061 lays out a methodology for developing metrics for software quality attributes. The standard defines an *attribute* as "a measurable physical or abstract property of an entity." A *quality factor* is a type of attribute, "a management-oriented attribute of software that contributes to its quality." A metric is a measurement function, and a software quality metric is "a function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality."

## What is the purpose of this measure?

Examples of purposes include:

- facilitating private self-assessment and improvement.
- evaluating project status (to facilitate management of the project or related projects)
- evaluating staff performance
- informing others (e.g. potential customers) about the
- characteristics (such as development status or behavior)

### **What is the scope of this measure?**

A few examples of the scope of metrics are:

- a single method from one person
- one project done by one workgroup
- a year's work from that workgroup
- the entire company's output (including remote locations) for the last decade

As the scope broadens, more confounding variables can come into play, potentially impacting or invalidating the metric. A metric that works well locally might fail globally.

### **Types of metrics**

Some different types of metrics include:

- Analysis metrics
- Architectural metrics
- Component metrics
- Design metrics
- Maintenance metrics
- Specification metrics
- Source Code metrics
- Testing metrics

## **References**

[1] Software Engineering - A Practitioner's Approach || Roger S. Pressman (5<sup>th</sup> edition)  
McGraw -Hill International Edition  
(Pg. 507)

[2] Software Engineering Metrics: What Do They Measure and How Do We Know? ||  
Cen Kaner and Walter P. Bond  
(<http://www.kaner.com/pdfs/metrics2004.pdf> )

Anurag Bhandari  
05103007  
G1, CSE,  
6<sup>th</sup> Semester