Implementation Of Hybrid Metrics To Evaluate Software Performance Encapsulating CK Metrics

Raj Kumari
University of Engineering And Technology
Punjab University
Chandigarh,India

Jaspreet
University Of Engineering And Technology
Punjab University
Chandigarh,India

Abstract

The software test metrics are the key element of quality driven testing. They help in ensuring that whether the software is appropriate or not. The best metrics suite till now is CK metric suite which covers almost all the features which should be there in a software quality assurance testing but previous researches found some theoretical drawbacks.Our basic aim to implement the hybrid metrics based upon CK metrics and improve the scenario of testing..

1. Introduction

Software measurement can play a key role in increasing the effectiveness of testing process. The quality of the resultant product and software development process are evaluated with the help of software metrics[1][2].

Quality assessment of object-oriented software on the basis of its analysis is becoming progressively more significant. This is mainly because of continuous increase in popularity of the paradigm. On the other hand, extensive embracement of object-oriented metrics in numerous application domains should only take place if the metrics can be proved valid, that is they correctly measure the software's attributes for which they were designed [3], [4], [5], [6], and as well as been validated empirically [5], [6], [7], [8].

Specifically, metrics are useful in measuring software complexity and quality, estimating cost and project effort etc. [9]

Useful metrics are those that are accompanied by accurate data(correct according to the rules of the definition of the metric), exact and consistent which means there is no large difference in the value, even if the person or measuring device changes. [10].

Chidamber and Kemerer proposed one of the firsts suites of OO design measures in 1991. Particularly, the metrics suggested are 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). [11]. Use of the CK set of metrics and other corresponding measures are increasingly rising in industry.

2. CK Metrics

Being one of the first suites of OO design measures the metrics proposed by C&K assist users in understanding design complexity as well as in detecting design flaws and in 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 measures proposed by C&K are as follows:

1. Depth Inheritance tree (DIT) – measure of the depth of the class within the inheritance hierarchy.

2. Weighted methods per class (WMC) - is the total number of methods implemented in a class.

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.

5. Number of children (NOC) - is the count of number of direct subclasses in class hierarchy which are subordinate to a class.

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[11] .

In this research, the developed system is used to assess the design quality of C# programs based on CK metrics. Modifications are made in implementation of the metrics in-order to improve the flaws

3. Hybrid Metrics

Taking CK metrics as the base, the metrics can be improved to produce better results.Thus we proposed hybrid metrics by varying the implementation of some of the CK metrics. Then we test a C# project by CK metrics and Hybrid metrics and show the results.

Various flaws in CK metrics[12] and the suggested implementation method to improve those flaws are:

1. Depth of Inheritance

Flaw: In the previous work done the CK metrics considered the entire piece of code into one block which makes the manual testing easier but automated testing complicated as the compiler will get stuck at a point if memory is blocked.

*Improvement:*This problem can be erased by considering classes into separate files which would make the compiler go smoother.

2. Number of Children

Flaw: The definition of NOC metric did not count all the descendants of the class but only the immediate sub-classes and thus gave the distorted view of system .

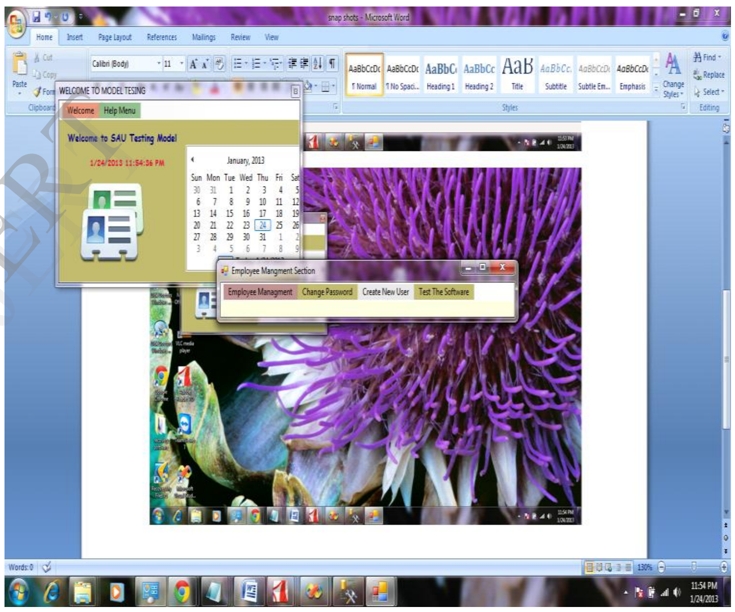
*Improvement:*This can be improved by implementing the concept of polymorphism and multiple inheritance in a link library so that no class would be immediate subclass and no class would be considered as a general class.So the problem of considering only subclasses will be removed.

3.Response For class

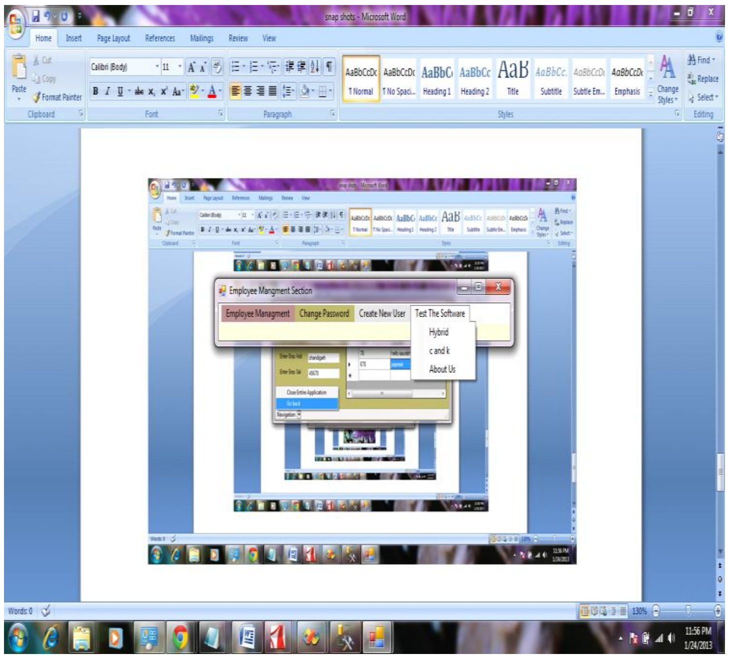
Flaw: For calculating RFC during the data collection C&K recommended only single level nesting. whereas in real programming practice there exists “Deeply nested callbacks” which are not considered here.

Improvement: By the implementation of multiple inheritance the nested callback problem can be removed. As all of the methods will have a reference into a single bind class and the object of that class can be used anywhere in the entire application to get the access of nested classes.

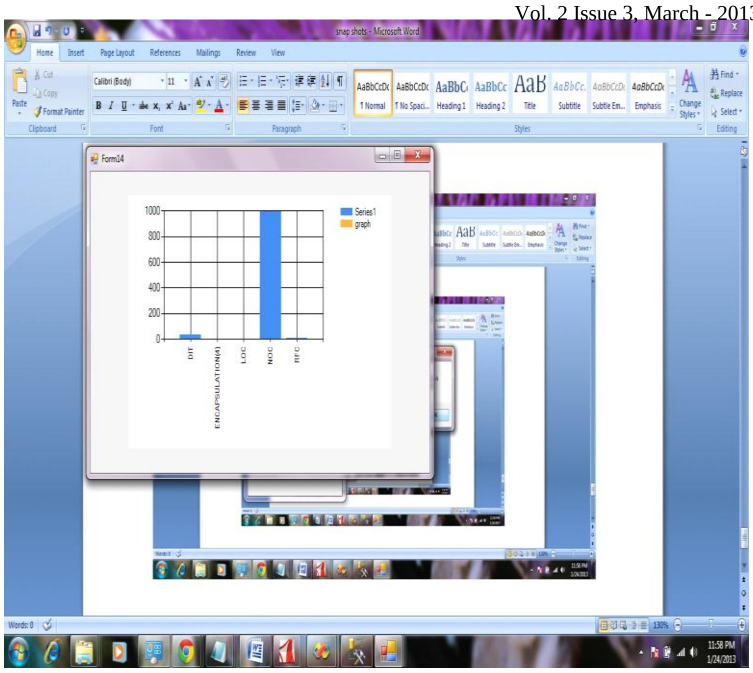
4. Screenshots



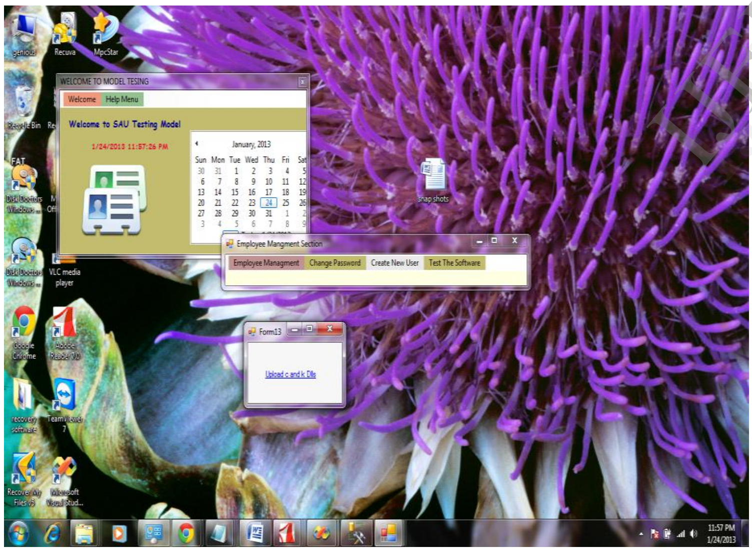
1) Basic software



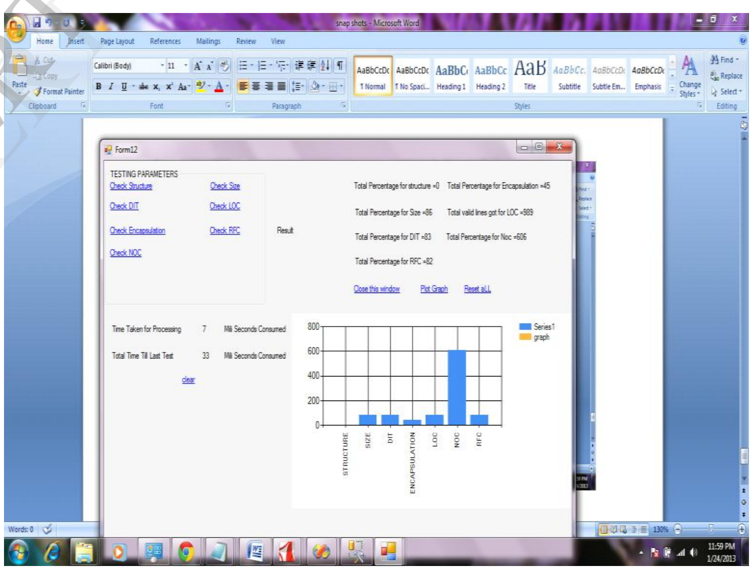
2) Testing the software



3) Results for CK metrics



3) Testing with CK metrics


Form12

TESTING PARAMETERS

Check Structure Check Size
Check DIT Check LOC
Check Encapsulation Check RFC
Check NOC

Result

Total Percentage for structure =0 Total Percentage for Encapsulation =45
Total Percentage for Size =86 Total valid lines got for LOC =989
Total Percentage for DIT =83 Total Percentage for Noc =606
Total Percentage for RFC =82

Close this window Plot Graph Reset aLL

Time Taken for Processing 7 Mili Seconds Consumed
Total Time Till Last Test 33 Mili Seconds Consumed

clear

4) Results for hybrid metrics

5. Conclusion and Future work

Thorough study of OO metrics has led us to wind up that CK metrics cover up every aspect of OOP but few flaws have been observed.The paper demonstrates the flaws as well as well as refinement of the metrics. Further improvement in the metrics can be done in future to make testing more effiiceint .

7. References

[1] Stark, George E; Durst, Robert C; and Pelnik, Tammy M. "An Evaluation of Software Testing metrics for NASA's Mission Control Center" 1992.

[2] Jaspreet Kaur and Raj Kumari, "Survey of software test design metrics," accepted in: 2013

[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] Sheikh Umar Farooq, S. M. K. Quadri, Nesar Ahmad, "Software Measurements and Metrics: Role in Effective s/w Testing", International Journal of Engineering Science and Technology (IJEST).

[11] S.R. Chidamber and C. F. Kemerer, "Towards a metrics suite for object-oriented design", in *Proc. 6th OOPSLA Conference*, ACM 1991, pp. 197-211.

[12] Al-Kilidar, H., Cox, K., Kitchenham, B.: The Use and Usefulness of the ISO/IEC 9126 Quality Standard. International Symposium on Empirical Software Engineering, 7 p. (2005) .