EMPIRICAL VALIDATION OF QUALITY FOR OSS USING CK METRIC SUITE

[1]Aman Kumar Sharma, [2]Anita Ganpati, [3]Tanuja Gupta
[1&2]Assistant Professor, [3]Research scholar
Computer Science Department, Himachal Pradesh University,
Shimla – India
sharmaas1@gmail.com, anitaganpati@gmail.com, tanugupta489@gmail.com

Abstract*:* *With the rapid advancements in the technology, now software development has became more prominent. Today, software's are used in every field so it is important to develop good quality software. Open Source Software (OSS) is becoming more pervasive as they are easily available on web, reliable, easy to use and are maintainable. Quality is the characteristic of software that differentiate it from others. So it is necessary to have quantitative measurements for assessing the quality of design. These measures allow the designer to access the software early in the process, make changes that will reduce complexity and that will improve the continuing capability of the product. Object Oriented (OO) design metrics is an essential part of software engineering. CK metric suite is selected in order to measure the quality of the OSS software. It has six parameters- Number Of Children, Weighted Methods per Class, Depth of Inheritance Tree, Coupling Between Objects, Lack of Cohesion in methods, Response For Class on the basis of which it measures the quality. The empirical study is done in order to evaluate the quality of OSS i.e. JasperReport and LlamaChat and the results obtained are in accordance with the theoretical results that is low WMC, high DIT, low RFC and high cohesion.*

Keywords: *CK metrics, WMC, RFC, DIT, NOC, LCOM, CBO.*

I. Introduction:

Software quality is crucial to development of software systems. Quality refers to the distinctive characteristic's of object or process that increases its usefulness. Such characteristics may denote some degree of excellence that differentiates the object from others. Reliability, Efficiency, Reusability, Maintainability are some of the different quality factors used to assess quality of a software product [15]. Metrics provide essential means for software practitioners to assess its quality. Software metrics is the measurement of particular characteristics of a program's performance or efficiency. Metrics are the means for attaining more accurate estimation of project milestones and developing a software system that contain minimal faults [1].

Since object oriented system is becoming more pervasive, it is necessary to have quantitative measurements for accessing the quality of designs at both the architectural and component level. The object oriented technology focus on object and its characteristics such as inheritance, cohesion, coupling, polymorphism due to which the traditional metrics such as lines of code, cyclomatic complexity has reduced its effectiveness and found to be inadequate[16] for object oriented software products.

Therefore, the development of new software metrics adapted the characteristics of the object-oriented technology is indispensable. Object -Oriented analysis and design of software provide many benefits such as reusability, easy decomposition of problem, understanding of object and providing aid in future modifications. It is necessary to provide dependable guidelines that one may follow to help ensure good OO programming practices and write reliable code. Object oriented programming metrics is an aspect to be considered, as it is a set of standards acquired which can measure the effectiveness of object oriented analysis technique in the design of the system [11]. With the rise to prominence of the OO paradigm various sets of OO metrics have been proposed as a means of assessing whether system under investigation exhibit characteristics of quality software [3, 4, 5, 9, 10, 12].

This paper is organised in the following manner. Section II reveals the literature based on metric suite for OO design. Section III focuses on analysis of experimental data and Section IV concludes the study.

II. Review LIterature:

Many researchers over a period of time have proposed a number of models for quality measurements such as Chidamber & Kemerer (CK), Metrics for Object – Oriented Design (MOOD), Lorenz & Kidd's L&K) metric suites, etc. Among these the most widely adopted is CK metric suite[16]. The CK suite is proposed by Chidamber and Kemerer [13]. CK Metric is based on Bunge's Ontology as the theoretical basis and analytically evaluated against Weyuker's measurements principles [15]. The measure obtained from CK metric suite helps the users to understand design complexity, to detect design flaws and to predict certain project outcomes, aid in external software quality such as software defects, testing and maintenance effort.[19] Use of the CK set of metrics and other complementary measures are quality growing in industry acceptance.

CK suite metric lacks sometime because it recommence only one level of nesting while calculating RFC and neglects deeply nested call backs. The NOC metric gives distorted view of the system as it counts only the immediate sub classes instead of all the descendants of the class. In some cases it violates the elementary rule of measurement theory that a measure should be concerned with a single attribute [14]. Beside these flaws this metric suite is considered to be a pioneering work in the area of OO design [6, 7]. It is preferred, as it has been generally found suitable for predicting class fault proneness during the early phases of life cycle [16]. It is chosen by SATC (Software Assurance Technology Centre) at NASA Goddard Space Flight Centre and still used widely till now [2].

In keeping with the key elements of object-oriented software, the set of six metrics as proposed by Chidamber and Kemerer in attempt to identify certain design traits in object–oriented software, are inheritance, coupling, polymorphism, encapsulation and cohesion. The six metric can be summarised as [8]:

A. Weighted Methods per class (WMC)

WMC counts the number of methods in a class. WMC was designed to measure the complexity of a class. Weighted Methods per Class is defined as the sum of the complexities of all methods in a class.

WMC= “ C_i= sum of complexities of all classes, where i= 1 to n
If all static complexities are considered to be unity, WMC = n, the number of methods. Higher value of WMC plays a negative role in quality evaluation.

B. Depth of Inheritance Tree (DIT)

Inheritance is when a class shares the behaviour of another class. This metric measures the maximum level of the inheritance hierarchy of a class; the DIT is the maximum length from the node to the root of the tree. DIT was intended to indicate the potential for reuse, and to indicate the complexity of the design.

C. Number Of Children (NOC)

Number Of Children counts number of immediate subclasses subordinated to a class in the class hierarchy. The NOC metric is the total number of descendent classes (subclasses) of a class. Chidamber and Kemerer proposed that it is better to have depth than breadth in the inheritance hierarchy i.e. high DIT and low NOC.

D. Coupling Between Objects (CBO)

Coupling Between Objects (CBO) for a class is the number of other classes to which it is coupled. CBO for a class is a count of the number of other classes whose methods or attributes are used by it. C&K suggest CBO as an indication of the effort needed for maintenance and testing. A high CBO is considered undesirable.

E. Response For a Class (RFC)

This metric counts the occurrence of calls to other classes from a particular class. In other words, it measures the number of local methods and the number of methods called by the local methods.
RFC = |RS| where RS is the response set for the class, given by
RS = $\{M\}\ U_{all\ i}\, \{R_i\}$ where $\{R_i\}$ = set of methods called by method i and {M} = set of all methods in the class. C&K view RFC as an indication of class complexity (and hence a reflection of the testing effort required).The value of RFC can be from 0 to 50 for a class [17].

F. Lack of COhesion in Methods (LCOM)

This metric purports to measure the lack of cohesion in the methods of a class. The LCOM is a count of the number of method pairs of similarity minus the count of method pairs whose similarity is not zero. The LCOM metric is a value of the dissimilarity of the methods in a class. Because of this, it helps to identify flaws in the design of classes. Cohesiveness of methods within a class is desirable since it promotes encapsulation and decreases complexity of objects.

III. DATA ANALYSIS:

Open Source Software (OSS) has gained popularity in recent times due to which JasperReports and LlamaChat are used. Both these software's are stable, accurate, have many versions and are used at number of areas. Further, both are acceptable to market standards. Different versions of both are released time to time to remove bugs and to incorporate new features.

LlamaChat is designed to provide an open source, chat server/client pair for use on the web. It is written in java and supports many advanced chat functionality including secure connections, emoticons, administrative class users, and more [20].

JasperReports is the popular open source reporting engine. It is written in Java and it is able to use data uploaded from any kind of data source and produce pixel-perfect documents that can be viewed, printed or exported in a variety of document formats including HTML, PDF, Excel, Open Office and Word [21].

Ckjm is mean and lean, following the UNIX tradition of doing one thing well. It does not offer a GUI and fancy diagrams (or even an XML output facility), and it calculates only few metrics other than the six ones specified by Chidamber and Kemerer. However, it does this job thoroughly and efficiently [22, 23]. The ckjm tool is used to calculate Chidamber and Kemerer object-oriented metrics by processing the byte code of compiled Java files. In spite of six CK metrics, it also calculate Ca (Afferent coupling, not a C&K metric) NPM (Number of Public Methods for a class, not a C&K metric).

In this study two java based open source software's along with their different versions are used with an assumption that latest version is better than the previous ones. For the purpose of authenticity at least four versions of each of this software's are considered for better accurate results.

The JasperReport [20] and LlamaChat [19] are evaluated using CKJM tool [21, 22]. The software's selected for evaluation is as:

It may be noted that the latest version of both these software's is at serial number 1 in Table 1.

Table1: Different versions of OSS's

S.No.	LLAMACHAT			JASPERREPORT		
	NAME ASSIGNED	VERSION	DATE OF RELEASE	NAME ASSIGNED	VERSION	DATE OF RELEASE
1	Ver4	Llamachat0.8	10/1/2003	Ver4	Jasperreport4.1.1	11/8/2011
2	Ver3	Llamachat0.7	24-12-2002	Ver3	Jasperreport4.0.2	18-04-2011
3	Ver2	Llamachat0.6beta	27-10-2002	Ver2	Jasperreport4.0.0	10/1/2011
4	Ver1	Llamachat0.5	16-10-2002	Ver1	Jasperreport3.6.0	31-08-2009

Different parameters were evaluated of CK suite using the tool.

Table2: WMC values

LLAMACHAT	AVERAGE	JASPERREPORT	AVERAGE
Ver4	6.714	Ver4	17.467
Ver3	7.143	Ver3	20.909
Ver2	7.037	Ver2	20.891
Ver1	9.333	Ver1	21.145

As expected, from the theoretical study the values for both the software's have decreased over time or have remained stagnant over the different versions released of each of this software's as shown in Table 2. Lower the values of WMC less will be the complexity [5, 15].

Table 3: RFC values

LLAMACHAT	AVERAGE	JASPERREPORT	AVERAGE
Ver4	20.476	Ver4	22.536
Ver3	19.444	Ver3	24.936
Ver2	20.429	Ver2	24.848
Ver1	20.476	Ver1	24.924

During this empirical study the values obtained for RFC in Table 3 lies between 0 and 50 which are in accordance with theoretical results depicted for good quality software [17].

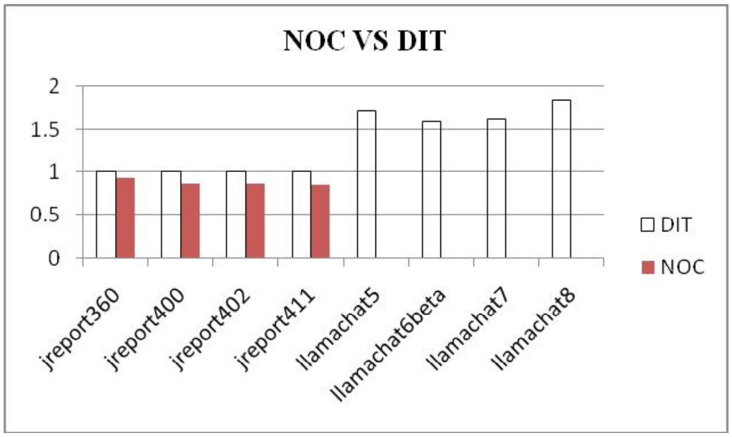

Figure 1: NOC and DIT values

The research [5] has highlighted that DIT values should be more than NOC values as depicted in Figure 1. The reason is that it is better to have depth than breadth in the inheritance heirarchy. Also, deeper a particular class, the greater potential for reuse of inherited methods. The NOC values for Llamachat are zero for all versions.

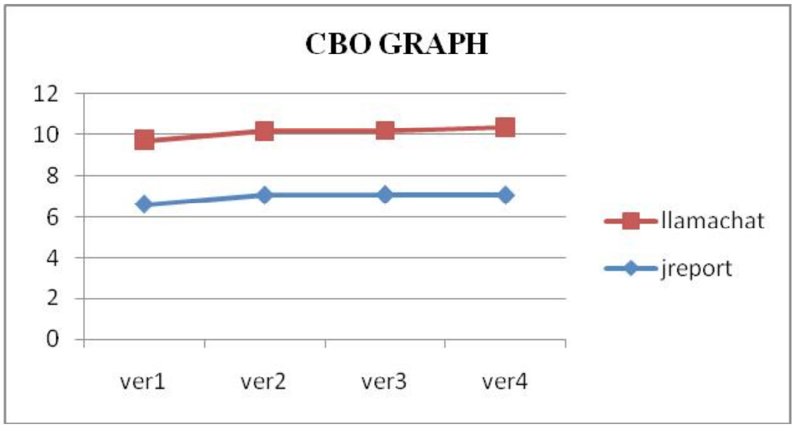

Figure 2: Graph depicting CBO values

Coupling between the classes in a package may increase but the package must have low coupling between other packages in the system[15]. The same is depicted in the Figure 2. The values of CBO as evaluated are of the different classes in a same package. Thereby, as per the theoretical study the same correlates with the results obtained.

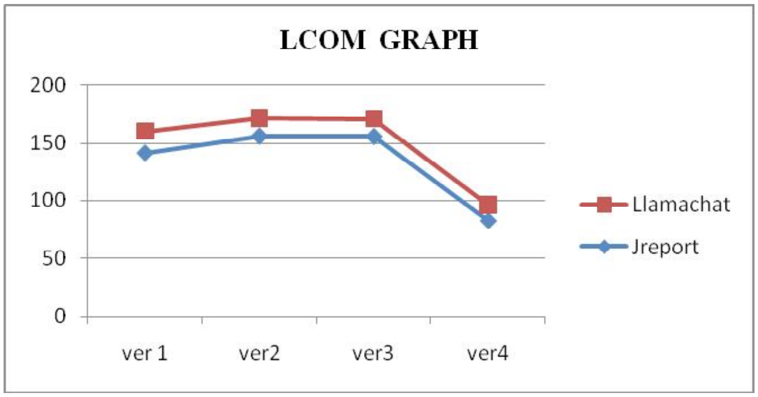

Figure 3: Graph for LCOM metric

Figure 3 shows that the LCOM values for both the OSS's are decreasing with the updated versions. Thus the latest versions in both the cases have less value of LCOM. This result depicts that the latest versions are more cohesive and high cohesion is an important factor contributing towards high quality software [15].

IV. Conclusions and Future Scope:

Thus, it can be concluded that CK metrics is able to prove the higher quality of the latest versions of both the OSS software's. Both the versions have less WMC values, low NOC in comparison to DIT values with respect to the previous versions. CK metric suite also reveals that latest versions RCF values ranging between 0-50 and depicts more cohesive classes, as the LCOM values are deceasing with the updated versions. However, the study is not generalized and more studies are required at a higher level of abstraction in order to make the study generalized.

V. References:

[1] Bellin D., Tyagi Manish., Tyler Maurice, "Object Oriented Metrics: An Overview", Computer Science Department, North Carolina A, T State University, GreenSboro, Ne 27411-0002.

[2] Benlarbi S.K., Emam E.I., Goel N., Rai S., "Threshhold for Object Oriented Measures", Proceeding of ISSRE, San Jose CA, USA, pp 24-37, 2000[3] Brito E., Abreu e., Goulao M., and Esteves R., "Toward the design quality evaluation of 00 software systems", 5th International Conf on Software Quality, 1995.

[3] hidamber S. R., and Kemerer C. E, "Moose Metrics for object oriented software engineering", Workshop on Processes and Metrics for Object Oriented Software Development, OOPSLA '93, Washington, 1993.

[4] Chidamber S. R., and Kemerer C. E, "A metric suite for object oriented design", IEEE Transactions on Software Engineering, pp 467-493, 1994.

[5] Chidamber S. R., and Kemerer C. E, "Towards a Metrics Suite for Object-Oriented Design", OOPSLA'91, Phoenix, Aeizona, USA, pp. 197-211, 1991.

[6] Chidamber S. R., and Kemerer C. E, “AMetrics Suite for Object-Oriented Design”, IEEE Transactions on Software Engineering, Vol.20, No.6, pp. 476-493, June 1994.

[7] Churcher N.I. and Shepperd M.J.,”Towards a conceptual Framework for Object-Oriented Metrics”, ACM Software Engineering Notes, Vol 20, pp 69-76, April 1995.

[8] Fenton N. E., “Software measurement: a necessary scientific basis”, IEEE Transactions on Software Engineering, pp 467-493,1994

[9] Fenton N. E. and P. S. L., “Software Metrics, A Rigorous and Practical Approach”, International Thomson Computer Press, 1996.

[10] Godbole, Nina S, “Software Quality Assurance: Principles and Practices”, Narosa Publishing House, 2, 2004

[11] Harrison R., Counsell S., and Nithi R.,”An Overview of Object Oriented Design Metrics”, IEEE, pp 230-234, 1997.

[12] Jubair, J, Al-Ja’afer, Khan, Eddin, Sabri, M, “Chidember Kemerer (CK) and Lorenze-Kidd(LK) Metrics to assess Java Programs”, King Abdullah II School for Information Technology, University of Jordan, Jordan.

[13] Kaur Amandeep Singh Satwinder, Dr. Kahlan K.S., Dr Parvinder Sandhil S.,” Empirical Analysis of CK and Mood metric suite”,International Journal of Innovation Managment and Technology, vol-1, pp5, Dec2010

[14] Kulkarni U.L., Kalshetty Y.R., Arde Varushali G., “Validation of CK metrics for object oriented design measurement”, Third International Conference on Emerging Trends in Engineering and Technology, IEEE Trans Software engineering, pp 646-651, 2010.

[15] Liu, X., Periyasamy, K., “A New Metric set for Evaluating Testing Efforts for Object-Oriented Programs”, Department of Computer Sciences, University of Manitoba, Winnipeg, Manitoba, Canada.

[16] Refactor IT tool suggest the value of RFC http://www.refactorit.com

[17] Sarkar Muktamyee, Borstler Jurgen, "Software Quality Assurance", Master Thesis, Department of Computer Science, Umea University, Sweden, June 23 (2005).

http://www.scribd.com/doc/30847633/Software-Quality Assurance

[18] Subramenyam Ramanath, Krishnan M.S., "Empirical Analysis of CK metric for Object-Oriented Design Complexity:Implication for software defects", IEEE Transaction an software Engineering, Vol 29, pp 4, April 2003.

[19] sourceforge.net/projects/llamachat

[20] jasperforge.org/projects/jasperreports

[21] www.spinellis.gr/sw/ckjm

[22] sourceforge.net/projects/ckjm