International Journal of Research (IJR) Vol-1, Issue-8, September 2014 **ISSN 2348-6848**

# Integration of Metric Tools for Software Testing: Including Maintainability Index Measurement

**Viplav Srivastava**
Department of Computer Science
Kanpur Institute of Technology, Kanpur, India
viplav.softech@gmail.com

## Abstract—

*Software metric is a mathematical definition mapping the entities of a software system to numeric measurement values. Furthermore, understand a software metrics tool as a program which implements a set of software metrics definitions. There are numbers of software metric tools available, use different-different methods to assess metric based software systems and hence project different results. The results are thus tools dependent and are in question for validations. Here an attempt is made to integrate five different object oriented free metric tools. A study has been done to measure the metrics values using the same set of standard metrics for a software projects. The results have been discussed before and showed the variations in results from different tools for same metrics. Measurements show that, for the same software system and metrics, the metrics values are tools dependent. This paper will include three more metrics which were not measured while integrating metrics tools earlier. For this focus and study is on integration of JHawk Tool, Analyst 4j and OOMeter Tool with other metric tools.*

***Keywords*—**

**Measurement, Verification, Software product Metrics, maintainability index and Software metric tool.**

## I. INTRODUCTION

Accurate metric 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 users to select the tool best suited for it. Previous papers 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.

## II. OBJECT ORIENTED METRICS

The metrics presented here are: method related metrics, class related metrics, inheritance metrics, metrics measuring coupling and metrics measuring general (system) software production characteristics. In this paper six metrics are considered for optimization. These metrics are: DIT (Depth of Inheritance), CBO (Coupling Between Objects), LCOM-CK (Lack of Cohesion of Methods) (as originally proposed by Chidamber & Kemerer), WMC (Weighted Methods per Class), TCC (Tight Class Cohesion), MI (Maintainability Index).

## III. SOFTWARE METRIC TOOL SELECTION

With the selection of software metrics tools, we limited ourselves to test systems written

in Java (source and byte code) and Eclipse based plug-in. SourceForge.NET provides a large variety of open source software projects and metric tools.

TABLE I
Requirements as a basis for the selection of Tools

| S.No. | Requirements | Type to suite requirement |
|---|---|---|
| 1. | Supporting language | Java. |
| 2. | measuring metrics | object oriented metrics. |
| 3. | license type | freely available. |
| 4. | characteristics | command line tools. |

**The selected tools are listed below:**

### *A. CCCC (C and C++ code counter)*

It is an open source command-line tool and analyzes C++ and Java files and generates reports on various metrics, including Lines of Code and metrics proposed by Chidamber & Kemerer and Henry & Kafura, it is developed by Tim Littlefair of Edith Cowan University.

### *B. CKJM (Chidamber & Kemerer Java Metrics)*

It is an open source command-line tool, it calculates the C&K object-oriented metrics by processing the byte-code of compiled Java files.

### *C. OOMeter*

It is an experimental software measurement tool developed by Alghamdi et al, it accepts Java/C# source code and UML models in XMI and calculates various metrics.

### *D.* Analyst 4j

Analyst4j is based on the Eclipse platform and available as a stand-alone Rich Client Application or as an Eclipse IDE plug-in, it features search, metrics, analyzing quality, and report generation for Java programs.

### *E. JHawk*

JHawk is a Java based open source framework, it compile Java files and calculate maintainability index and other metrics.

## IV. 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 II. The hash "#" 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.

$$CBO = \frac{\text{Number of links}}{\text{Number of classes}}$$

Numbers of links are number of classes used associations, use links for all the package's classes. A class used several times by another class is only counted once. Numbers of classes are number of classes of the package, by recursively processing sub-packages and classes, for the UML modeling project, this variable represents, therefore, the total number of classes of the UML modeling project.

**DIT** - Depth of Inheritance Tree is the maximum inheritance path from the class to the root class.

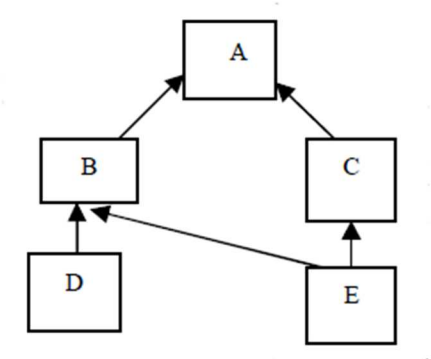

Figure1: Sample measurement of DIT

**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.

**WMC** - Weighted Methods per Class (using Cyclomatic Complexity as method weight) is the sum of weights for the methods of a class. It is an indicator of how much effort is required to develop and maintain a particular class. A class with a low WMC usually points to greater polymorphism. A class with a high WMC, indicates that the class is complex (application specific) and therefore harder to reuse and maintain. The lower limit for WMC in Refactor IT is default 1 because a class should consist of at least one function and the upper default limit is 50.

**TCC** Tight Class Cohesion The Tight Class Cohesion metric measures the cohesion between the public methods of a class.

TABLE II

Tools and metrics used in evaluation

| Tools | Metrics | | | | | |
|---|---|---|---|---|---|---|
| Name | CBO | DIT | LCOM-CK | WMC | TCC | MI |
| CCCC | # | # | | | | |
| CKJM | # | # | # | | | |
| OOMETER | # | # | # | | # | |
| ANALYST 4J | # | # | # | # | | |
| JHAWK | | | | | | # |

NDP – number of pairs of methods directly accessing the same variable.

NIP – number of pairs of methods directly or indirectly accessing the same variable.

NP – number of pairs of methods: n (n-1)/2

Tight class cohesion TCC = NDP/NP

**MI** Maintainability Index to calculate MI value of Cumulative Halstead Length, Effort and Volume is to be calculated.

Cumulative Halstead Length

Is the sum of total number of operators and total number of operands present in the given code.

N=N1+N2

Cumulative Halstead Volume

N is Cumulative Halstead Length

V = N x log n

MI= 171-5.2ln(V)-0.23V(g)-16.2ln(LOC)

Where LOC is Line of Code, ln(V) is Halstead Volume and (g) is Cyclomatic Complexity.

## V. CONCLUSION

Today a large number of software metrics tools exist. But give different values for the same projects and hence none of them have been validated experimentally for the software metric values they measure. Most tools computed different values for the same metrics on the same projects. From the study it is observed 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. All five different object oriented metrics measured by them have been optimized by investigating the results manually.

## VI. REFERENCES

[1] Rüdiger Lincke, Jonas Lundberg, 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] Kemerer, C. F. "An empirical validation of software cost estimation models",Comm.ACM 30,5th May 1987, pp. 416-429.

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

[4] Shubha Jain, Preeti Katiyar, Prof. Raghuraj Singh," An integration and optimization of software metric tools", international conference on advance computing and communication technologies -2013.

[5]http://cccc.sourceforge.net.

[6] http://depfind.sourceforge.net.

[7] http://www.spinellis.gr/sw/ckjm.

[8]www2.informatik.huberlin.de/swt/intkoop/jcse/tools/jmt.html.

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

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

[11]http://www.projectcodemeter.com/cost_estimation/help/GL_maintainability.htm.

[12] http://www.virtualmachinery.com.

[13]Shubha Jain, Viplav Srivastava, Preeti Katiyar, "Integration of Metric Tools for Software Testing" International Journal of Enhanced Research in Science Technology & Engineering, ISSN: 2319-7463 Vol. 3 Issue 5, May-2014, pp: (445-447)

## Authors Profile

Viplav Srivastava received the **B.Tech** degree in Information Technology from Venkateshwara Institute of Technology, Uttar Pradesh Technical University, Meerut, India, in 2011. Currently pursuing **M.Tech** in Computer Science and Engineering from Kanpur Institute of Technology, Kanpur, India. His research interest is in Software Testing.