International Journal of Enhanced Research in Management & Computer Applications, ISSN: 2319-7471
Vol. 4 Issue 2, Feb.-2015, pp: (21-24), Impact Factor: 1.296, Available online at: www.erpublications.com

Analysis and Integration of Software Testing with the help of Metric Tools

Khushi Ram

Email id: **khushiram.chauhan@gmail.com**

Abstract: Software Metrics is a Measurement Based Technique which is applied to processes, products and services to supply engineering and management information and working on the information supplied to improve processes, products and services, if required. It is a mathematical definition mapping the entities of a software system to numeric metrics values. Furthermore, we understand a software metrics tool as a program which implements a set of software metrics definitions. There are number of software metric tools available, use different methods to assess metric based software systems and hence project different results. The results are thus tool dependent and are in question for validation. Measurements show that, for the same software system and metrics, the metrics values are tool depended. This paper will focus and study on integration of Software testing with the help of metric tools.

Keywords: Software product Metrics, metric tool Measurement, testing, Verification.

Introduction

Accurate measurement is the priority of any software metric tool. A large body of software quality metrics has been developed, and numerous tools exist to collect metrics from program representations. This large variety of tools allows a user to select the tool best suited. This paper will show that different metrics tools show different metrics values for same measurement and same project to overcome with this problem. We came with the integration of metric tools to get the optimized metric value. Option to select those tools whose license type is free.

Object Oriented Metrics

The metrics presented here are: class related metrics, method related metrics, inheritance metrics, metrics measuring coupling and metrics measuring general (system) software production characteristics. In this paper nine metrics are considered for optimization. These metrics are: DIT (Depth of Inheritance), NOC (Number of Children), CBO (Coupling Between Objects), RFC (Response for a Class), WMC (Weighted Method Complexity), WAC (Weighted Attributes per Class), LCOM-HS (Lack of Cohesion of Methods) (as proposed by Henderson-Sellers), LCOM-CK (Lack of Cohesion of Methods) (as originally proposed by Chidamber & Kemerer).

Software Metric Tool Selection

With the selection of software metrics tools, we limited ourselves to test systems written in Java (source and byte code). SourceForge.NET provides a large variety of open source software projects.

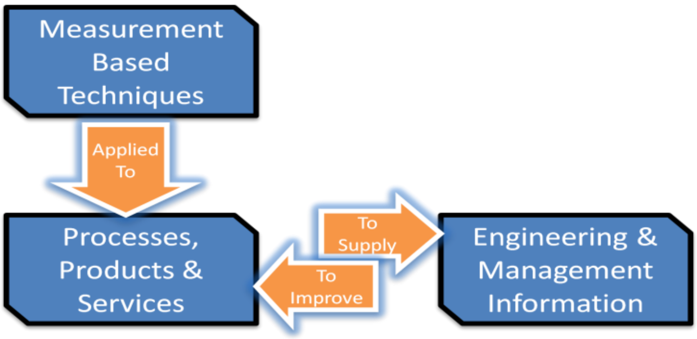

Measurement Based Techniques
Applied To
Processes, Products & Services
To Supply
To Improve
Engineering & Management Information

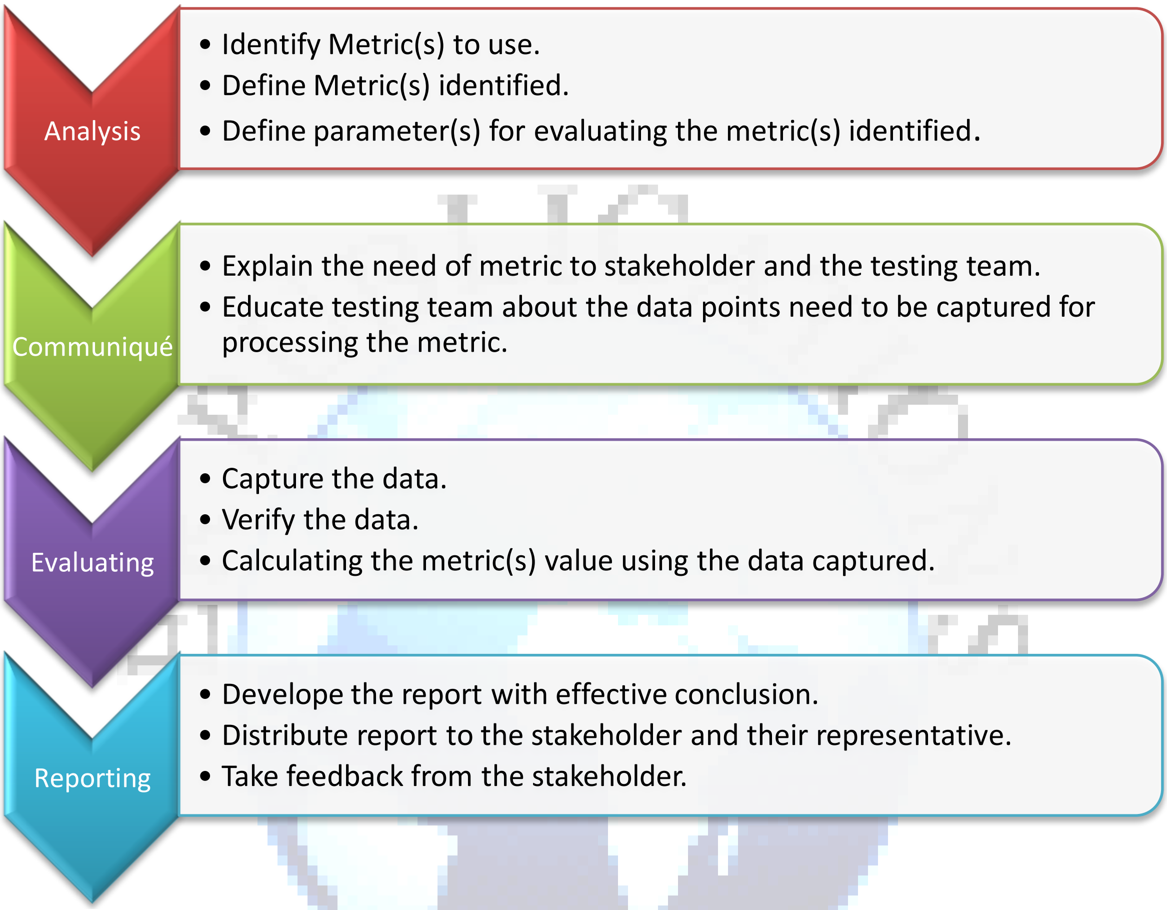
Figure 1: Block diagram for Software Metrics

Metric Selection for Optimization

Six software metrics have been selected for this study. These metrics work on different program entities, e.g., method, class, package, program, etc.The tools and metrics are shown in Table 1. The crosses “#" marks that a metrics can be calculated by the corresponding metric tool. It follows a brief description of the metrics finally selected:

CBO (Coupling Between Object classes) is the number of classes to which a class is couple.

DIT (Depth of Inheritance Tree) is the maximum inheritance path from the class to the root class.



Analysis

- Identify Metric(s) to use.
- Define Metric(s) identified.
- Define parameter(s) for evaluating the metric(s) identified.

Communiqué

- Explain the need of metric to stakeholder and the testing team.
- Educate testing team about the data points need to be captured for processing the metric.

Evaluating

- Capture the data.
- Verify the data.
- Calculating the metric(s) value using the data captured.

Reporting

- Develope the report with effective conclusion.
- Distribute report to the stakeholder and their representative.
- Take feedback from the stakeholder.

Figure 2: Process involved in setting up the metrics

Based on the types of testing performed, following are the types of software testing metrics: -

1. Manual Testing Metrics
2. Performance Testing Metrics
3. Automation Testing Metrics

LCOM-CK (Lack of Cohesion of Methods) (as originally proposed by Chidamber & Kemerer) describes the lack of cohesion among the methods of a class.

$$LCOM(C) = \begin{cases} P - Q & \text{if } P > Q \\ 0 & \text{otherwise} \end{cases}$$

• P = #pairs of distinct methods in C that do not share variables
• Q = #pairs of distinct methods in C that share variables

NOM (Number Of Methods) is the methods in a class.

RFC (Response For a Class) is the set of methods that can potentially be executed in response to a message received by an object of the class.

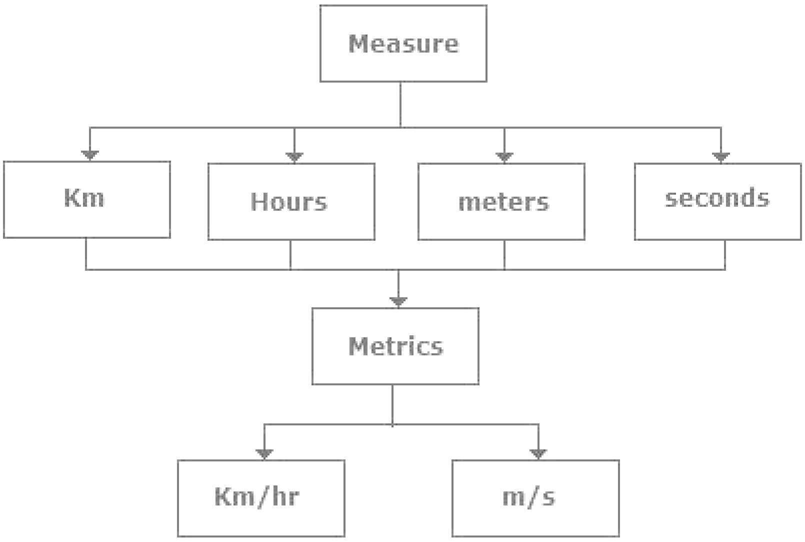


Figure 3: Measurement and Metrics Example

Importance of Software Testing Metrics:

As explained above, Test Metrics are the most important to measure the quality of the software.

Suppose, if a project does not have any metrics, then how the quality of the work done by a Test analyst will be measured?

For Example: A Test Analyst has to,

- Design the test cases for 5 requirements
- Execute the designed test cases
- Log the defects & need to fail the related test cases
- After the defect is resolved, need to re-test the defect & re-execute the corresponding failed test case.

In above scenario, if metrics are not followed, then the work completed by the test analyst will be subjective i.e. the test report will not have the proper information to know the status of his work/project.

Conclusion

A Metric is a quantitative measure of the degree to which a system, system component, or process possesses a given attribute. From the study it is concluded that a new metric tool can be developed which covers metrics values which were emitted before. For more accurate values manual investigation can be done. Since metrics results are strongly dependent on the implementing tools, a validation in terms of manual investigation only supports the applicability of some metrics as implemented by a certain tool. The metrics are used for generating the daily/weekly status report with accurate data during test case development/execution phase & this is also useful for tracking the project status & Quality of the software.

References

[1]. Rüdiger Lincke, Jonas Lundberg and Welf Löwe, “Comparing software metrics tools”, software technology group school of mathematics and systems engineering växjö university, Sweden issue, July 20–24, 2008, Seattle, Washington, USA.
[2]. Shubha Jain, Preeti Katiyar and Prof. Raghuraj Singh,” An integration and optimization of software metric tools”, international conference on advance computing and communication technologies -2013.
[3]. http://cccc.sourceforge.net/. Accessed on 5/11/2013

[4]. http://depfind.sourceforge.net/. Accessed on 6/7/2013

[5]. http://www.spinellis.gr/sw/ckjm/. Accessed on 10/8/2013

[6]. www2.informatik.hu-berlin.de/swt/intkoop/jcse/**tools**/jmt.html

[7]. B. Henderson-Sellers, L. Constantine and I. Graham. “Coupling and Cohesion (To-wards a Valid Metrics Suite for Object Oriented Analysis and Design)”. In proc. Object Oriented Systems, Vol. 3, 1996, pp. 143-158

[8]. Jarallah S. Alghamdi, Raimi A. Rufai and Sohel M. Khan “OOMeter: A Software Quality Assurance Tool” Proceedings of the Ninth European Conference on Software Maintenance and Reengineering (CSMR’05).

[9]. http://www.prosci.com/metrics.htm

[10]. http://www.westfallteam.com/Papers/

[11]. http://www.stpmag.com/downloads/stp-0507_testmetrics.htm

[12]. Kemerer, C. F. “An empirical validation of software cost estimation models”, Comm.ACM 30,5 (May 1987), pp. 416-429.

[13]. W. li and S. henry “Maintenance metrics for the object oriented paradigm”. in proc. software metrics symposium, 1993, pp.52-60.