

Assignment

Student Name: Vivek Kumar

Branch: BE-CSE (LEET)

Semester: 4th

Subject Name: Computer Network

UID: 21BCS8129

Section/Group: 809/A

Date of Performance: 06/03/2022

Subject Code: 20CSP-256

1. Aim/Overview of the practical:

Describe the software metrics for analysis and design models.

Theories:

The standard of measure for the estimation of quality, progress and health of the software testing effort is called software metrics. It can be divided into three groups: product metrics, process metrics, and project metrics. The product characteristics like size, features of the design, complexity, performance, level of quality, etc., is described using product metrics. In contrast, software development and maintenance are improved using process metrics. The project's characteristics and execution are described by project metrics whose examples include the count of software developers, cost, etc.

It is necessary to develop software metrics based on some guidelines. Those guidelines are:

It must be simple and computable. The derivation must be easy to learn, and the time and effort involved must be average.

The results given must be objective and consistent. The results should not be ambiguous.

It must make use of units and dimensions if there are mathematical computations involved.

The development of metrics should be based on an analysis model, design model or structure of the model, and it should be independent of the programming language.

It is effective if and only if it can deliver high-quality software products.

It must be able to adapt to the changing requirements of the project, which is calibration must be easy.

The cost of developing the metrics must be reasonable. One must be able to obtain it easily.

If we are using software metrics for making any decisions, it must be validated before being applied to make decisions.

The developed metrics must be robust to changes that it should not be sensitive to changes in project, process, or small product.

As and when the value of the software characteristics represented by its changes, the value must also change and for this to happen, the range must be in a meaningful range. Let us say, for example, the range of software metrics is zero to five.

Types of Software Metrics

1. Process Metrics

Process metrics are used to measure the characteristics of the process of software development. The example includes the efficiency of detection of fault etc. The characteristics of the methods, tools, and techniques used for software development can be measured using process metrics.

2. Product Metrics

The characteristics of the software product are measured using product metrics. Some of the important characteristics of the software are:

Software size and complexity

Software reliability and quality

Computation of these metrics is done for different stages of the software development lifecycle.

3. Internal Metrics

The properties which are of great importance to a software developer can be measured using the metrics called internal metrics. An example is a measure of Lines of code (LOC).

4. External Metrics

The properties which are of great importance to a user can be measured using the metrics called external metrics. An example is portability, reliability, usability, etc.

5. Project Metrics

The progress of the project is checked by the project manager using the metrics called project metrics. Various metrics such as time, cost, etc., are collected by using the data from the projects in the past, and they are used as an estimate for the new software. The project manager checks the progress of the project from time to time, and effort, time and cost are compared with the original effort, time and cost. The cost of development, efforts, risks and time can be reduced by using these metrics. The quality of the project can also be improved. With the increase in quality, there is a reduction in the number of errors, time, cost, etc.

Advantages

The design methodology of the software systems can be studied comparatively.

The characteristics of various programming languages can be studied for analysis and comparison using software metrics.

The software quality specifications can be prepared using software metrics.

The compliance of requirements and specifications of software systems can be verified.

The effort that needs to be put into the development and design of software systems can be inferred.

The complexity of the code can be determined.

The decision of whether to divide a complex module or not can be done.

The utilization of resource managers to their fullest can be guided.

Design trade-offs and comparing maintenance costs and software development costs can be done.

The progress and quality of different phases of the software development life cycle can be measured, and feedback can be given to the project managers using software metrics.

The allocation of resources to test the code can be done based on software metrics.

Disadvantages

It is not easy to apply metrics in all cases. It is difficult and expensive in some cases.

It is difficult to verify the validity of historical or empirical data on which the verification and justification.

Software products can be managed, but the technical staff's performance cannot be evaluated using software metrics.

The available tools and the working environment is used to define and derive the software metrics, and there is no standard in defining and deriving them.

Certain variables are estimated based on the predictive models, and they are not known so often.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			