

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.
- Within the software development process, many metrics are that are all connected. Software metrics are similar to the four functions of management: Planning, Organization, Control, or Improvement.

Classification of Software Metrics

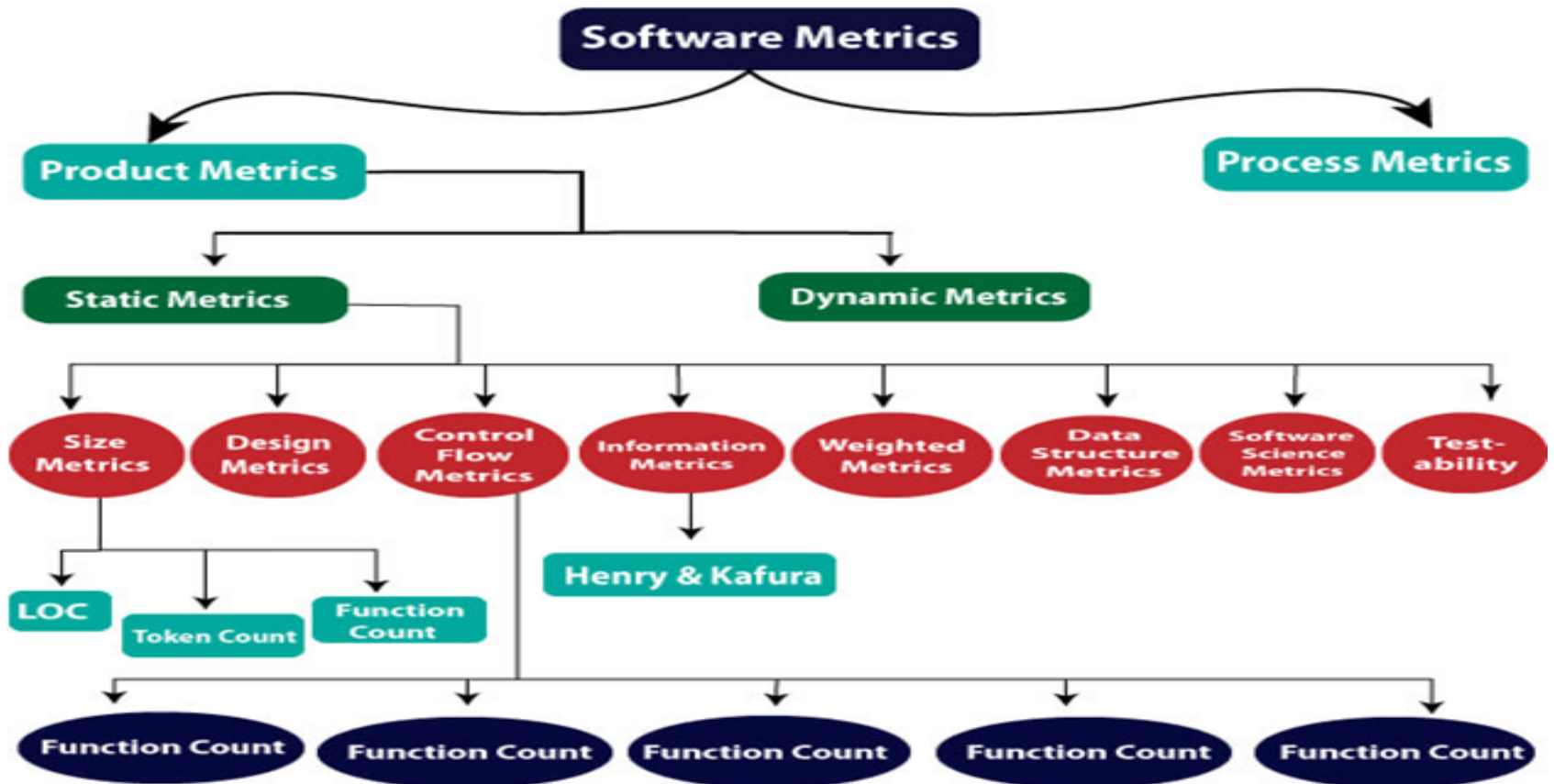
- **Software metrics can be classified into two types as follows:**
- **1. Product Metrics:** These are the measures of various characteristics of the software product. The two important software characteristics are:
 - Size and complexity of software.
 - Quality and reliability of software.

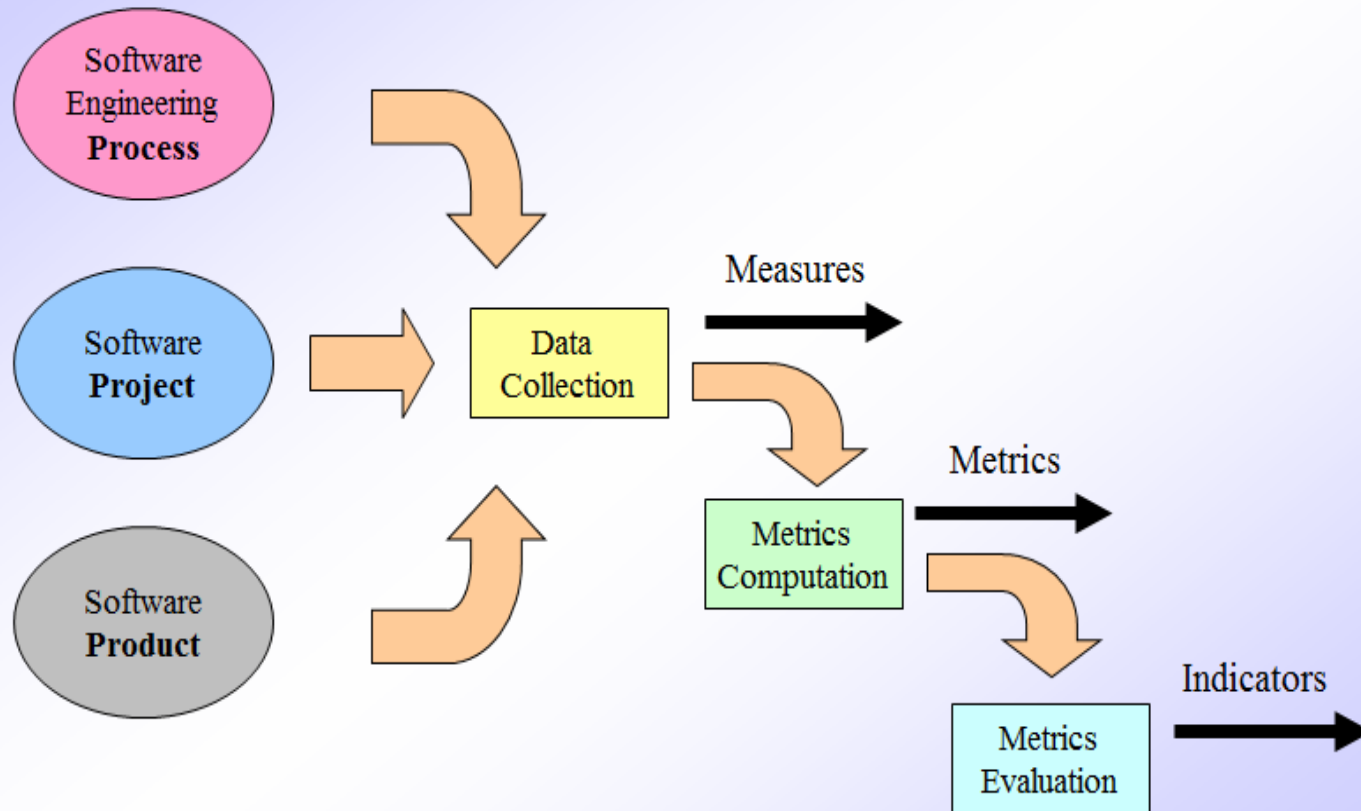
These metrics can be computed for different stages of SDLC.

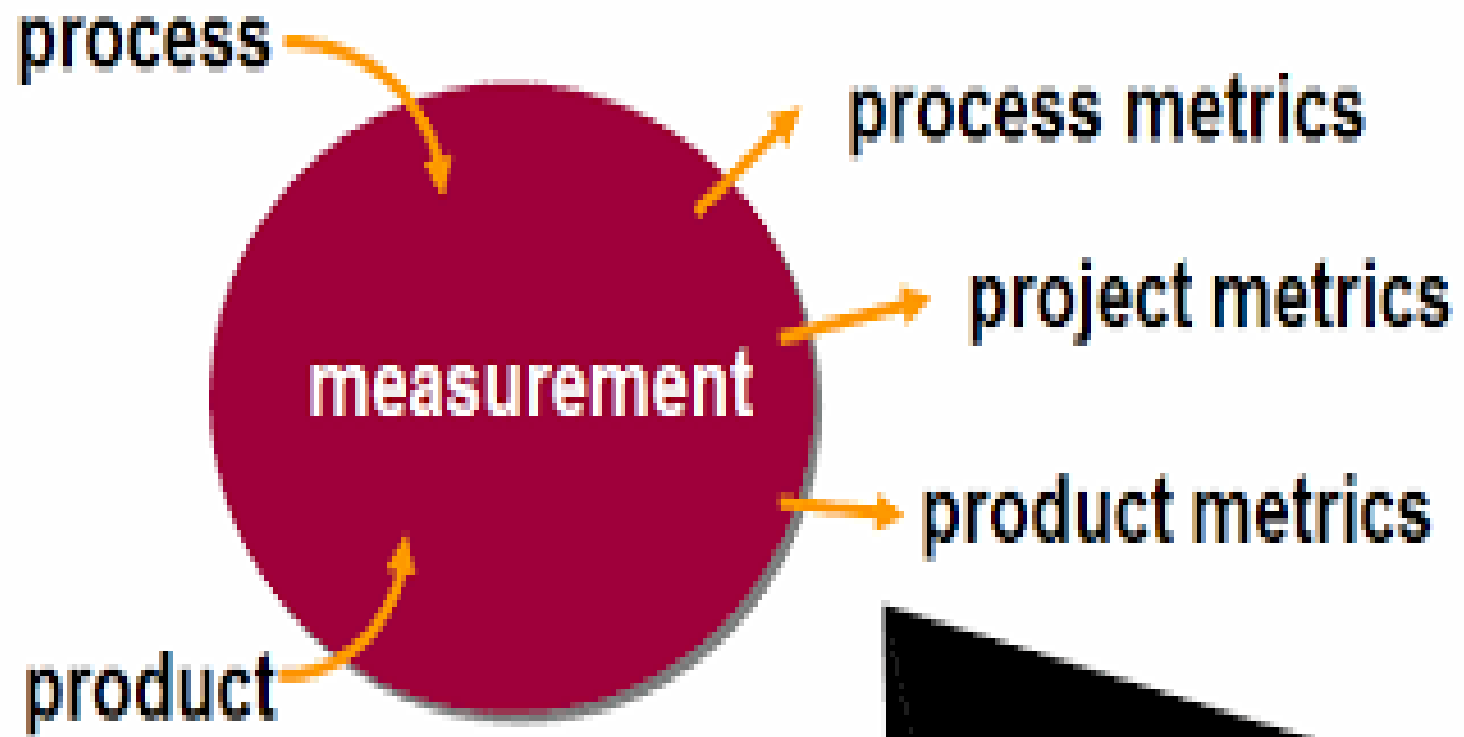
- **2. Process Metrics:** These are the measures of various characteristics of the software development process. For example, the efficiency of fault detection. They are used to measure the characteristics of methods, techniques, and tools that are used for developing software.

-

Classification of Software Metrics







Types of Metrics

- **Internal metrics:** Internal metrics are the metrics used for measuring properties that are viewed to be of greater importance to a software developer. For example, Lines of Code (LOC) measure.
- **External metrics:** External metrics are the metrics used for measuring properties that are viewed to be of greater importance to the user, e.g., portability, reliability, functionality, usability, etc.

- **Hybrid metrics:** Hybrid metrics are the metrics that combine product, process, and resource metrics. For example, cost per FP where FP stands for Function Point Metric.

- **Project metrics:** Project metrics are the metrics used by the project manager to check the project's progress. Data from the past projects are used to collect various metrics, like time and cost; these estimates are used as a base of new software. Note that as the project proceeds, the project manager will check its progress from time-to-time and will compare the effort, cost, and time with the original effort, cost and time. Also understand that these metrics are used to decrease the development costs, time efforts and risks. The project quality can also be improved. As quality improves, the number of errors and time, as well as cost required, is also reduced

- **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.

- **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.

Size Oriented Metrics

- LOC Metrics
- It is one of the earliest and simpler metrics for calculating the size of the computer program. It is generally used in calculating and comparing the productivity of programmers. These metrics are derived by normalizing the quality and productivity measures by considering the size of the product as a metric.

Following are the points regarding LOC measures:

- In size-oriented metrics, LOC is considered to be the normalization value.
- It is an older method that was developed when FORTRAN and COBOL programming were very popular.
- Productivity is defined as $KLOC / EFFORT$, where effort is measured in person-months.
- Size-oriented metrics depend on the programming language used.
- As productivity depends on KLOC, so assembly language code will have more productivity.

- **Advantages of LOC**
- Simple to measure
- **Disadvantage of LOC**
- It is defined on the code. For example, it cannot measure the size of the specification.
- It characterizes only one specific view of size, namely length, it takes no account of functionality or complexity
- Bad software design may cause an excessive line of code
- It is language dependent
- Users cannot easily understand it