www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 5 May, 2013 Page No. 1653-1656

EMPIRICAL ANALYSIS OF CORRELATION BETWEEN C&K METRICS AND TESTABILITY

Jaspreet kaur, Raj Kumari

student
Panjab University Chandigarh,India

Assistant Professor
Panjab University Chandigarh, India

ABSTRACT
Software quality is very crucial in the development of software systems.Software Metrics help in identifying potential problem areas and finding the problems in those areas decreases the cost. C&K metric suite is one of the well known and most popular metric suites known for measuring the design of object oriented programs. As we know that testing determines the conformance of the software's implementation to its specification so it is one of the important phases in object-oriented systems. This paper focuses on the analysis of the impact of various C&K metrics on testing.

KEYWORDS: Software quality,metrics,CK metric suite, testing.

1. INTRODUCTION

. Software metrics are used by software industry to quantify the development, operation and maintenance of software. Measurement of software plays an important role in the increment of effectiveness of testing process. The quality of the resultant product and software development process are evaluated with the help of various software metrics.

Quality assessment of object-oriented software on the basis of its analysis is becoming more significant progressively .Primarily this is because of the consistent boost in popularity of the standard. On the other hand, widespread embracement of object-oriented software metrics in various application domains should only take place if the metrics can be proved applicable, which means they accurately evaluate the software's attributes they were designed for.

Metrics are very helpful in measuring software's validity,sensitivity,complexity ,quality, estimating cost and project effort etc.

Chidamber and Kemerer (referred to as C&K) proposed one of the first suites of OO design measures in 1991. Particularly, the metrics included were Weighted Methods Per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), Coupling between Object Classes (CBO), Response For a Class (RFC), and Lack of Cohesion in Methods (LCOM). [10]. Use of the C&K set of metrics and other corresponding measures are increasingly rising in industry.

2. C&K METRICS

C&K metrics suite aids users in understanding design complexity of the project as well as detecting design flaws and analyzing certain project outcomes.

Since the proposal of metrics by C&K, several researches have been carried out by other researchers to validate the metrics both theoretically as well as empirically. The six metrics proposed by C&K are as follows:

1. Depth Inheritance tree (DIT) –for a class is the measure of the number of base classes in the inheritance hierarchy (including the *System.Object* class thus DIT >= 1)..

2. Weighted methods per class (WMC) - is the total number of methods implemented in a class where weight of each method is taken as unity.

3. Response for class (RFC) – set of methods that are executed in response to a message received by object of that class.

4. Coupling between Objects (CBO)- is the measure of strength of connection established by a association between entities of various classes.

5. Number of children (NOC) - is the count of number of direct subclasses in class hierarchy which are subordinate to a class. Its values depend upon the Degree of inheritance in classes.

6. Lack of cohesion in methods (LCOM) – gives the degree of similarity of methods that access one or more of the attributes that are same [10].

2. TESTABILITY

The standard definition of testability as defined by IEEE is —the degree to which a system or component facilitates the establishment of test criteria and performance of tests to determine whether those criteria have been met— [13]. In the case of unit testing of object oriented system, the testing for classes brings in some issues. Firstly, a class cannot be tested directly, only an instance of it can be tested and secondly while considering an object in an object oriented system, the state linked with that object also influences the path of execution and methods of a class can communicate amongst themselves through this state. This is the reason of considering the unit testability of the object oriented system with respect to the test case design for unit testing [17]. Testing effort is directly influenced by the time taken to test as greater will be the time spent in testing higher will be the effort which will in turn decrease the testing ability of the programmer to test the software.

3. ANALYSIS OF CORRELATION BETWEEN C&K METRICS AND TESTING TIME

An automated tool Ndepend was used to compute various C&K metrics data and for calculating testing time project analyzer was used. Both these tools perform analysis on C# projects.

We took four projects to analyze correlation between various metrics and their impact upon testing time. The data of various metrics being collected with the help of Ndepend tool is as follows:

Proje-ct Name	WMC	DIT	LCOM	CBO
Payroll manage-ment	216	119	13.69	1054
Model testing	143	255	11.69	699
editor	46	22	1.76	125
Notepa-d	16	8	.89	58

Ndepend tool calculates these metrics on each file in single the project separately so these values are sum of all the values calcluated separately on all files of a project.

The testing time (in hours) is computed with the help of **Project Analyzer** tool for the same projects for which metrics were calculated.

Project Name	Testing Time (in hours)
Payroll management	342
Model testing	339
editor	44
notepad	28

The metric NOC is directly proportional to DIT as greater will be the number of children greater will be the depth of inheritance therefore we considered DIT alone.

From the above data the following relations are derived

DIT ∞ Testing Time

WMC $\propto$ Testing Time

LCOM $\propto$ Testing Time

CBO $\propto$ Testing Time

DIT is directly proportional to testing time as the deeper a class is within the hierarchy, the more the number of methods it is will inherit. Therefore making testing more time consuming. Deeper trees involve more methods and classes which increases the design complexity. This increases the testing effort and decrease the testability. This leads to testability being inversely proportional to DIT.

TE $\propto$ DIT

ITb $\propto$ 1/DIT

WMC is directly proportional to testing time as greater will be the number of methods per class the more will be the time taken by testing.Greater the number of methods involved greater will be the testing effort and decrease testability. This leads to testability being inversely proportional to WMC

TE $\propto$ WMC

ITb $\propto$ 1/WMC

LCOM is directly proportional to testing time. The higher value of LCOM indicates that the methods may be coupled to one another via attributes thereby increasing the complexity of class design and the likelihood of occurrence of errors during the development process. This leads to increase in effort of testing (TE), decreasing the interface testability. Thus, we say that LCOM is inversely proportional to testability.

TE $\propto$ LCOM

ITb $\propto$ LCOM

CBO is directly proportional to testing time. The larger the number of couples the higher will be the sensitivity to change and errors in other parts of design and make testing difficult. This would increase the testing effort (TE) and decrease the testability. Therefore, we say that testability is inversely proportional to CBO.

TE $\propto$ CBO

ITb $\propto$ 1/CBO

4. CONCLUSION

We explored the test case design and testability with the help of C&K metrics suite. The results have shown us that these metrics are useful in measuring testability and the effort of testing. Specifically,our results allow for explanations of the CBO, DIT,LCOM and WMC metrics in terms of test case construction factors. To wind up these results will help us to improve the scenario of testing keeping in mind the impact of these metrics thereby promoting increase testability and decrease in testing time as well as test effort.

REFERENCES

[1] John A. Fodeh and Niels B. Svendsen, Release Metrics : When to stop Testing with a clear conscience, Journal of Software Testing Professionals, March 2002.

[2]http://www.testrepublic.com/forum/topics/1178155:Topic:33849

[3] N.E. Fenton, "Software Measurement: A Necessary Scientific Basis," *IEEE Trans. Software Eng.*, vol. 20, no. 3. pp. 199-206, 1994.

[4]] B.A. Kitchenham, N. Fenton, and S. LawrencePfleeger, "*Towards a Framework for Software Measurement Validation*," *IEEE Trans. Software Eng.*, vol. 21, no. 12, pp. 929 944, 1995.

[5] V.R. Basili, L.C. Briand, and W.L. Melo, "A Validation of Object-Oriented Design Metrics as Quality Indicators," *IEEE Trans. Softwere Eng*., vol. 22, no. 10,

pp. 751-761, 1996.

[6] N.F. Schneidewind, "Methodology for Validating Software Metrics,"*IEEE Trans. Software Eng.*, vol. 18, no.

5, pp. 410-422, 1992.

[7] L. Briand, K. El Emam, and S. Morasca, "Theoretical and Empirical Validation of Software Metrics," ISERN Technical Report 95-03, 1995.

[8] Rachel Harrison and Steve J. Counsell "An Evaluation of the MOOD Set of Object-Oriented Software Metrics*" IEEE Trans. Software Eng.*, vol. 24, no. 6,1998

[9] F.T. Sheldon,K. Jerath and H. Chung, "”Metrics for Maintainability Class Inheritance Heirarchies",J. Softw, Maint. Evol.:Res.Pract.2002;14:11_14.

[10] M. Lorenz and J. Kidd: Object-Oriented Software Metrics. PrenticeHall, 1994

[12] Saida Benlarbi, Khaled El Emam, Nishith Goel. Issues in Validating Object Oriented Metrics for Early Risk Prediction, Accessed on April 2008.

[13] IEEE Standard Glossary of Software Engineering Technology ANSI/IEEE Standard, Washington, DC, USA,2000.

[14] R.S. Pressman, Software Engineering: A Practitioner‘s Approach, McGraw-Hill, 1997.

[15] P. Jalote, An Integral Approach to Software Engineering‖, Spring Verlog, 1997.

[16] S. Chidamber and C. Kemerer: A Metrics Suite for Object-Oriented Design, In IEEE Transactions on Software Engineering, 20(6):476-493, 1994.

[17] Divya Prakash Shrivastava and R.C. Jain, ―Metrics for Test Case Design in Test Drive Development‖, International Journal of Computer Theory and Engineering(1793-8201), Vol.2, No.6, December, 2010