ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Role of Software Metric in Software Development

[1]K Suresh
[1] Department of Computer Science and Engineering, Galgotias University, Yamuna Expressway Greater Noida, Uttar Pradesh

Abstract: The Software system keeps growing in magnitude, making it increasingly difficult to comprehend and handle. Software metrics are evaluation units of the software. Software metrics provide a quantitative framework for forecasting and preparing the software's entire process.Thus qualifying software is enhanced and can be regulated beyond challenge.Software metrics have been described as a technique of quantification of characteristics in software procedures, product lines and initiatives.Software metrics has a direct link in software development to measurement.Proper calculation is the typical situation in any engineering area and software development is no exemption, as device volume and complexity grows, software technical analysis becomes a more challenging task.Many software developers are concerned about the reliability of the software, how its performance can be evaluated and improved.The paper presents a summary of the different software quality metrics used in the software development and the paper also emphasizes on the need for software metrics and its relation to software quality.

Keywords: Software Development, Software Engineering, Software Metric, Software Quality.

INTRODUCTION

The object-oriented application development model is much more valuable and the "object-oriented metrics" used are important for quality management evaluation. Object-oriented layout includes the software quality and all the attributes associated with any massive scale or smaller scale task.It's a level where a device entity carries a specific characteristic or features.Software metrics are important agent for the whole development cycle of the software[1].Software metric gives software development evaluation through necessary software documentation, prototypes, applications, and evaluations.Fast advances in large distributed software have grown in complexity making it hard to monitor the performance.Successful outcome of software quality assurance includes measurements from software. Software metrics ideas are meaningful, justifiable, and well-established, and several commodity quality linked metrics have also been established and applied.Low size evaluation is among the key reasons for eventually failure of large software-intensive development initiatives[2].

Size is the crucial aspect in deciding cost, plan and initiative.Inability to forecast correctly leads in cost delays and cancellations that weaken trust in the system and weaken support.Scale calculation is a complex operation, the findings of which should be updated regularly along the development cycle with real estimates.Measurements for size involve "source lines of code, function points, and feature points."Complexity is a size feature that significantly influences design mistakes and hidden faults, eventually leading to quality issues, inefficiencies, and scheduling drops.Complexity has to be evaluated, monitored, and regulated constantly.Another aspect that leads to size estimation errors is creeping specifications that also need to be standard and carefully monitored. Software metrics evaluate various elements of software intricacy and thus performs a key function in examining and enhancing software quality.Recent previous analysis has shown that software metric provides useful details on software's existing performance elements, including its maintenance, scalability and efficiency.

SOFTWARE METRIC

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Proper calculation is the previous state in any area of engineering and software engineering is no exemption. Software measurements have a strong association with software engineering evaluation[3].Software metrics can decrease the rationality of errors during software quality evaluation as well as provide a numerical framework for software quality governance.Metrics are the software's mathematical quality, which are used to determine the error.File-level, class-level, element-level, process-level, application-level, and numerical value-level metrics exist in the software.This enables the project leader and software developers find errors and make the flaw avoidance technique.Software metrics can be used for every step of software engineering.Software metrics may be created during requirement evaluation, for example, to assess the cost estimate and the resource required. There is a need to develop metrics to count function points at the moment of system design. Metrics extended at the application stage are also used to evaluate software size[4].

Software Metrics provides an estimate for the software and software manufacturing process.The software metrics are to provide certain numerical explanations of the characteristics.Such characteristics obtain from the software product, the method of software development and the connected assets. They are "product, process and resources."Measurement of software offers consistent metrics for the product development method and similar products.It identifies gathers and examines observable process information, thus promoting the awareness, assessment, monitoring and enhancement of the software application system.Software metrics is a feature with software data processing, and throughput is a value that might evaluate how the software is influenced by the given characteristic.Useful metrics should allow design advancement that is effective in estimating method or product range. So appropriate metrics are supposed to be:

- Easy, accurate interpretation such that it is evident how well the metric can be assessed.
- Objective, to the maximum extent possible.
- Easy to acquire (i.e. at low cost).
- Robust that is comparatively indifferent to (logically) unimportant process or product modifications.

TYPES OF SOFTWARE METRICS

There are three types of Software Metrics:Process Metric, Product Metric and Project Metric and Process Metric[5].The different types of Software Metrics is shown below in figure 1.

*Process Metrics:*Process metrics demonstrates the "software development" mechanism.It is primarily aimed at the length of the project, the expenses incurred and the form of technique used.The process metrics can be used to increase the creation and maintenance of software.Instances include the usefulness of the elimination of defects during creation, the patterning of the emergence of test defects and the response time of the repair phase.This is the software metric used only to evaluate the quality of the software system[6]. This tests the lifespan of software development like the type of process, the role of the personnel and the duration required to complete the system.Process metrics help to determine the final system dimensions & determine whether a project will operate as per timetable.Process metrics are regarded as management metrics, which are used to calculate the process characteristics used to get the software. Process metrics involve indicators of costs, metrics of attempt, metrics of development and metrics of manufacture.

*Product Metrics:*Product metrics are also called quality metrics, which are used to calculate the software's features. Product metrics involve metrics of "product no reliability", metrics of usability, metrics of consistency, metrics of reliability, expense metrics, and metrics of scale, metrics of difficulty, and metrics of design.Product metrics supports improve the quality of the various components of the framework & correlations between current systems[7].It's one of the software metrics which we use to evaluate the quality of the software systems; mainly, it analyses the finished product of the framework like computer code or documents of layout.The outside parameters include the metric to be measured: accessibility and scalability of the program, functionality and performance, and the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

individual parameter includes software scale, accuracy, sophistication, bugs and testing.

*Project Metrics:*The project metrics are used for monitoring the situation and progress of the project. Through adjusting the task, project metrics mitigate issues or potential risks and help refine the software design program[8]. Project metrics explain the nature and implementation of the project.Examples cover software engineer amount, personnel trend over the software's development cycle, expense, plan, and profitability.

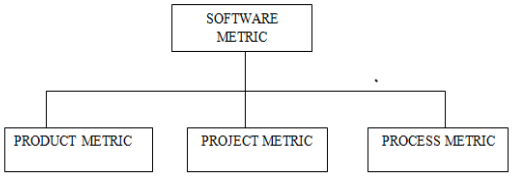

Fig. 1: Types of Software Metric

CLASSIFICATION OF SOFTWARE METRICS

The Classification of different types of software quality metrics is shown below in Fig. 2.

Size Metric:Size related metrics are the form of metrics that enables to determine the size of the software. There are several forms of software metrics for determining the software dimensions[9].The size metrics is an attempt to measure the program's Length, and the extensively utilized metrics are a "Line of Code (LOC)."The size metrics have several drawback since the production process is done that could not be calculated.Software Metric tests just project execution.The size metrics are being used to calculate "software product width, size, volume and total significance". The different types of Size Metric are:

- *Line of Code (LOC):*LOC is among the earliest types of metrics used to determine the size of the device; however the biggest problem created in LOC is "What should be included analysis."A line of code is any line of software document which is not a remark or an empty line, irrespective of the number of documents or pieces on the line.It specifically includes all lines containing file headers, definitions, and executable and non-executable statements. One of the main drawbacks with Line of Code is that it does not take into account the code's goodness: if one uses Line of Code to calculate efficiency, such a metric would punish a simple, excellently-designed program.LOC has been used in software development for a range of tasks: preparation, tracking of project development, prediction.From the measurement analysis point of view, LOC is a meaningful metric for a project's length characteristic because the experimental relationship "is shorter than" is accurately expressed by the lines-of-code relationship.
- *Function Point (FP) Metrics:* Function point metrics is the form of measures used to measure the line of code when code usability is present and therefore cannot be used earlier. There is a strategy for addressing software size measurement earlier in the development expansion.It depends primarily on investigations, user inputs, user feedback and the factors used to determine the value in determining the size of the system and therefore the purpose necessary for the growth[10].These thus include an objective development calculation of the scale of the final system and are possibly the only indicator of scale not linked to code. Measuring function points is predicated on recognizing and tallying the tasks that must be performed by the system.
- *Bang:* Such a feature metrics can be determined using the appropriate type of the information basic set of structured software description and maybe some implementations. This provides overall output assessment and is delivered to the customer.

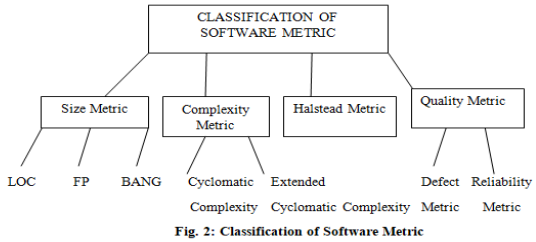

Fig. 2: Classification of Software Metric

*Complexity Metric:*The Complexity Metric is used to measure the complexity of the software. The various types of Complexity metric are:

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

*Cyclomatic Complexity:*The term Cyclomatic comes from a variety of essential cycles in linked, undirected charts. If the software's length is high, it is hard to track the number of directions easily. Because of the above explanation, "McCabe" recommends looking at the number of simple routes, which is also the number of cyclomatics[11].A program controlling loop can be described by a map that has a specific input point and escape point, and where all nodes can be reached from the input and the escape from all nodes is available.The aim is to calculate the complexity, taking into account the amount of directions in the program's control chart. The Cyclomatic Complexity is given by the following equation:

V (G) = E - N + 2P or V (G) =E-N+2
Where V (G) = Cyclomatic Complexity, E =Number of edges, N =Number of nodes, P =Number of connected components or parts

Extended Cyclic Complexity (ECC): "Mc. Cabe" assesses the program's intricacy with Extended Cyclomatic Complexity. But it completely misrepresents the distinction with specific condition in variability situations involving the conditional comment. "Myers" indicate ECC which could be explained as,

ECC=e V (G) =Pe+1, where Pe is the number of predicate nodes.

*Halstead Metrics:*Halstead is predicated on the idea that a program is composed of "operators and operands only", and that awareness of the quantities of separate and replicated operators and operands is necessary to present a variety of software characteristics like program length, duration, degree, programming ability.The main objective of such a theory is to completely discover the "software production effort that consists of some length (N), volume (V) and vocabulary (n)." It defines the following parameters-

Program Vocabulary (n): Throughout computer languages, the software can usually be interpreted as some collection of symbols and those symbols that relate to "operators and operands." Halstead defined vocabulary (n) as,

n = n1+n2 Where n1= the no of specific operators in the code. n2= the no of specific operands in the code.

Volume of program (V): The program volume (V) can be calculated as the necessary volume of space in the system.

*Quality Metric:*The Software Metric defines the quality of the software product. The various types of software metric are-

*Defect Metrics:*There is no efficient method for calculating the overall number of mistakes, the number of modifications, the software with the number of expected errors, the mistakes found by the software audits and the amount of code checks that can be viewed as a supplement to the errors.

*Reliability Metrics:*The consistency of the inner commodity is typically measured in reliability measures using the no of bugs found in the program and how long the software performs before the incident happened.

ADVANTAGES

The various advantages of software metric are as follows[12].

- For evaluation, correlation and crucial survey of different software languages with regard to the features.
- Equating and assessing skills and effectiveness of participants in software development.
- Preparing performance requirements for applications.
- Inspection in conformity with the standards and parameters of software systems.
- Inferences on the efforts to be taken in the design and production of software systems.
- Have a concept of the code's difficulty.
- Choices on further separation of complicated unit should or should not be taken.
- In supplying resource managers with instructions on the correct use.
- Comparing and making compromises between software engineering and storage costs.
- In supplying software engineers with input on success and performance during the different stages of the development cycle of software design.

- Assignment of software tools to evaluate the code.

LIMITATIONS

- The implementation of software metrics may not always be simple and, in some instances, difficult and expensive.
- Software metrics verification and evidence is based on actual / scientific data whose authenticity is impossible to check.
- These are valuable to handle software applications but not to assess the technological staff's results.
- Software metrics are commonly defined and derived on the basis of assumptions that are not consistent and may rely on the appropriate tools and operating environment.
- Much of the statistical models depend on calculations of some parameters that are not always particularly well known.
- Since software development is a complicated process, with large variability on both methods and goals, it is hard to pin down or quantify software attributes and amounts, and to establish a reliable and simultaneous measurement measure, regarding making certain predictions previous to specification layout.
- A valid theoretical validity of the metric is sometimes not feasible even though the metric characteristics are not well described.
- Empirical verification, a huge number of metrics was never exposed to quantitative verification.
- The metrics may sometimes be used in an improper way within the environment or background.
- Experimental assumption, sometimes there is no clear observational hypothesis on the test.
- Calculation objective, metrics are not necessarily described in a clear, excellently-defined way.

CONCLUSION

Software metrics have developed rapidly with the fast development in the technology sectors.Software metrics become the software management pillar and essential for application development achievement.It can be expected that the average success rate in software efficiency and software performance will increase with the use of software metrics.Software metrics improve the software's scalability and reduce the cost of maintaining the software.With the fast evolution in software field, software product evaluation becomes more complicated and the need for advanced software metrics has risen through time.There are different numbers of software metrics present, all of such metrics rely primarily on software development, maintenance and development, so to fix and diagnose different bugs these metrics should be included in the initial stages of application development life cycles, thereby avoiding differential mistakes.A measurement program based on a firm's goals can help connect, track progress toward and finally achieve certain objectives. People have to work to achieve whatever they think is relevant. The paper highlights the different types of software metrics, classification of various software metrics followed by its advantages and disadvantages.

REFERENCES

[1] A. Meneely, B. Smith, and L. Williams, “Validating software metrics,” *ACM Trans. Softw. Eng. Methodol.*, 2012.

[2] G. O’Regan, “Software Metrics Software Metrics,” 2014, pp. 151–183.

[3] K. P. Srinivasan and T. Devi, “Software Metrics Validation Methodologies in Software Engineering,” *Int. J. Softw. Eng. Appl.*, vol. 5, no. 6, pp. 87–102, 2014.

[4] W. Itzfeldt, “Quality metrics for software management and engineering,” in *Managing Complexity in Software Engineering*, 2011, pp. 127–152.

[5] K. Mordal, N. Anquetil, J. Laval, A. Serebrenik, B. Vasilescu, and S. Ducasse, “Software quality metrics aggregation in industry,” in *Journal of software: Evolution and Process*, 2013, vol. 25, no. 10, pp. 1117–1135.

[6] C. Symons, “metrics Software Industry Performance,” *IEEE Softw.*, vol. 27, no. 6,

pp. 66–72, 2010.

[7] D. Galin, "Software Product Quality Metrics," in *Software Quality: Concepts and Practice*, 2018, pp. 346–374.

[8] S. Dhawan and Kiran, "Software Metrics – A Tool for Measuring Complexity," *Int. J. Softw. Web Sci.*, pp. 63–66, 2012.

[9] K. Yamashita *et al.*, "Thresholds for Size and Complexity Metrics: A Case Study from the Perspective of Defect Density," in *Proceedings - 2016 IEEE International Conference on Software Quality, Reliability and Security, QRS 2016*, 2016.

[10] C. Jones, "Function points as a universal software metric," *ACM SIGSOFT Softw. Eng. Notes*, 2013.

[11] C. Ebert and J. Cain, "Cyclomatic Complexity," *IEEE Softw.*, 2016.

[12] E. Bouwers, A. Van Deursen, and J. Visser, "Software metrics: Pitfalls and best practices," in *Proceedings - International Conference on Software Engineering*, 2013.

[13] V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Certain Investigations on Strate gies for Protecting Medical Data in Cloud", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014

[14] V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Investigations on Remote Virtual Machine to Secure Lifetime PHR in Cloud ", International Journal of Innovative Research in Computer and Communication Engineering Vol 2, Issue 10, October 2014

[15] V.M.Prabhakaran, Prof.S.Balamurugan, S.Charanyaa," Privacy Preserving Personal Health Care Data in Cloud" , International Advanced Research Journal in Science, Engineering and Technology Vol 1, Issue 2, October 2014

[16] Ishleen Kaur, Gagandeep Singh Narula and Vishal Jain, "Identification and Analysis of Software Quality Estimators for Prediction of Fault Prone Modules", INDIACom-2017, 4th 2017 International Conference on "Computing for Sustainable Global Development".

[17] Ishleen Kaur, Gagandeep Singh Narula, Ritika Wason, Vishal Jain and Anupam Baliyan, "Neuro Fuzzy—COCOMO II Model for Software Cost Estimation", International Journal of Information Technology (BJIT), Volume 10, Issue 2, June 2018, page no. 181 to 187 having ISSN No. 2511-2104.

[18] Ishleen Kaur, Gagandeep Singh Narula, Vishal Jain, "Differential Analysis of Token Metric and Object Oriented Metrics for Fault Prediction", International Journal of Information Technology (BJIT), Vol. 9, No. 1, Issue 17, March, 2017, page no. 93-100 having ISSN No. 2511-2104.