International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

General Article

A Systematic Analysis on MOOD and QMOOD Metrics

Bhavna Katoch[Á*] and Lovepreet Kaur Shah[Á]

[Á]Lovely Professional University, Punjab india

Accepted 15 March 2014, Available online 01 April 2014, **Vol.4, No.2 (April 2014)**

Abstract

Measuring all the essential factors is the evidence of a good software. Maintainability is one of the important quality factor quantified by various Object Oriented Metrics. This paper delivers a review on Quality metrics –MOOD and QMOOD with their quality factors. These widely used metrics improve the maintainability of class diagram and will improve maintainability of final product and reduce cost. This paper also cover desirable features of MOOD and QMOOD metrics. Result shows the comparison between these two metrics.

Keywords: *Quality Metrics, Maintainability, MOOD, QMOOD*

1. Introduction

In Recent years, the productivity of software are growing in size, complexity and also in terms of cost. In huge system , the software life cycle cost exceeds with 80-90% of the total system cost going in software maintenance. For other system, Maintainability factor contributes 40-70% of the cost of software production. The reason is that to modify the delivered program to meet the changing need and growing need of the customer (Software Design for Maintainability, Technique). There are number of metrics, so the metrics that satisfy the quality characteristics must be used.

It helps an engineer to assess software early in the process, making changes that reduces complexity and improve long term usefulness of the end product (Pressman,R.S , 2001).

Software Maintainability is one of the key characteristics of the software system. To ensure quality of Object Oriented Software, Researchers developed many metrics such as Chidamber and Kemerer metrics(C & K), Metrics for Object Oriented Design (MOOD), Quality Model For Object Oriented Design(QMOOD) and Lorenz and Kidd. In this article ,we theoretically validate the MOOD Metrics and QMOOD Metrics and will discuss about the various features of MOOD and QMOOD metrics. The comparison among these Metrics also shown.

2. Literature Review

Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD)This paper discuss about development of Maintainability Model For Object Oriented Software in Design phase. In this paper , two multivariate models are developed. These models are Understandability and Modifiability Model. The development of Maintainability Model uses the development of Understandability Model and Modifiability Model. Maintainability of the overall system is increased by adding two new factors understandability and modifiability into it. These models have developed by using the technique of multiple linear regressions (S. W. A. Rizvi and R. A. Khan ,2010). Thus early measure of maintainability provides an opportunity to improve the maintainability of class diagram and therefore maintaining the final software product. The values of understandability, modifiability and maintainability are used in the software development process. The three values help software designer to review the design and helps to quantify the corrective measures to enhance the maintainability of overall product.

Testability Estimation of Object Oriented Design In this paper, author describes the testability. Testability is one of the most important factors to indicate quality of the software. It is used to correct the measurement of the software metrics, and used to improve the test processes. The accurate measure of software quality depends on testability measurement, and as a result estimating efforts in measuring testability is a complex problem (Abdullah, Dr. Reena Srivastava, Dr. M.H. Khan,2013). To solve this problem, author uses the various methods. In this paper, author shows the results of a systematic review, that is conducted to collect evidence on software testability estimation of object oriented design. The aim of this paper is to find the existing known comprehensive and complete model or framework for evaluating the testability of object oriented design at an initial stage.

A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD)This paper discusses about the maintainability of the software project. Maintainability is a effort required to locate and fix a fault in the program within its operating environment. In this

*Corresponding author: **Bhavna Katoch**

paper, author tries to solve the maintenance problem by adding two new factors in the MOOD metrics (Kiranjit Kaur, Sami Anand Asst. Prof.(L.P.U.),2013). These two factors are: portability and reliability. These factors may reduce the maintenance cost of the software project. In the purposed methodology, author adds two new factors in to Maintenance model which reduces the maintenance cost of software project.

A Maintainability Estimation Model and Tool The given paper discusses a multivariate linear regression for the establishment of the maintainability estimation model in terms of two new factors flexibility and extendibility. The two new factors calculate from computation formulas. Thus quantify maintainability model for the class diagram and develop the Maintainability Estimation Tool or MET. This tool help a software designer for the improvement of the maintainability of class diagram during design phase and hence helps to reduce the increasing cost of software maintenance phase (Alisara Hincheeranan and Wanchai Rivepiboon ,2012).This tool calculate the Object Oriented Design Metrics in terms of Object Oriented Concepts. The MET tool helps a software designer for the evaluation of maintainability of software system early in software development life cycles, may help software developers for improving the maintainability of software system before delivering it into the market for further use.

3. MOOD Metrics Suit

A. Introduction

MOOD Metrics is Metrics for Object Oriented design which is designed by Fernando Brito e Abreu in the year 1996 to increase the quality of Object Oriented Software. In the existing maintainability model basic quality factors are there. These factors are given in Figure1.

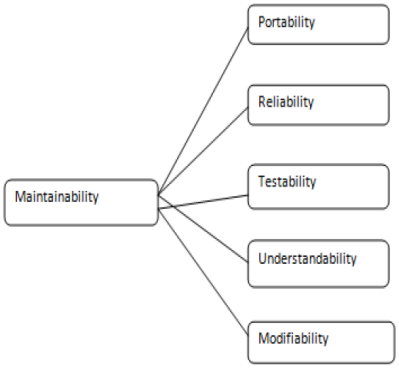

Fig.1. Existing Maintainability Model

MOOD metrics include six metrics

a. **MHF(Method hiding Factor)** is defined as the division of the addition of all the invisible methods defined in all classes with all the methods under consideration. The invisibility of a method is the percentage of the total classes from which this method is not visible.
b. **AHF(Attribute Hiding Factor)** is defined as the division of the addition of all the invisible attributes defined in all classes with all the of attributes defined in the system under consideration.
c. **MIF(Method Inheritance Factor)** is defined the division of the addition of the inherited methods in all the classes of a system under consideration to the total number of available methods for all classes.
d. **AIF(Attribute inheritance Factor)** is defined as the division of the addition of inherited attributes in all classes of the system under consideration to the total number of available attributes for all classes. A class which inherits lots of methods /attributes from its ancestor classes helps in high MIF /AIF. A child class that redefines its ancestors' methods /attributes and adds new ones will leads to lower MIF /AIF (Mood and Mood2 metrics).
e. **PF:** Polymorphism Factor is equals the number of actual method overrides divided by the maximum number of possible method overrides.
f. **CF:** Coupling Factor is the division of the actual couplings among classes to the maximum number of possible couplings.

If A calls methods of B then Class A is coupled to class B. If B calls methods of A then B is coupled to A. B is not coupled to A, if there is no call from B to A.

MOOD metrics evaluate how Object Oriented concept defines on MOOD like inheritance , polymorphism, information hiding and coupling can make influence on quality characteristics like reliability and maintainability.

B. Features of MOOD Metrics

- MOOD metrics allow replicability. This means different people at different times or places can get the same values while measuring same systems.
- These metrics are also expected to be system *size independent.*
- The MOOD metrics definitions make no reference to specific language constructs. However, since each language has its own constructs that allow for implementation of OO mechanisms in more or less detail, a binding for the case studies language (C++) is included ahead8. Similar bindings will be proposed for other OO languages in the near future (Fernando Brito e Abreu (INESC/ISEG) Miguel Goulão, Rita Esteves (INESC/IST),1995).
- Useful in Projects with heavy use of Object Oriented programming
- In MOOD metrics model, there are two main features are methods and attributes. Attributes are used to represent the status of object in the system and methods are used to maintained or modifying several kinds of status of the objects (N. Fenton et al,1996).
- Improving the OO design process to achieve better maintainability and reliability
- Improving the OO estimation process to achieve better resource allocation
- Expresses the result in Percentage

4. QMOOD Metrics Suit

A. Introduction

Quality Model for Object Oriented Design was introduced by Bansiya and Davis(J. Bansiya and C. G. Davis 2002) in the year 2002 and consists of a hierarchy of four levels and three mappings. The QMOOD metrics set used to Quantify quality attributes like effectiveness, Understandability, Extendibility, Reusability and Functionality.

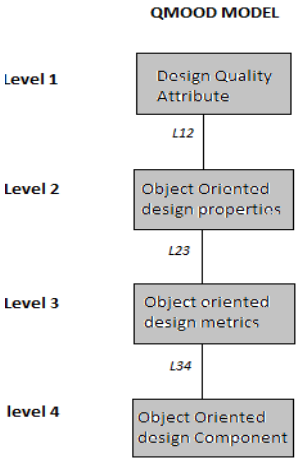

Fig. 2 The QMOOD Model

QMOOD set contain 11 design properties with related metrics as shown in Table I

Table I Design Metrics for Design properties

Design property	Metrics
Design Size	Design Size in Class(DSC)
Hierarchies	Number of hierarchies(NOH)
Abstraction	Average Number of Ancestors(ANA)
Encapsulation	Data Access Metrics (DAM)
Coupling	Direct Class Coupling(DCC)
Cohesion	Cohesion Among Methods in Class(CAM)
Composition	Measure of Aggregation(MOA)
Inheritance	Measure of Functional Abstraction(MFA)
Polymorphism	Number of Polymorphic Methods(NOP)
Messaging	Class Interface Size(CIS)
Complexity	Number of Methods(NOM)

Table II Computation formulas for quality attributes

Quality Attribute	Index Computation Equation
Reusability	-0.25* Coupling+0.25*coupling+0.5*Messaging+0 .5*Design Size
Flexibility	0.25*Encapsulation-0.25*Coupling+0.5*Composition+0.5*Poly morphism
Understandability	-0.33*Abstraction+0.33*Encapsulation-0.33* Coupling+0.33*Cohesion-0.33*Plomorphism-0.33*Complexity-0.33*Design Size
Functionality	0.12*Cohesion+0.22+Polymorphism+0.22* Messaging+0.22*Design Size+0.22*Hierarchies
Extendibility	0.5*Abstraction-0.5*Coupling+0.5*Inheritance+0.5*polymo rphism
Effectiveness	0.2*Abstraction+0.2* Encapsulation+0.2*Composition+0.2*inher itance+0.2*Polymorphism

The Quality attributes of QMOOD Metrics set given in Table II are quantified by the given Equations.

B. Features of QMOOD Metrics

- It helps to access Object Oriented design property
- Provides search-based refactoring
- Quantify Quality Attributes with the help of equations

Conclusion

MOOD metrics expresses the result in percentages and based on set theory. The MOOD metrics are most suitable metrics to assess object oriented programs and proved successfully used to assess Java Program Whereas QMOOD defines relation between quality attributes and design properties with the help of equations given above. QMOOD is an comprehensive model. And is one of the refinements of Object Oriented Metrics. In order to implement the QMOOD metrics for replicable studies, it is necessary to define them more precisely but not in the case with MOOD Metrics. As QMOOD use in search-based refactoring whereas MOOD Metric does not provide complete quality model and not suitable for search-based refactoring. In such way, selection of Metrics depends upon requirement of quality characteristic. In the future, work can be done by adding some more quality attributes in both the Metrics Model.

References

Software Design for Maintainability,Technique DFE-6,pp.-1-6.

Pressman,R.S(2001),Software Engineering-A Practitioner's Approach, *The McGraw-Hill ,Fifth Edition*.

S. W. A. Rizvi and R. A. Khan (2010), Maintainability Estimation Model for Object-Oriented Software in Design Phase (MEMOOD) *Journal Of Computing*,2(4),pp. 26-32.

Abdullah, Dr. Reena Srivastava, Dr. M.H. Khan(2013), Testability Estimation of Object Oriented Design: A Revisit, *International Journal of Advanced Research in Computer and Communication Engineering*,2(8),pp.-3086-3090

Kiranjit Kaur, Sami Anand Asst. Prof.(2013), A Maintainability Estimation Model and Metrics for Object-Oriented Design (MOOD), *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 2(5),pp.-1841-1843.

Alisara Hincheeranan and Wanchai Rivepiboon (2012), A Maintainability Estimation Model and Tool , *International Journal of Computer and Communication Engineering* ,1(2),pp.-143-146.

Mood and Mood2 metrics: http://www.aivosto.com/project/help/pm-oo.html

Fernando Brito e Abreu (INESC/ISEG) Miguel Goulão, Rita Esteves (INESC/IST)(1995), Toward the Design Quality Evaluation of Object-Oriented Software Systems, *5th International Conference on Software Quality*,pp.-1-12

N. Fenton et al(1996), Software metrics: a rigorous and practical approach, *International Thomson computer press*

J. Bansiya and C. G. Davis(2002),A hierarchical model for object-oriented design quality assessment, *IEEE Transaction on software engineering*, 28, pp. 4-17