# Software Metrics and Metric Tools- A Review

Sandeep Kaur
Department of CSE
GIMET Amritsar
Punjab

Navjot Kaur
Department of CSE
GIMET Amritsar
Punjab

***Abstract***— Object-Oriented design is turn out to be more significant in software development environment as stated by the IEEE standard thesaurus of software engineering. It also conclude that software metrics are much more vital in software engineering for determining the software quality characteristics. There are many approaches by virtue of which we can measure the software cost estimation plus predicates on numerous types of deliverable items. Metric tools are used to estimate the measures like lines of code object point, function points etc. This paper highlights mostly the classification of software quality metrics like size metrics, complexity metrics etc. and different metrics tools.

***Keywords-*** *Object-oriented, Metrics, Metric Tools*

*****

## I. Introduction

In software development environment object-oriented design is much more valuable as object-oriented design metrics are vital feature to compute software quality [1].Object-oriented is a categorising tactic which is capable enough to classify the problem in terms of object and it may provide many paybacks on, adaptability, reliability, reusability besides decomposition of problem into small objects and provide some future modifications [2]. Software engineer can easily measure and predict necessary resource and development effort for a project by virtue of using software metrics. Metrics offers a vision towards essential needs for the creation and designing the model through the test. It allows the software engineer to access the quality before the product is built as it provide a quantative means to access quality of inner characteristics of the product [3].

According to the IEEE standards glossary they defined a metrics as an "a quantative measure of degree to which a component, system, or a given attributes" [4]. Metrics are the vital source of information by virtue of which a software developer can takes a decision for design good software. Software metrics can be calculated by metrics tools. Section II provides the deep understanding of software metrics. It highlights the different kinds of software metrics. Section III represents the comparative analysis of the metrics tools available in the software market followed by section IV consisting of conclusion of the paper.

## II. Software Metrics[5,6]

Software metrics are used to measure the software quality to check whether it satisfies the requirements. It helps in project estimation and progress monitoring. Metrics can define as "Quantifiable measures that could be used to measure characteristics of a software system" The main aim of the software metrics is to improve quality, reduce costs, control/ monitor schedule etc. and are used for better understanding the quality of the product and the program. These are useful to create software quality product within budget and time.

### *A. Metrics Categorization*

Metrics can be divided into further two more categories [6] i.e. System Level Metrics and Component Level Metrics

1) System Level Metrics[7]: these consists of following metrics
   a. Quality Metrics: - The metrics which take into consideration the quality as its main motive are known as quality metrics. These metrics help in increasing the quality of the particular system. These are of following types:
      - Adaptability: it provides the system with the ability to adapt to the changes in the environment.
      - Complexity of interface and integration: here the component interface or integration code complexity is measured.
      - Integration Test Coverage: It evaluates the fraction of the system which undergoes integration testing.
      - Reliability: It calculates the probability of the fault free system
      - End-to-end test coverage: The system undergoes end to end testing satisfactorily
      - Customer satisfaction: It calculates the degree to which software meets customer expectations and requirements.
      - Fault Profiles: It detects the number of collective faults.
   b. Management Metrics: - These metrics which defines the amount of management on the particular software components are known as

management metrics. These are related to the cost of the particular system. These metrics play an important role in determining the cost effective component based systems. Metrics used for it are:

- Time-to-market: Time spent between the start of the system development till it is deployed to the users or customers as an application.
- Cost: it evaluates Total software development expenditure, including costs of component acquisition, integration, and quality improvement.
- Software requirement utilization: It measures the reusability of the software components.
- Software Engineering Environment: It measures the capability and maturity of the environment in which the software product is developed.

c. Requirement Metrics: - The requirement metrics depend on the process to product system. It derives out the metrics components needed depending on the requirements supplied. These are:

- Requirement Conformance: It describes the compliance of the integrated component forming a particular system.
- Requirement Stability: It amounts to the level of changes in the software systems checking whether the system ate stable in given conditions or not.

2) Component Level Metrics: - Component-level design metrics deals with the internal characteristics of a software component and include measures of the "three Cs"—module cohesion, coupling, and complexity.

- Cohesion Metrics: These metrics provide an indication of the cohesiveness of a module.
- Coupling metrics: Module coupling represents the degree of independence between modules
- Complexity metrics: Complexity defines the inter relationships in the system between the components. These metrics measure the complexity of a component. Complexity metrics can be used to predict critical information about reliability and maintainability of software systems from automatic analysis of source code [or procedural design information].Some of the mechanisms of the complexity metrics are:
  - ✓ Cyclomatic Complexity
  - ✓ Number of Logical Operators
  - ✓ Essential Cyclomatic Complexity
  - ✓ Maximum Nesting of Control Structures
  - ✓ Estimated Static Path Count

Thus, the metrics play a very important role in determining the benefits and aspects of the component based software engineering systems.

## III. Metrics Tools

These allows to determine a software system according to the metrics by extracting the required entities from the software and providing the equivalent metrics values. These tools are utilized to measure the object oriented paradigms in the program. The set of tools, metrics, and test systems is determined by practical considerations.

Ideally, we had installed all metrics tools accessible, measure a random selection of software systems, and make differentiation of the results for all known metrics. But reality implies yet a number of practical limitations i.e. firstly we cannot quantity apiece metric with each tool, since the selection of implemented metrics changes from tool to tool. Hence, by increasing the set of metrics would reduce the set of comparable tools and vice versa. We need to compromise and select the metrics which appear practically interesting to us.

Secondly, metric tool's availability very is limited, and we cannot know of all available tools. We found the tools by searching through the internet, using the standard search engines and straight-forward search terms. We selected only tools available without legal restrictions and which were meaningful to compare, Finally, we are not able to compare all metrics values of all classes for all systems to a "gold standard" deciding on the correctness of the values. Such a "gold standard" simply does not exist and it is impossible to calculate it since the original metrics definitions are too imprecise. After a pre-analysis of the collected information, we decided to limit the set of tools according to metrics calculated, analyzable languages and availability/license type. We conclude that the majority of metrics tools available can derive metrics for Java programs; others analyze C/C++, UML, or other programming languages. In order to compare as many tools as possible, we prefer to analyze Java programs. But some of them are simple "code counting tools" and others calculate more sophisticated metrics like CK suit, LK suit, Halstead metrics etc. [15]. Furthermore, all tools are not freeware or open source plus commercial versions do not offer appropriate evaluation licenses. Thus our superior criteria motivation on language i.e. Java (source- or byte-code), Our concluding collection left us by means of following software metrics tools:

*A. Analyst4j:*

It is based on the Eclipse platform and available as a stand-alone Rich Client Application or as an Eclipse IDE plug-in. It consists of features i.e. Search, metrics, analyzing quality, and report generation for Java programs [11]

*B. LOC Metric*

This tool is mainly used to measure the total lines of code (LOC), blank lines of code (BLOC), and comment lines of code (CLOC), lines with both code and comments (C&SLOC), logical source lines of code (SLOC-L), McCabe VG complexity (MVG), and number of comment words (CWORDS). Physical executable source lines of code (SLOC-P) are calculated as the total lines of source code minus blank lines and comment lines [8].

*C. Code Counter*

This metrics is mainly used to measure the NOS, COMMENT LINES, TOTAL of the two metrics. The main feature of this tool is, it measure and generate the report for many languages like ASP, HTML, C++, C etc. The main advantage is it will measure many languages and will generate the three forms of reports according to the user needs. The main features of code counter are Easy to use with just 3 steps, Quickly counts several types of source code - including, C or C++, Java, JSP,VB, PHP, HTML, COBOL etc.[11]

*D. Source Monitor*

It is also a freeware software tool which is basically used to check files, lines, branches, classes, complexity, calls, comments, method/class, maximum depth, average depth. It also includes built-in comment and file extension definitions for C, C++ java, html and assembly languages [13].

*E. Code Analyzer*

It is a software source file metrics application [9]. Metrics calculated by code analyzer are Total Files (For multiple file metrics),Total Lines, Code Lines, Comment Lines, Whitespace Lines, Average Line Length Code Lines/File (For multiple file metrics),Comment Lines/File (For multiple file metrics), Code/Comments Ratio , Code/Whitespace Ratio, Code/(Comments + Whitespace) Ratio. It also includes built-in comment and file extension definitions for C, C++ java, html and assembly languages[14].

*F. Eclipse Metrics Plug-in 1.3.6*

It is an open source metrics calculation tools which measures various metrics and detects cycles in package and type dependencies. [10]

Table I Metrics Tools and Their Parameters

| **Metric Tools** | **Parameters measured** |
|---|---|
| **Analyst4j** | LOC,DIT,RFC,WMC,CBO |
| **Loc Metric** | loc, bloc, cloc, c&sloc, sloc-l, mvg, cwords, sloc-p |
| **Code Counter** | Loc, Comment Lines, Total |
| **Source Monitor** | Files,Lines,Branches,Classes,Complexity, calls,comments etc |
| **Code Analyzer** | total Files**,** total Lines**,** Code Lines, comment Lines **,** whitespace Lines , Average Line Length CodeLines/File, CommentLines/File Code/Comments Ratio |
| **Ecilpse Metric Plug-in** | packages, methods, loc, interfaces, attributes etc |

## IV. Conclusion

In this paper we have presented a complete review on software quality metrics and metrics tools which adds an improved understanding of the software metrics. This paper discusses the different quality, management and requirement metrics as system level and cohesion, coupling and complexity metrics as component level metrics respectively. It also offers the comparative analysis of software metrics tools. This paper offers some aid for researchers for well understanding and assortment of software metrics and metrics tools for their own tasks.

## Acknowledgment

I would like to gratefully and sincerely thank to Ms Navjot Kaur for giving his valuable time for guidance in research work and support during the research period. Finally, I honestly thank to my parents, family, and friends, who provide the advice and financial support. The product of this research paper would not be possible without all of them.

## References

[1] C. Neelamegam, M. Punithavali, “A survey on object oriented quality metrics”, Global journal of computer science and technologies, pp 183-186,2011.

[2] B. Henderson, seller, “object oriented metrices:measure of complexity”, Prentice Hall, 1996.

[3] A. Shaik, C.P.K. Reddy, B. Manda, prakashine, K. Deepti, “Metrics for object oriented design software system:A Survey”, Journal of emerging

trend in engineer and applied science (JETEAS), pp 190-198, 2010

[4] R.S.Pressman,”Siftware Engineering-A practioners Approach” Fourth Edition, Mc. Graww Hill International Edition 1997

[5] Sahra Sedigh-Ali Arif Ghafoor and Raymond A. Paul “Metrics for COTS Based Systems”

[6] Norman Fenton & Martin Neil”Software metrics: Roadmap”

[7] Divya Chaudhary , Prof. Rajender Singh Chhillar “ Component Based Software Engineering Systems: Process and Metrics”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7, July 2013

[8] http://www.locmetrics.com/

[9] http://www.codeanalyzer.teel.ws/

[10] http://metrics.sourceforge.net/

[11] http://www.javalobby.org/java/forums/t93556.html

[12] http://sourceforge.net/projects/cccc/

[13] http://www.campwoodsw.com/sourcemonitor.html

[14] http://sourceforge.net/projects/codeanalyze-gpl/

[15] Amit Sharma, Sanjay Kumar Dubey, "Comparison of Software Quality Metrics for Object-Oriented System",International Journal of Computer Science & Management Studies, Special Issue of Vol. 12, June 2012