

PRACTICAL : 6

Aim:

Background: There are some metrics which are fundamental and the rest can be derived from these. Examples of basic (fundamental) measures are size, effort, defect, and schedule. If the fundamental measures are known, then we can derive others. For example, if size and effort are known, we can get Productivity ($=\text{size}/\text{effort}$). If the total numbers of defects are known, we can get the Quality ($=\text{defect}/\text{size}$) and so on.

Problem Description: Online loan system has two modules for the two basic services, namely Carloan service and House loan service. The two modules have been named as Car_Loan_Module and House_Loan_Module. Car_Loan_Module has 2000 lines of uncommented source code. House_Loan_Module has 3000 lines of uncommented source code. Car_Loan_Module was completely implemented by Mike. House_Loan_Module was completely implemented by John. Mike took 100 person hours to implement Car_Loan_Module. John took 200 person hours to implement House_Loan_Module. Mike's module had 5 defects. John's module had 6 defects. With respect to the context given, which among the following is an INCORRECT statement?

Choose One:

1. John's quality is better than Mike.
2. John's productivity is more than Mike.
3. John introduced more defects than Mike.
4. John's effort is more than Mike

Software required: NONE.

Theory:

Software Measurement:

A measurement is a manifestation of the size, quantity, amount, or dimension of a particular attributes of a product or process. Software measurement is a titrate impute of a characteristic of a software product or the software process. It is an authority within software engineering. Software measurement process is defined and governed by ISO Standard.

Need of Software Measurement:

Software is measured to:

1. Create the quality of the current product or process.
2. Anticipate future qualities of the product or process.
3. Enhance the quality of a product or process.
4. Regulate the state of the project in relation to budget and schedule.

Classification of Software Measurement:

There are 2 types of software measurement:

1. Direct Measurement: In direct measurement the product, process or thing is measured directly using standard scale.
2. Indirect Measurement: In indirect measurement the quantity or quality to be measured is measured using related parameter i.e. by use of reference.

Metrics

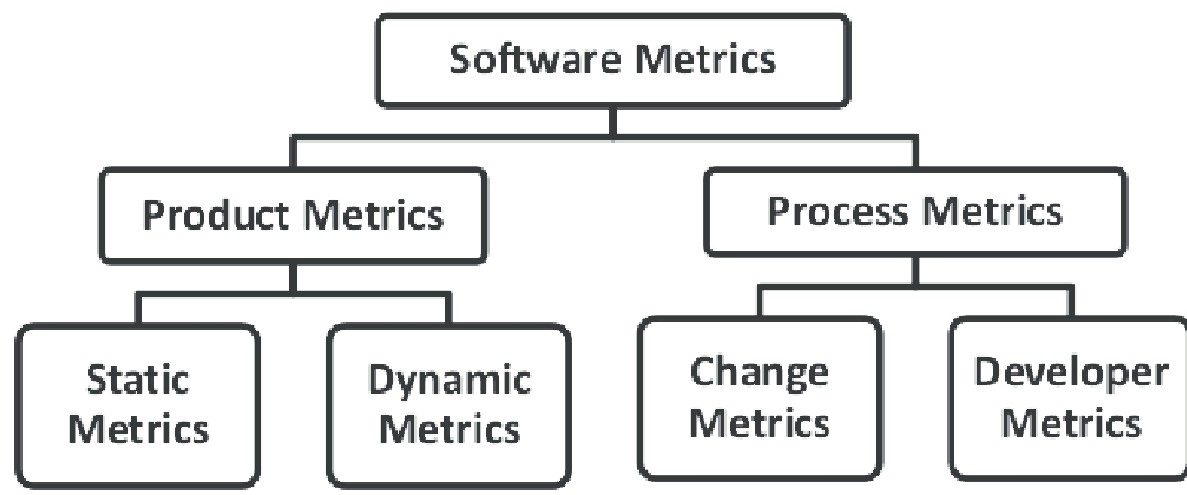
A metrics is a measurement of the level that any impute belongs to a system product or process.

There are 4 functions related to software metrics:

1. Planning
2. Organizing
3. Controlling
4. Improving

Characteristics of software Metrics:

1. Quantitative: Metrics must possess quantitative nature. It means metrics can be expressed in values. Understandable: Metric computation should be easily understood, the method of computing metric should be clearly defined.
2. Applicability: Metrics should be applicable in the initial phases of development of the software.
3. Repeatable: The metric values should be same when measured repeatedly and consistent in nature.
4. Economical: Computation of metric should be economical.
5. Language Independent: Metrics should not depend on any programming language.



Classification of Software Metrics:

There are 2 types of software metrics:

1. **Product Metrics:** Product metrics are used to evaluate the state of the product, tracing risks and under covering prospective problem areas. The ability of team to control quality is evaluated.
2. **Process Metrics:** Process metrics pay particular attention on enhancing the long-term process of the team or organization.
3. **Project Metrics:** Project matrix is describing the project characteristic and execution process.
 - Number of software developer.
 - Staffing pattern over the life cycle of software.
 - Cost and schedule.
 - Productivity.

Background:

There are some metrics which are fundamental, and the rest can be derived from these. Examples of basic (fundamental) measures are size, effort, defect, and schedule. If the fundamental measures are known, then we can derive others. For example, if size and effort are known, we can get Productivity ($=\text{size}/\text{effort}$). If the total numbers of defects are known, we can get the Quality ($=\text{defect}/\text{size}$) and soon.

Problem Description:

Online loan system has two modules for the two basic services, namely Car loan service and House loan service. The two modules have been named as Car_Loan_Module and House_Loan_Module. Car_Loan_Module has 2000 lines of uncommented source code. House_Loan_Module has 3000 lines of uncommented source code. Car_Loan_Module was completely implemented by Mike. House_Loan_Module was completely implemented by John. Mike took 100 person hours to implement Car_Loan_Module. John took 200 person hours to implement House_Loan_Module. Mike's module had 5 defects. John's module had 6 defects. With respect to the context given, which among the following is an INCORRECT statement?

Choose One:

1. John's quality is better than Mike.
2. John's productivity is more than Mike.
3. John introduced more defects than Mike.
4. John's effort is more than Mike.

Ans: 3. John introduced more defects than Mike.

Justification:

Here, Car_Loan_Module was completely implemented by Mike and House_Loan_Module was completely implemented by John. House_Loan_Module has more lines of uncommented source code than Car_Loan_Module. We can see that Mike's module had 5 defects while John's module had 6 defects. So, it does not mean that John introduced more defects than Mike; therefore it is an Incorrect Statement.