Available online at http://www.journalcra.com

International Journal of Current Research
Vol. 8, Issue, 07, pp.34525-34527, July, 2016

INTERNATIONAL JOURNAL OF CURRENT RESEARCH

ISSN: 0975-833X

RESEARCH ARTICLE

SOFTWARE METRICS & THEIR INFLUENCE ON SOFTWARE QUALITY

*,1,2 Anupama Munshi

[1]Research Scholar PhD. Department of Computer Sciences, Jagannath University Jaipur
[2]Asst.Prof Jagannath International Management School (JIMS), Kalkaji New Delhi

ARTICLE INFO

Article History:
Received 24th April, 2016
Received in revised form
20th May, 2016
Accepted 06th June, 2016
Published online 16th July, 2016

Key words:

Software Metrics, Software Quality, Project, Process.

ABSTRACT

Software Metrics are instrumental in improving the quality of the Software. The sustainability of any software depends on the quality of that particular software. In this technical era where there is no dearth of software's, we need to find out various factors that affect the quality of the software, and an important one of them being the Software Metrics. This paper attempts to define the various software metrics available for the measurement of software and lays stress on the relationship between the software metrics and the quality of the software. Since the basic objective of software development is to provide high quality software, this paper provides an insight into the need of good software metrics which can help in the improvement of the quality of the software product. Also the paper lays emphasis on the impact of Metrics on the quality of the Software.

Citation: Anupama Munshi, 2016. "Software metrics & their influence on software quality", *International Journal of Current Research*, 8, (07), 34525-34527.

INTRODUCTION

The continuous improvement is an important requirement for any organization in general and Software Industry in particular. A Continuous Improvement Process (CIP) is a continual effort to improvise on the processes, products, services or projects. Software metric is technically defined as the standard of measurement by which the efficiency, progress, performance, productivity and the quality of a deliverable process, project or product can be assessed. Software metrics are very important in building the predictability and improving the quality of a software project. It has been rightly said that *"If you don't measure something, you can't manage it"*. Software metrics are instrumental in improving the quality of software. Therefore it can be said that Software metrics can be considered as the means of measuring software qualities which in turn are required for quantitative comparison, cost estimation and quality evaluations. The efforts done in CIP can seek "incremental" improvement over time or "breakthrough" improvement all at once. Evaluating whether there is an incremental improvement and setting up procedures to track and measure these improvements is the tricky part and this is where Metrics play a significant role. Hence Software metrics enable Project managers to:

- Assess status of ongoing project in terms of schedule, cost and profitability.
- Foresee any potential risks.
- Nail down the problems much before they become severe.
- Keep a check on project profit1ability.
- Assess productivity of team.
- Assess quality of work products to be delivered.

The Software Metrics Cycle can be therefore shown as:

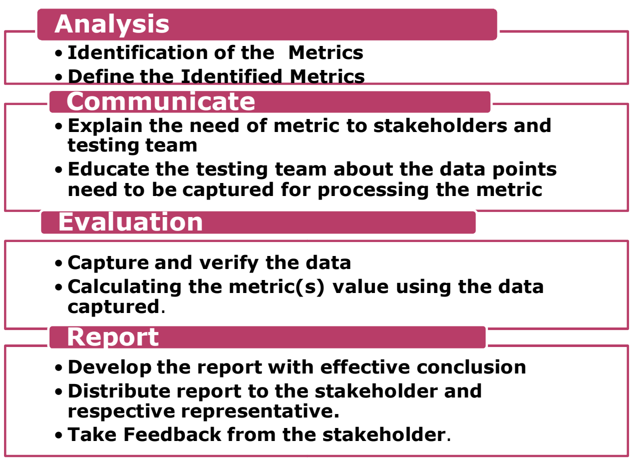

Fig. The Metrics Cycle:

***Corresponding author:** *1,2 Anupama Munshi,*
[1]Research Scholar PhD. Department of Computer Sciences, Jagannath University Jaipur
[2]Asst.Prof Jagannath International Management School (JIMS), Kalkaji New Delhi

Categories of Software Metrics

There are numerous project management metrics defined based on complexity and nature of project. However, the following performance metric groups cover all the important aspects of a project to measure during execution.

1: Schedule and Effort/Cost Variance: The objective of this metric is to measure the performance as well as progress of the project against signed baselines. This metric is very important and is the base for profitability of project. It integrates project scope, cost and schedule measures to help the Project manager (PM) to assess and measure project performance and progress. Under this method, at any given point in time, project performance to date is used to extrapolate the expected costs and duration at project completion. EVM develops and monitors three key dimensions of each work package:

- ***a)*** *Planned Value (PV)*
- ***b)*** *Earned Value (EV)*
- ***c)*** *Actual cost (AC)*

Using these three variables project Schedule variance and Cost variance metrics can be derived which shows if the project is running over or under budget; project is running behind or ahead of schedule.

2: Productivity: Resource Utilization: The goal of this metric is to measure productivity of resources involved in project and allow project Manager check over or under-utilization cases.

Utilization% = Total Effort spent by resource/Total Budgeted Effort for the resource

Any over-utilization and under-utilization indicated by this metric has an impact on the project's profitability. It is important for the PM to track this metric very closely and find out the reason for deviations and the action items to bring back resource utilization to optimal level. Using this, PM can analyze effort distribution across different project phases/activities. For e.g. Effort distribution can tell PM that how much effort is being spent on defect resolution, customer support or design activities. Effort distribution from time reporting systems can also tell the areas of improvement for better estimation/planning for the next project.

3*:* Quality and Customer Satisfaction: Throughout the execution of project, Quality Assurance should always be on the radar of project manager. Quality here is defined as the number of severe, medium or low defects delivered through the lifetime of the project. It indicates the health of the deliverable to the end user and drives the Customer Satisfaction. PM needs to define, based on project type, what severe, low and medium means. Quality should be reported throughout the life of the project; the later defects are caught, the more impact they will have on the project.

Relationship between Software Metrics and Software Quality

In today's world the size of the software projects is increasing considerably on a daily basis, therefore there is a desperate need of Software metrics that enhance the quality of the software with appropriate and accurate measurements. Software metrics and software quality factors compose the software quality metrics and these metrics provide the measures of 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.

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

Where

r_f = *rating of quality factor f*
c_i = *regression coefficients*
m_i = *various measurements identified as related to quality factor f*

The measurements m_i is applied at specific times during the development.

Quality may also be defined as the number of: severe, medium or low defects delivered through the lifetime of the project. Quality must be reported throughout the life of the project; the more lately the defects are caught, the more impact they have on the project.

It can be said that all the software characteristics are measurable and therefore can be quantified, also that they are all related to one or more software quality characteristics. Metrics for both process and software tell us to what extent a desired characteristic is present in our software systems. The definition of software quality assurance emphasizes on the importance of measurement or metrics. It is therefore the software metrics that indicate the value of standards, processes and procedures that SQA assures are being implemented correctly with in a software project. SQA also collects relevant software metrics to provide input into a SPI (such as a CMMi continuous improvement initiative). This exercise of constantly measuring the outcome, then looking for a causal relationship to standards, procedures and processes makes SQA and SPI pragmatic disciplines. The whole process of setting up a SDLC, then selecting the correct metrics and then establishing causal relationships to parts of the SDLC is more of an art than a science. It is therefore for this reason that there a few, if any, off the shelf answers to SQA, SQC and SPI. It is a question of where are the risks and challenges of a given environment. In the subsequent table there is an attempt to relate the ideas of quality metrics, quality characteristics, SPI, SQC and SQA together with some examples by way of clarifying the definition of these terms.

The example below has been taken from Software Assurance Technology centre (SATC) at NASA.

The table shows the relationship in terms of cross references Goals, Attributes (software characteristics) and Metrics.

Table 1. SATC Software Quality Model

Table 2 Goals	Attributes	Metrics
Requirements Quality	Ambiguity	No. of Weak phrases
	Completeness	No. to be added and to be determined
	Understandability	Document Structure
	Volatility	Count of Changes.
Product (code) quality	Reusability	Correlation of complexity/size
	Internal Documentation	Comment Percentage
	External Documentation	Readability Index
Implementation Effectivity	Resource Usage	Staff hours spent on life cycle activities
	Completion Rates	Task Completions
Testing Effectivity	Correctness	Errors and Criticality.

Conclusion

In the context of this paper it can be ascertained that there is a need to link metrics to an organizational or project objective to demonstrate if at all we are achieving what we committed. There needs to be a verification that the chosen metrics are covering different areas that will define if a project reaches the end with or without success. It is necessary to stay away from over analyzing data provided by one metric. It can be concluded that software metrics play a significant role in determining the quality of the software. It can also be deduced that it is useless to use a metric that cannot be used to base or define action plans. Target values need to be defined for metrics like performance, once in order to take corrective actions, the team must be aware of the expected minimum and upper limit value ranges. This paper also finds that it is very essential to measure various attributes of software in accordance with the project being developed. This paper establishes the fact that software metrics have a great impact on the software quality and hence are paramount while developing the software.

****