ISSN:0975-9646

Role of Software Metrics to Improve Software Quality

Vikash Chauhan
M.Tech Scholar
Deptt. of Computer Science & Engg.
KNIT Sultanpur, UP, India

Dharmendra Lal Gupta
Assistant Professor
Deptt. of Computer Science & Engg.
KNIT Sultanpur, UP, India
Member IEEE

Sarvottam Dixit
Professor
Mewar University, Chittorgarh
Rajasthan, India

Abstract—This paper describes software metrics and provides the relevant matters about the need of software metrics in software development, the basic quality attributes of software metrics and how can we improve the quality of software using software metrics is also mentioned. Quality in fact aids easy and high productivity, which has brought software metrics to the forefront. Currently software metrics are being used by software companies to improve the quality of software so that they can improve their productivity and make profit. This paper also describes the role of software metrics in software engineering.

Keywords-Software metrics; software metrics component; software quality factors.

I. INTRODUCTION

"You cannot control what you cannot measure" – Tom DeMarco. This statement justifies like that in real life Thermometer requires to measure the temperature of body and Scale requires to measure the length of table, so in this way software quality can also be measured in terms of metrics. Measurement of project or software is very important because without measuring the quality we can't achieve the goal of software. By measuring we can control and improve the performance of our project or software. Correct measurement is also very important because we have to find only that factors which improve the quality of our project. According to our problem we should select the quality factors and using these factors, we should select the software quality metrics. Software quality metrics are dependent on software quality factors so selection of quality factors is very important.

If one does not have any information about where he/she is now certainly won't know where he/she will be in the future. In 2004, the Standish Chaos report found only 29% project met their quality criteria for success of project and this report also told that cancelled cost of that projects were near about $55 billion, so if we will use the proper metrics in developing of software then we can increase the quality and can decrease the probability of failure of projects.

II. SOFTWARE QUALITY FACTORS

Some important software quality factors have been shown in table I which is mentioned in alphabetically order.

TABLE I. Software quality factors with their details

Quality Factors	Detail Description
Accuracy	It is the extent to which the output of a program is sufficiently precise their intended use [2] [3] [4].
Analyzability	Attributes of software that relate to the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified.
Adaptability	Attributes of software that relate to on the opportunity for its adaptation to different specified environments without applying other actions or means than those provided for this purpose for the software considered.
Clarity	It measures how clearly a person can understand a program. Clarity is the extent to which a program has enough information for a reader to find its objectives, assumptions, constraints, inputs and outputs [2] [3] [6].
Completeness	It is the extent to which our software fulfills the overall mission requirements. It covers a broad area [2] [3] [6].
Complexity	Complexity seen as a criterion, which is a level of attribute that falls somewhere between the measureable software metrics and quality factors [6].
Conciseness	It is the ability of a program to satisfy functional requirements using a minimum amount of software [3] [6].
Consistency	It can be divided into two groups: internal and external. Internal is the degree to which software satisfies specifications and external is to which a software product contains uniform notation, symbols and terminology [3]
Correctness	It measures the extent to which the software design and implementation confirm to specifications and standards [6].
Compliance	Attributes of software that make the

Quality Factors	Detail Description
	software adhere to application related standards or conventions or regulations in laws.
Changeability	It is relate to the effort needed for modification, fault removal or for environmental change.
Conformance	It is set of attribute that make the software adhere to standards or conventions relating to portability.
Efficiency	The ratio of useful work performed to the total energy expended (Gilb 1977).
Expandability	It is associated with making changes to a program. It is the amount of effort required to increase the capability or performance of software [6].
Flexibility	It is defined as the amount of effort required to change the mission, function, or data of software to satisfy other requirements. This is broad enough to include adaptability and modifiability [6].
Functionality	It is the set of attributes that relate to the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.
Fault tolerance	It is the set of attributes of software that relate to its ability to maintain a specified level of performance in cases of software faults.
Integrity	Integrity is the measure of the ability of a program to perform correctly on different sets of input (Kreitzberg 1982). It is measure of how well a program has been tested [5].
Installability	Installability is the effort needed to install the software in a specified environment.
Interoperability	It is the set of attributes of software that relate to its ability to interact with specified systems.
Learnability	Learnability is the users' effort for learning its application (for example, operation control, input, and output).
Maintainability	It is the measure of the effort and time required to fix bugs in the program (McCall, Richards, and Walters 1977).
Modifiability	It measures the cost of changing or extending a program (McCall 1977).
Modularity	Modularity is the expressed in terms of program subunits that are developed in dependently and then interfaced (Baker 1979).
Maturity	Maturity is the frequency of failure by faults in the software.

Quality Factors	Detail Description
Operability	Attributes of software that relate to the users' effort for operation and operation control.
Performance	How well will the product perform under the adverse condition [6]?
Portability	How easily a software product will run on a computer configuration other than the current one (Boehm et al. 1978).
Reliability	Reliability is the extent to which a program can be expected to perform its intended functions satisfactorily [1].
Reusability	It is the effort needed to convert software for another use (Bowen, Wigle, and Tsai 1985).
Recoverability	Recoverability relates to the capability to re-establish its level of performance and recover the data directly affected in case of a failure.
Resource Behaviour	It relate to the amount of resources used and the duration of such use in performing its function.
Replaceability	Replaceability relates to the opportunity and effort of using it in the place of specified other software in the environment of that software.
Simplicity	Simplicity is the extent that it uses data and control structures for organizing the program, and uses easily understood constructors [1].
Suitability	Suitability relates to the presence and appropriateness of a set of functions for specified tasks.
Security	Security is the ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.
Stability	Stability relate to the risk of unexpected effect of modifications.
Survivability	Survivability is the extent to which the software performs and support critical function without failure within a specific time period.
Testability	Testability is the extent to which the software facilitates the establishment of acceptance criteria and supports evaluation of its performance (Boehm1978). Testability is the measure of our ability to test software (McCall).
Time behaviour	It relate to response and processing times and on throughput rates in performing its function.
Understandability	Understandability is the ease with which a program can be understood (McCall 1977). It can be viewed as the extent to which the purpose of the

Quality Factors	Detail Description
	product is clear (Boehm et al. 1978).
Usability	Usability measures the effort required to train a person to use the software (Bowen, Wigle, and Tsai 1985).
Verifiability	Verify the specified software operation and performance.

These above properties decide the quality of software. According to these properties we select the proper software metrics to apply in our software.

III. SOFTWARE METRICS AND SOFTWARE QUALITY METRICS

A. SOFTWARE METRICS

Today size of project is increasing and it is very hard to maintain the quality of project so software metrics have to be required in software industries. Basically there are following three types of software metrics: process metrics, project metrics and product metrics [8] [9].

1) Process Metrics:

Process metrics show the development process of the software. These are collected across the entire project and over long periods of time. These metrics are used in strategic decision. In process metrics we measure the effectiveness of a process by deriving a set of metrics based on outcomes of the process such as:

- Conformance to the schedule
- Delivered work products
- End users report the defects
- Errors not fixed before release of the software
- Expended calendar time
- Expended human effort
- Time and effort to complete each generic activity

2) Project Metrics:

Project metrics are useful to monitor the project situation and status. It minimize the development schedule of the project, improve quality, minimize defects, amount of rework of project reduce, overall project cost is reduced. Project metrics enable a software project manager to

- Maintain work flow or tasks of project
- Assess the status of the current project
- Evaluate project team's ability to control quality of software work products
- Track potential risks of project
- Uncover problem areas before their status becomes critical

3) Product Metrics:

Product metrics provide the solution to understand the attribute of the software product and quality at any phase (analysis, design, coding, and testing) of its development. It can also measure

- Complexity of the software design, procedural designs and source code
- Maintainability
- Performance
- Portability
- Product scale
- Size of program
- Testing techniques

B. SOFTWARE QUALITY METRICS

Software metrics and software quality factors compose the software quality metrics. These metrics provide measures of the software attributes and may be in the form of checklists used to grade a document produced during the development.

Software Quality Metrics (SQM) = Software Metrics (SM) + Software Quality Factor (SQF)

Relationships between the set of metrics related to quality attributes (factors) and rating of quality factors have been established via regression analysis performed on empirical data. This relationship can be shown via linear equation. An example is given below.

$$r_f = c_1 m_1 + c_2 m_2 + c_3 m_3 + \cdots\cdots\cdots\cdots + c_i m_i$$

where:

$r_f = rating\ of\ the\ quality\ factor, f$

$c_i = regression\ cofficients$

$m_i = various\ measurements\ identified\ as\ relating\ to\ quality\ factor, f$ [7]

By creating the above relationship it is used as predictor. The measurements m_i are applied at specific times during the development. There are following aspects of this approach.

- At highest level it is user-oriented
- At lower level oriented it is software-oriented
- Provide attributes' qualifications
- It is easy to use and can be applied any time during the software development
- Additional metrics, function, and criteria can be added as the software technology changes [7]

IV. DISCUSSION, CONCLUSION AND FUTURE WORK

In this paper we described the quality attributes of software by which we can improve the performance of software. We have presented software metrics that are used according to our problem. We have described here that the differences between software metrics and software quality metrics. First of all we have to select quality attributes that are used in our project or software after that we have to select software metrics suite and lastly according to our priority we have to select one of the metrics and apply it in our project. There may be different combinations of quality attributes to make many different metrics. These metrics have different properties. We have many metrics at the

present time so we have to choose very carefully according to our need. It is not necessary that we should use only one metrics in whole project, we should use such metrics that provide better and efficient result. Software engineering provides us many rules and regulations to select the appropriate metrics.

In this area of research work there are a lot of queries, and their solutions which may be found in our future work. These queries may be as given below:

1. What are the bases to relate these quality attributes?
2. If we change one attribute, what will be the effect on another attributes?
3. How we identify which software quality metric is more important than other software quality metrics and why?

REFERENCES

[1] McCall, Jim A., Paul K. Richards, and Gene F. Walters,"Factors in Software Quality", RADC-TR-77-369, Volumes I, II, and III, RADC, Griffiss Air Force Base, NY, November 1977.
[2] Gilb, T., "Software Metrics", Winthrop, Inc., Cambridge, MA, 1977.
[3] Boehm, B. W. et al., "Characteristics of Software Quality", North-Holland Publishing Company, New York, NY, 1978.
[4] Walters, G. F., "Applications of Metrics to a Software Quality Management (QM) Program", Software Quality Management, Petro celli, New York, NY, 1979.
[5] Kreitzberg, Charles B. and Ben Shneiderman, FORTRAN Programming: A Spiral Approach, Harcourt Brace Jovanovich, Inc., 1982.
[6] Bowen, Thomas P., Gary B. Wigle, and Jay T. Tsai, "Specification of Software Quality Attributes", RADC-TR-85-37, RADC, Griffiss Air Force Base, NY, Volumes I, II, and III, February 1985.
[7] Joseph P. Cavano Rome Air Development Center, James A. McCall General Electric Company. "A Framework for the Measurement of Software Quality".
[8] Tu Honglei1, Sun Wei1, Zhang Yanan1, "The esearch on Software Metrics and Software Complexity Metrics", International Forum on Computer Science-Technology and Applications, 2009.
[9] Mrinal Singh Rawat, Arpita Mittal, and Sanjay Kumar Dubey "Survey on impact of software metrics on software quality", IJACSA Vol 3, No. 1, 2012.