**Software Metrics**

A software metric is a measure of software characteristics which are measurable or countable. Software metrics are valuable for many reasons, including measuring software performance, planning work items, measuring productivity, and many other uses.

Measurement can be used throughout a software project to assist in

* estimation,
* quality control,
* productivity assessment, and
* project control.

Finally, measurement can be used by software engineers to help assess the quality of technical work products and to assist in tactical decision making as a project proceeds

**What guidelines should be applied when we collect software metrics?**

* Use common sense and organizational sensitivity when interpreting metrics data.
* Provide regular feedback to the individuals and teams who collect measures and metrics.
* Don’t use metrics to appraise individuals.
* Work with practitioners and teams to set clear goals and metrics that will be used to achieve them.
* Never use metrics to threaten individuals or teams.
* Metrics data that indicate a problem area should not be considered “negative.” These data are merely an indicator for process improvement.
* Don’t obsess on a single metric to the exclusion of other important metrics.

**Project Metrics**

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

**SOFTWARE MEASUREMENT**

* ***Direct measures***of the software engineering process include cost and effort applied. Direct measures of the product include lines of code (LOC) produced, execution speed, memory size, and defects reported over some set period of time.
* ***Indirect measures***of the product include functionality, quality, complexity, efficiency, reliability, maintainability, and many other "–abilities"

**ESTABLISHING A SOFTWARE METRICS PROGRAM**

The Software Engineering Institute has developed a comprehensive guidebook [PAR96] for establishing a “goal-driven” software metrics program. The guidebook suggests the following steps:

**1.** Identify your business goals.

**2.** Identify what you want to know or learn.

**3.** Identify your sub goals.

**4.** Identify the entities and attributes related to your subgoals.

**5.** Formalize your measurement goals.

**6.** Identify quantifiable questions and the related indicators that you will use to

help you achieve your measurement goals.

**7.** Identify the data elements that you will collect to construct the indicators that

help answer your questions.

**8.** Define the measures to be used, and make these definitions operational.

**9.** Identify the actions that you will take to implement the measures.

**10.** Prepare a plan for implementing the measures.