

A Review on Role of Software Metrics in Software Quality

Pavithra S R¹ Shanmuga Priya. N²

¹MCA Student ²Assistant Professor

^{1,2}Department of Master of Computer Applications

^{1,2}Dr. SNS Rajalakshmi College of Arts and Science, Coimbatore, Tamil Nadu, India - 641 049, India

Abstract— Software metrics provides a quantitative basis for planning and predicting software development processes. The quality of the software can be controlled and improved easily. Quality in fact aids higher productivity, which has brought software metrics to the forefront. Many metrics and models have been developed to promote and utilized resulting in remarkable successes. It examines the area of software engineering to see why software metrics are needed and also reviews their contribution to software quality and reliability. Results can be improved further as we acquire additional experience with variety of software metrics. Size measures include source lines-of-code, function points, and feature points. Complexity is a function of size which impacts design errors and latent defects, ultimately resulting in quality problems, cost overruns, and schedule slips. Complexity must be continuously measured, tracked, and controlled. Software metrics measure different aspects of software complexity and therefore play an important role in analyzing and improving software quality. It has indicated that they provide useful information on external quality aspects of software such as its maintainability, reusability and reliability. Software metrics provide a mean of estimating the efforts needed for testing. Software metrics are often categorized into products and process metrics.

Key words: Software Metrics, Software Quality, Software Reliability, Lines of Code, Function Points, Object Oriented Metrics

I. INTRODUCTION

The Software metrics are valuable entity in the entire software life cycle. They provide measurement for the software development, including software requirement documents, designs, programs and tests. Rapid developments of large scaled software have evolved complexity that makes the quality difficult to control. The execution of the control over software quality requires software metrics. The software metrics is to be understandable and established. Many metrics related to the product quality have been developed and used. It is essential to introduce definition of software metrics. Software metrics provides size of the software product and the process of software production. The software product begins from an initial statement of requirement to a finished software product, including source and object code and the several forms of documentation exhibited during the various stages of its development. [6]The metrics should enable the development of models that are efficient of predicting process or product spectrum.

- Objective, to the greatest extent possible.
- Easily obtainable that is reasonable cost.

II. CLASSIFICATION OF SOFTWARE METRICS

The Classification of software metrics is processed in three types as follows: [9]

- Process Metrics
- Project Metrics
- Product Metrics

A. Process Metrics

Process metrics are also called as management metrics and used to measure the properties of the process which is used to obtain the software. It include the cost metrics, efforts metrics, and advancement metrics and reuse metrics and help in predicting the size of final system to determine whether a project on running according to the schedule.

1) Effort required in the process

- Time to produce the product
- Effectiveness of defect removal during development
- Number of defects found during testing
- Maturity of the process

B. Project Metrics

Project metrics are used to monitor project situation and status. It preclude the problems or potential risks by calibrating the project and help to optimize the software development plan and describes the project characteristics and execution.

C. Product Metrics

Product metrics describe the attributes of the software product at any phase of its development. [2] It may measure the size of the program, complexity of the software design, performance, portability, maintainability, and product scale. Product metrics are used to presume and invent the quality of the product and used to measure the medium or the final product.

III. BASIC RELIABILITY METRICS

Reliability metrics are used to quantitatively convey the reliability of the software product. The metric is to be used depends upon the type of system to which it applies and the requirements of the application domain. It is difficult to find a way to measure software reliability, and most of the aspects related to software reliability. The software sizes have no uniform definition and cannot measure the reliability directly, something can be measured that reflects the characteristics related to reliability. Test coverage metric estimate fault and reliability by performing tests on software products, assuming that software reliability is a function of the portion of software that is successfully verified or tested.[4]

- Complexity is directly related to software reliability representing complexity is important. Complexity-oriented metrics is used to determine the complexity of a program's control structure, by simplifying the code into a graphical representation.
- Quality metrics measures the quality at various stages of software product development. An important quality metric is defect removal efficiency (DRE). DRE

provides a measure of quality because of various quality assurance and control activities applied throughout the development process.

IV. SOFTWARE METRICS FOR RELIABILITY

The Metrics are used to improve the reliability of the system by identifying the areas of requirements. The different types of software metrics that are used is as follows:[5]

- Requirement Reliability Metric
- Design and Code Reliability Metric
- Testing Reliability Metric

A. Requirement Reliability Metric

Requirements indicate what features the software must contain. It specify the functionality that must be included in the software. The requirements must be written such that is no misunderstanding between the developer and the client.[10] The requirements must contain valid structure to avoid the loss of valuable information. The requirements should be thorough and in a detailed manner so that it is easy for the design phase. The requirements should not contain inadequate information. Requirement Reliability metrics evaluates the above said quality factors of the required document.

B. Design and Code Reliability Metric

The quality factors that exists in design and coding plan are complexity, size and modularity. Complex modules are difficult to understand and there is high probability of occurring errors.[8] The reliability will decrease if modules have a combination of high complexity and large size or high complexity and small size. These metrics are also applicable to object oriented code, but in this, additional metrics are required to evaluate the quality.

C. Testing Reliability Metric

Testing the reliability metric ensures that the system is equipped with the functions that are specified in the requirements. The errors due to the lack of functionality decreases and approach is evaluating the code, finding the errors and fixing them. To ensure that the system contains the functionality specified test plans are written that contain multiple test cases. Each test case is based on one system state and tests some functions that are based on a related set of requirements. The objective of an effective verification program is to ensure that every requirement is tested. The implication being that if the system passes the test and the requirement's functionality is included in the delivered system.

V. MEASUREMENT OF METRICS

The Measurement of metrics is indicated with size, quantity, amount or dimension of a particular attribute of a process in three steps as follows:[7]

- Line of Code
- Token Count
- Function Point Metrics

A. Line of Code

It is one of the earliest and simpler metrics for calculating the size of computer program. It is generally used in calculating and comparing the productivity of programmers.

- Productivity is measured as LOC and man-month.
- The any line of program text excluding comment or blank line, regardless of the number of statements or parts of statements on the line, is considered a Line of Code.

B. Token Count

The metrics, a computer program is considered to be a collection of tokens, which may be classified as either operators or operands. Software metrics can be defined in terms of these basic symbols. These symbols are called as token.

C. Function Point Metrics

The function point analysis to measure software application is enumerated from analysis of the requirements and logical design of the application. The points are counted and can be applied to Development projects, Enhancement projects, and existing applications as well. The key elements of function point analysis are used to capture the functionality of the application.

VI. BENEFIT OF SOFTWARE METRICS

- The Comparative study of various design methodology of software systems. For analysis, comparison and critical study of various programming language with respect to their characteristics.[1]
- The comparing and evaluating capabilities and productivity of people involved in software development.
- The preparation of software quality specifications.
- It verification of compliance of software systems requirements and specifications.
- The making inference about the effort to be put in the design and development of the software systems.
- The getting an idea about the complexity of the code.
- The taking decisions regarding further division of complex module is to be done or not.
- It providing guidance to resource manager for their proper utilization.
- It comparison and making design tradeoffs between software development and maintenance cost.
- Its providing feedback to software managers about the progress and quality during various phases of software development life cycle.
- It allocation of testing resources for testing the code.

VII. COMMON USES OF METRICS

Software metrics are used to objective reproducible measurements that can be useful for quality assurance, performance, debugging, management, and estimating costs. [3]

- To find defects in code, predicts defective code, predicts project success, and predicts project risk.
- There is still some debate around which metrics matter and what they mean, the utility of metrics is limited to quantifying one of the following goals are Schedule of a software project, Size and complexity of development involved, cost of project and quality of software.

VIII. CONCLUSION

The Software metrics program that is based on the goals of an organization will help communicate, measure progress towards, and eventually accomplish those goals. The people will work to accomplish what they believe to be important. The designed metrics with documented objectives can help an organization obtain the information it needs to continue to improve its software products, processes, and services while maintaining a focus on what is important. The practical, systematic, start-to-finish method of selecting, designing and implementing software metrics is a valuable aid. The Better measures and better metrics are the stepping stones to software engineering excellence. The report highlights both measurement problems and also increase the usage of effective metrics such as function points and defect removal efficiency. The concept of software metric in different application like to measure the performance of task scheduling algorithm, to access the quality of the software, to test the functionality of the software. This paper we can study different type of software metrics which are used during the software development.

REFERENCES

- [1] Prabhjot Kaur, "A Review of Software Metric and Measurement", July 2016.
- [2] Kunal Chopra, Monika Sachdeva. 2015. "Evaluation of Software Metrics For Software Projects". *International Journal of Computers & Technology*. Volume 14, No. 6, May -June, 2015.
- [3] Ramesh Kumar, Dr.Rajesh Verma, "Software Quality Metrics: Concept and Significance", March 2014.
- [4] Gurpreet Kaur, Kailash Bahl, "Software Reliability, Metrics, Reliability Improvement Using Agile Process", May 2014.
- [5] Ritika Wasen, P.Ahemed, M.Qasim Rafiq "New papadigm for software reliability estimation "Volume 44, No. 14, April 2012.
- [6] Mrinal Singh Rawat, Arpita Mittal, Sanjay Kumar Dubey, "Survey on Impact of Software Metrics on Software Quality", 2012.
- [7] Gurdev Singh, Dilbag Singh, Vikram Singh, "A Study of Software Metrics", January 2011.
- [8] Misirli, A.Caglayan, B.Miransky, A. Bener, A., Ruffolo, N. (2011). Different Strokes for Different Folks: A Case Study on Software Metris for Different Defect Categories. WETSoM'11 (May 24, 2011), Waikiki, Honolulu, HI USA.
- [9] Arti Chhikarai, R.S.Chhillar, "Impact of Aspect Orientation on Object Oriented Software Metrics". Vol. 2 No. 3 Jun-Jul 2011, *Indian Journal of Computer Science and Engineering (IJCSE)*
- [10] Rakesh.L, Dr.Manoranjan Kumar Singh, and Dr.Gunaseelan Devaraj, "Software Metrics: Some degree of Software Measuremen and Analysis", (IJCSIS) *International Journal of Computer Science and Information Security*, Vol. 8, No. 2, 2010.
- [11] Nachiappan Nagappan, Brendan Murphy, and Victor Basili, "The Influence of Organizational Structure on Software Quality: An Empirical Case Study", January 2008.
- [12] Dindin Wahyudin, Alexander Schatten, Dietmar Winkler, A Min Tjoa, Stefan Biff, "Defect Prediction using Combined Product and Project Metrics ", March 2008.