

An Analysis on Type of Software Metrics Used In Evaluating Quality of the End Deliverable

MD. Imran Alam

Research Scholar (Computer Science), CMJ University, Shillong, Meghalaya

Abstract— *Software metrics play a very important role in the management of software professional projects. The evolution and growth of metrics in the last decade has been on the higher side and it is used to track development process, the impact of restructuring and estimate the code quality. The paper below describes the definition of the term, the benefits of it, and the various types and in the final analysis how to determine the quality of metrics. In the overall scenario holds good for this concept in the days to come.*

Index Terms— *Software metrics, Analysis, end deliverable.*

1. INTRODUCTION

Software metrics have always been important for the software engineers to assure software quality. In normal terms software engineers attempt to derive a set of indirect measures that provide an indication of quality of some representation of software. Out of the numerous proposed metrics, selecting a particular complexity metric is again a problem, so every metrics has its own advantages as well as disadvantages. In fact there is an ongoing effort to develop and find such a complex metrics which addresses most of the issues of parameters of software (Gervasi, 2008).

Software metrics is an important issue in modern software development. It allows a company to analytical plan for the business development, support planning and control activities, benchmark products and services and ascertain the output of the future course of action. A metrics system is business intelligence fundamental (Carroll and Daugherty, 2007).

This issue has been in focus for a considerable amount of time, and even supported by standards, the sophistication of the business requires an integrated software metrics approach. This approach could provide the following

- Determination of necessary data collection
- Implementation guidance
- Establishment of metrics based on the needs of the business.

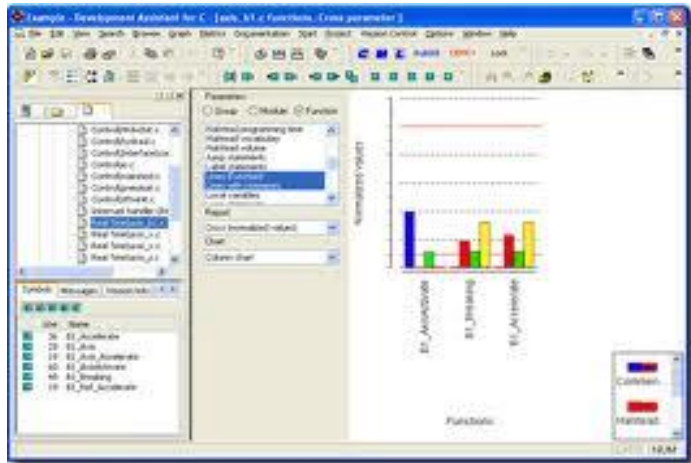
Other than all this it includes auditing, documenting as well as controlling. Considering a different angle in the field of software engineering the term metrics is used in reference to multiple concepts. The examples in this regard could be the quantity to be measured, the measurement models applied as well as the measurement of the models themselves. In the present time the inventory of software metrics is so diversified that it includes many individual proposals, that it is not considered economic feasible for the researches as well as the industry to investigate each and every metrics individually (Abran, 2010).

Many experts from the academic as well as from the domain of industry would agree on the fact that most of the software products and their development process in the world of today are of low quality. Facts are a testimony to the fact that 24 % of the software projects fail to take off and only 32 % of the projects are complete. In order to understand the impact of code changes and track complex issues as well as code quality software metrics are used in the software development cycle. In this regard the existing framework as well as new modules are developed. Most metrics are developed for a scope (Dong, 2010).

DEFINITION OF SOFTWARE METRICS

The term metrics means different things to different people. Software metrics means “continuous application of measurement based techniques to the software development process and its products to supply meaningful and timely information, together with the use of those techniques to improve the process as well as the products. The metrics are standardized ways of measuring

the attributes of software processes, products and services in order to provide the information needed to improve the process, products as well as services. The same metrics can be used to monitor the progress of those projects and provide the feedback for continual improvement.



The use of measurement in this regard is common. It is a common phenomenon that people use some form of measurement in their life in some form or other. In this regard there is confusion among the measurement aspects as the first thing to be understood is the entity aspect. The measurement attributes on the internal as well as the external aspects which are an epitome of the internal as well as the external characteristics. One of the most common examples of measurement is the mapping system which is commonly accepted and defined. In ancient system there were no real standard measurement aspects which really complicated things (Westfall, 2010).

It also helps to understand whether the organization is achieving the software goals or not. One needs to understand the concept of useful metrics which should possess the following characteristics

- They should be simple and easy to understand
- Inexpensive to use
- Robust
- Easily accessible by all the stakeholders online
- Consistent used over a period of time

Useful metrics must be accompanied by data that is correct. In this regard most of the data is collected over a period of time and is associated with a particular activity.

Organizations with maintenance and development standards and tools will have an easier time for collecting metrics as data is available in soft form. In this regard if the definitions of metrics are missing or they need to collect it manually it may be a problem. Another measure is that measures need to be taken so that the senior management is aware and they quickly assess the status of each project as well as health of the organization. It is recommended that an organization should concentrate on a few metrics which may not be more than 5 in number which are considered important from the organization as well as the customer's point of view (Limaye, 2009).

An organization must clearly define the process for collecting metrics and expected tools from the tools used for data collection, metrics generation as well as distribution. Output of any process may act as an input to the others. The reports of metrics must help the organization for planning improvement activities which are defined by the metrics according to the level of priority as and when determined by them. Metrics need to be used for development process development as well as testing process development.

TYPES OF SOFTWARE METRICS

It has been found out that the cost of software has a significant impact on the development of the project. Software is of different types and different types of software require varied costs for their development. Software metrics is defined as the qualities that can directly as well as indirectly characterize the software product, its operational and the development part. Software metrics can be classified into direct as well as indirect metrics. This classification is based on the ease of identification purpose. Another type of classification which can be associated could be based on the qualities of the software. So it is quiet necessary to consider only important attributes of the software.

Customers will be turned off metrics that require a great deal of work to collect additional or new data. In this regard it is suggested to start with the new process and once a desired level of success has been achieved, the metrics customers will be more open to more radical ideas. They may even come up with their own set of metrics as well. With all the possible metrics it could be considered that a host of metrics can be implemented as well. The metrics that are needed during the initial stages of the operation will be quite different from the ones that will be required during the testing stage. In this regard some metrics are collected and recorded on periodic basis while others may be needed on an event to event basis. In the midst of this some metrics may be used only once during the life cycle of the software project and this may be due to the

investigation or a case study aspect. It needs to be understood that the software metrics are the outputs of software development, operations as well as the maintenance process. Some of the examples of software entities include

- Requires documentation
- Software design specifications
- Codes
- Project plans, schedules and reports.

Lines of Control (LOC) is the most commonly used as well as misused methods of software metrics. This method may be also be used in other metrics to normalize the measures so projects releases or products of different sizes may be compared. The problems as well as variations of LOC can be well documented. Some of the problems include

- Problems in counting LOC for systems using multiple languages
- Difficulty in estimating LOC in the early stages of the software cycle.

In fact no industry specific standards are implemented for LOC. So it is necessary that specific criteria for encountering LOC in an organization should be adopted (Westfall, 2010).

Other types of classification related to metrics are as follows

- Software quality metric- a quantitative measure of the degree to which software possess a given attribute that affects its quality. Examples are reliability as well as portability
- Management metric – can be used to measure the activities of the management like budgets, value as well as the schedules
- Software quantity metric- A quantitative measure of some physical attribute of software (Brykczynski and Stutz, 2004)

ADVANTAGES OF SOFTWARE METRICS

Even in an insurance business which is well acquainted with the benefits of metrics in the core business, it is pretty difficult to sell the benefits of software metrics. With targeted and appropriate metrics, data are collected as well as analyzed which relate to the software product and the development process and support effective management.

In this regard the first stage is to identify the product or the services. Some of the goals which can be achieved with a common metrics programme are as follows

- To establish a common understanding throughout the organization
- To determine the information requirements and management processes
- To identify and accomplish the activities for measurement
- To use measurement results for decision support as well as the basis for effective communication
- To collect, analyze, interpret as well as store the results of measurement (Bundschuh and Dekkers, 2008)

Metrics use the past measurements to give guidance for future directions. It could be a basis for controlling as well estimating the progress of the project.

Metric data is very useful when used in conjure with a quality improvement programme. In fact the goals of the quality improvement programme must be consistent with the support of the overall quality improvement programme. The software metrics should support the desired improvements for software quality. In this regard direct activities for improving quality as well productivity should be established. They are used for managing as well as monitoring software projects. If metric data are available to the software project managers on a frequent basis, corrective action can be made to a project plan which will increase the probability of successful project completion (Fries, 2006).

As more and more organizations have adopted metrics, metric usage has evolved out to be a strategic advantage of sorts. An example from the industry will make things more clear. A federal government decided to measure the size of their software portfolio some years back. The result was they found how large their portfolio and that was it. They realized that a single method did not provide any insight for decision making. The question is what they could have done differently. In this regard the organization should have determined the goals, ask determined questions and then clearly set out the metrics to meet the demand.

In the midst of this it should be kept in mind that software metrics measures only about software products as well as services, but never takes into consideration individual people. In fact ignore this crucial aspect certainly leads to

damage control and ultimately the failure of a management programme. However metrics needs to be seen in the larger context of thinks and there might be many differences to the results of management metrics.

Accurate estimation of the problem size is fundamental to satisfactory estimation of other project estimators such as effort, time duration for completing the project as well as the total cost of the software deployed in the project. Two major metrics are used to determine the size which are lines of code (LOC) as well as the function point. (FP). The use of both the domains in the project size estimation has its own set of advantages as well as disadvantages associated with it. Out of both the metrics available LOC is the most popular one as it counts the number of instructions in the developed programme (Mall, 2009).

One of the major advantages of software metrics is to understand what may be done to improve the software process. This relates to the software entities as well as the software mechanisms derived from the process. In a way measurements provide valuable feedback in the evolution of the software process. In the domain of software engineering, the term software metric is used. It is a qualitative measurement derived from any attribute of the data. It makes it possible for the software engineers to measure as well as predict the progress of the software (Peters, 2000).

HOW TO ASCERTAIN THE QUALITY OF SOFTWARE METRICS

Software metrics provide numerical methods for measuring quality issues associated with the development of computer systems. Numerical measures give the project manager a clear and concise indication of how the project is progressing. There are normally two types of software metrics which are

- Results metrics- This measure is normally adopted at the end of a project or the end of a particular stage of the project
- Predictor metrics- They are implemented during the various stages of the project and provide numerical feedback as to the structure as well as the complexity of the code sections. In a way they may be used to ascertain the programme size which is used for determining the final resources when the programme goes live.

Some of the following models have been developed over the years to put the measurement techniques for the quantitative assessment of the software quality. These are all indirect techniques that do not measure quality but

some aspect associated with it. One of the major ones in this regard is software quality indices (Hellingsworth, Hall and Anderson, 2001).

The software industry is more than 60 years old, which makes it a fairly old industry. One could assume that after 60 years, the software industry would have had well established methods for measuring productivity as well as quality, along with an accurate benchmark of data derived from thousands of measured projects. However this is not the exact case. When compared to the year 2008, the software has dozens of metrics available, in fact some of them have handful of users as well. Very few software metrics are supported by formal standards as well as training. Benchmarks vary from being complete to so sparse in terms of what is collected that the value is difficult to ascertain (Jones, 2008).

In fact major topics like service oriented metrics have severe shortage of researched as well as published results. For the first ten years when the software industry started that is the period from 1947 to 1957 most of the applications were quiet small and less than 1000 code statements were issued in all. All of these were written in small assembled languages and in certain times they were merged with machine language as well. The first attempts to measure productivity was adopted using the code of lines mechanisms. In this regard coding took around 50 %, debugging and testing took around 40 % and the rest of the components composed of around 10 %. The precise reason for such a situation arising was the code lines were considered fairly effective and at the same time troublesome also. After 1957 the situation began to change rapidly with the low level languages being replaced by powerful languages such as COBOL as well as FOTRAN as well as APL.

Here are some project metrics that you may use with your quality management plan for your software project which are

- Benchmarking- In this mechanism the current project activities with similar projects in the past.
- Histograms- These are bar graphs that describe project issues and rank order of the causes of those problems
- Control charts- which show process that are not reliable or all that stable as well
- Project audits- audits which are performed to determine whether the project confers to the defined parameters.

- Procurement metrics- Metrics which are used to evaluate the contractors as well as the vendors (Luckey and Phillips, 2006).

The organization may have specific project metrics which one needs to be familiar with. There are several project metrics available; some may be helpful for a particular project than the others. The key factor in this regard is to be flexible and use a variety of project metrics that suits the need for each project.

CONCLUSION

The concept of software metrics has been established and a variety of software metrics have been established over time. The approach to software metrics must be made in a careful, scientific way marked by a traditional scientific paradigm of hypothesis, evaluation, criticism as well as review. Progress in this regard will be on all counts incremental. The search for exact truth may not be true but we need to put value propositions to the truth aspect. In the domain of software metrics it is important to show flexibility to the users. It is indeed a difficult proportion as most of the research in the field concentrates on precise results. On the other hand the users could not get the precise results from it or the desired results cannot be expected from it.

The benefits of software metrics must be justifiable in relation to the business value as well as the objectives of the business. On the whole software metrics have been proven to relate to software quality. Many software organizations fail to get the desired results from their investment in software metrics. In this regard software managers are advised to give priority to critical issues. And pay attention to significant software metrics programmes

REFERENCES

1. Gervasi, O. (2008). Computational Science and Its Applications - ICCSA 2008: International. Berlin: Springer.
2. Carroll, S., and Daughtrey, T. (2007). Fundamental Concepts for the Software Quality Engineer, Volume 2, Volume 2. Milwaukee: Quality Press.
3. Abran, A. (2010). Software Metrics and Software Metrology. New Jersey: John Wiley
4. Dong, J. (2010). Formal Methods and Software Engineering: 12th International Conference on Formal Engineering Methods. Berlin: Springer.
5. Westfall, L. (2010). The Certified Software Quality Engineer Handbook. Milwaukee: Quality Pres.s.
6. Limaye (2009). Software Testing. New Delhi: Tata McGraw Hill.
7. Brykczynski, B., and Stutz, R. (2004). Software Engineering Project Management, 2nd Ed. New Jersey: Wliey.
8. Bundschuh, M., and Dekkers, C. (2008). The IT Measurement Compendium. Berlin: Springer.
9. Fries, R. (2006). Reliable Design of Medical Devices, Second Edition. Boca Raton: CRC Press
10. Mall, R. (2009). Fundamentals Of Software Engineering 3Rd Ed. New Delhi: Phi.
11. Peters (2000). Software Engineering: an Engineering Approach. New Delhi: Wiley.
12. Hellingsworth, B., Hall, P., and Anderson, H. (2009). Higher National Computing. MA: Butterwoth Heineman.
13. Jones (2008). Applied Software Measure.3E. New Delhi: Tata McGraw Hill
14. Luckey, T., and Phillips, J. (2006). Software Project Management For Dummies. NJ: Wiley.