Assessment of Quality Software Using Correlation Analysis

Ei Phyo wai, May Phyo Wai, Thiri Haymar Kyaws
Computer University (Pathein), Myanmar
wai.eiphyo@gmail.com,mayphyooo@gmail.com, thirihaymarkyaw@gmail.com

Abstract

Software quality is the conformance of explicitly documented standards, and explicitly stated functional and performance requirements and implicit characteristics that are expected of the all professionally developed software. Software quality is a complex mix of factors that will vary across different applications and the user's requests them .The quality of software has improved from time to time because new productivity has promoted the research on software technology. Object-Oriented technology can provide software product with higher quality and lower maintenance costs. Object-Oriented development requires not only different approaches to design and implementation but also different approaches to software metrics. Accordingly, Object-Oriented metrics play an important role in object technology as well as in software engineering. The quality measurements of the process and products of software development projects such as reusability, maintainability, complexity, etc can be predicted due to the merits of software metrics. In facts, we propose how to assess the software quality by using the Object-Oriented software metrics and to show the relationships between this Object-Oriented metrics.

Keywords: Object-Oriented technology, Object-Oriented metrics.

1. Introduction

Many Object-Oriented metrics have been proposed over the last decade. A few of these metrics have undergone some form of empirical validation, and some are actually being used by a number of organizations as part of an effort to manage quality. This is encouraging as such estimates can be used to allocate maintenance resources, and for obtaining assurances about software quality [7].The aim of Object-Oriented metrics is to predict the quality of the Object-Oriented software product. Various attributes, which determine the quality of the software, include maintainability, fault-proneness, understand-ability, reusability.

Object-Oriented metric may be either class-oriented or operation-oriented. At the class-oriented level, the metrics are used to assess the collaboration between classes and the cohesion of methods that reside within a class. Such metrics focus class-size and provide degree of interaction between classes. At operation-oriented level, the metrics are developed to measure the relationship between the methods. Such metrics focus on the length and operations. However, the primary thrust for object-oriented metrics is at the class –level [2, 4].

This paper is organized by colleting of six sessions. The sessions 1 is Introduction and the related works with my paper are stated in session 2.In session 3, proposed system framework and the used software metrics is represented. Implementation of the system is shown in session 4.Conclusion of the system is presented in session 5 and references in session 6.

2. Related Works

S.R Chidamber and C.F Kemerer [7] presented Object-Oriented metrics for Object-Oriented designs, described the evaluations these metrics, and considered the basic components of the Object-Oriented paradigm. McCabe & Associated, McCabe [6] proposed traditional metrics and Object-Oriented metrics .It showed Object-Oriented metrics that are used for a clear visualization and design architectures. Object-Oriented tools can help to support the features the migrate system from traditional language to Object-Oriented language.

L.Brand, K.E Eman and S. Morasca [4] presented the estimations of product metrics and associated their quality measures. L.J Kazmier [1] presented the statistics calculations with computer application. There are two statistics analysis, basic statistics and correlation analysis in the proposed system. The basic statistics are mean, deviation. Mean calculation in the basic statistics are used in correlation analysis. The

relations between seven metrics are calculated by using correlation analysis.

S.Z.Aung [8] proposed the relation-ships between Object-Oriented metrics and how to relate the Object-Oriented metrics and reduced suite metrics.

3. Proposed System Framework

With Object-Oriented (OO) analysis design, it is the time to investigate OO metrics to predict the respective software quality. Among the many metrics models, OO metrics model is popular. The OO software metrics may be either control metrics or predictor metrics. Control metrics are usually associated with software process. Example of control metrics is the average effort and time required to repair defects. Predictor metrics are associated with software products. Example of predictor metrics are the complexity of a module, and the number of attributes and operations associated with objects in a design. Product metrics are classified either dynamic metrics or static metrics. Dynamic metrics are collected by measurements made of a program in execution. Static metrics are collected by measurements made of representations of the system such as the design, program or documentation.

In this section, our proposed system framework is the assessment of quality software by using seven OO metrics and for presenting how to relate between metrics by using Correlation analysis.

3.1 Relations between Object-Oriented mechanisms and the Object- Oriented metric

source	Metrics	OO structures
Object-Oriented	WMC	Methods
Object-Oriented	CBO	Coupling
Object-Oriented	DIT	Inheritance
Object-Oriented	NOC	Inheritance
Object-Oriented	NPV	Attributes
Object-Oriented	NPM	Methods
Object-Oriented	NOV	Attributes

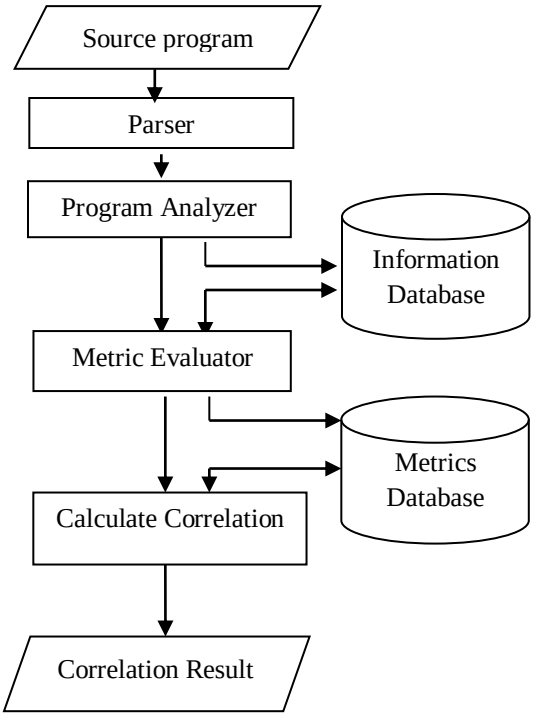

Figure 1. Overview of Proposed System Design

3.1.2 Some Object-Oriented Key Terms

- Attributes: it defines the structural properties of classes, unique within a class, generally a norm.
- Class: it encapsulates data structure and operations defined for manipulation of the data structures. The data structures are the data attributes of a class.
- Methods: it is an operation upon an object, defined as part of the declaration of a class.
- Coupling: it is defined like that object X is coupled to object Y if only if X sends a message to Y.
- Inheritances: it is a relationship among classes; where in an object in a class acquires characteristics from one or more other classes.
- Object: it is an instance of class which is able to save a state and which offers a

number of operations to examine or affect this state.

- Message: it is a request that an object makes of another object to perform an operation.

3.2 Measurements of Object-Oriented Metrics

The proposed seven OO design metrics are described in the following:

Metric 1: Weighted Methods per Class (WMC)

Consider a class C with methods M1…Mn that is defined in the class. Let c_1,…,cn be the complexity of the methods . Then

$$WMC = c_1+c_2+\ldots+c_n; \quad (1)$$

If all methods complexities are considered to be unity, then WMC = n; n is the number of methods implemented within a class or the sum of the complexities of the methods. The larger the number of methods in a class, the larger the potential impact on children since children will inherit all the methods in the object. A larger number of methods can result in a too application specific object, limiting the possibility of reuse [3, 6, 7].

Metric 2: Coupling Between Objects (CBO)

CBO = a count of the number of other classes to which it is developed. (2)

The larger the number of the couples, the harder maintenance is. This metrics can evaluate reusability and maintainability [2, 7].

Metric 3: Depth of Inheritance Tree (DIT)

DIT = the maximum length from the node to the root of the tree. (3)

Inheritance is when a class shares the same structure and behavior defined in another class. When a subclass inherits from one super class it is called a single inheritance and when a subclass inherits from more than one super class it is called multiple inheritance. Inheritance through classes increases it efficiency by reducing the redundancy. But the deeper the inheritance hierarchy, the harder to understand it is behaviors. This metric can evaluate reusability and understandability [3, 6, 9].

Metric 4: Number of children (NOC)

NOC = the number of immediate subclass subordinated to a class in the hierarchy (4)

NOC is an indicator of the level of reuse in a system and the level of testing required. The greater the number of the children in an inheritance hierarchy, the greater the reuse, since inheritance is a form of reuse. The metric can evaluate reusability [3, 4, 6].

Metric 5: Number of Public Methods (NPM)

NPM = the number of public methods in a class (5)

If the number of public methods per class gets too large, extensibility will be hard; the class objects tend to have much more functions [4].

Metric 6: Number of Variables per Class (NOV)

NOV = the total number of variables in a class (6)

The total numbers of variable metric include public, private and protected variables. If the number of variables per class gets too large, the class needs to provide much more information to other classes or within same class, and so maintenance more difficult [4,10].

Metric 7: Number of Public Variables (NPV)

NPV = the number of public variables in a class (7)

The facts that one class has more public variables than another might imply that the class has more relationship with other objects and as such, is more complex [6].

3.3 Related Object-Oriented metrics and their guidelines

metrics	Objectives	Descriptions
WMC	LOW	Reusability is limited. Complexity is low.
CBO	LOW	Maintainability is easy. Reusability is hard.
DIT	LOW	Understandability is easy. Reusability is hard.
NOC	LOW	Reusability is difficult. Maintainability is good.
NPM	LOW	Extendability is easy.
NPV	LOW	Simplicity is good.
NOV	LOW	Maintainability is good.

3.4 Statistical Analysis

Statistics refer to the distribution being used in the analysis of data. For each and every one of the metrics, the maximum, minimum, mean and deviation are calculated on every source code.

3.4.1 Evaluation of Mean

The arithmetic mean is synonymous with the average and is computed by summing all numbers and dividing by the number of numbers. The sample mean is represented by $\bar{x}$. N is the number of data, x is the actual data.

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i}{N} \qquad (8)$$

3.4.2 Correlation Analysis

The correlation between two variables is the degree to which there is a linear relationship between them. The most common measure of correlation is the Pearson's correlation. These methods produce a correlation coefficient, r, which tells the degree and magnitude of the relationship between two sets of data. The magnitude, r, is between 1and -1, where values near -1 are weak correlations and values near 1 are strong correlations. The sign of r tells the direction of the relationship. A negative r denotes an inverse correlation and a positive r denotes a direct correlation [1].

Hence, given two variables X = $\{x_1, x2,\ldots,x_n\}$ and Y = $\{y_1,y_2,\ldots,y_n\}$, the Pearson Correlation Coefficient is defined as

$$r = \frac{\sum_{i=1}^{n} X_i Y_i - n\bar{X}\bar{Y}}{\sqrt{(\sum_{i=1}^{n} X_i^{\,2} - n\bar{X}^2)(\sum_{i=1}^{n} Y_i^{\,2} - n\bar{Y}^2)}} \qquad (9)$$

Where $\bar{X}$ and $\bar{Y}$ are means of X and Y, respectively.

4. Implementation of the Proposed System

The parser of the system parses the treemap.jpx, test java program. This program consists of 64 classes, 30 of them are inheritance classes. There are 112 variables and 220 methods. There are 464 methods calls in this program. Our parser parses this treemap program. The parsed information, program information, is stored in information database. The information about classes, attributes, methods, methods call, and inheritance are reached into the information database as class tables, attributes tables, etc.

The metrics analyzer analyses the measurements of the proposed Object-Oriented metrics. After measuring the metrics, the metrics result comes out. This is shown in figure 2.

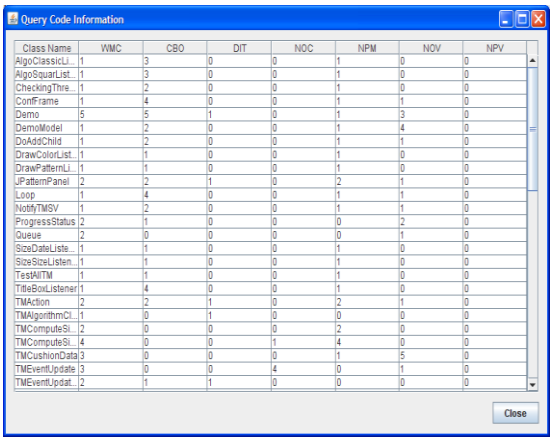

Figure 2. Metrics Table

Then the correlation analyzer calculated the relationships between these metrics. If the correlation value is near -1, this shows weak correlation between theses metrics and if near 1, this shows strong correlation. The sign (+,-) indicates direct or indirect relationships. The correlation result is shown in figure 3.

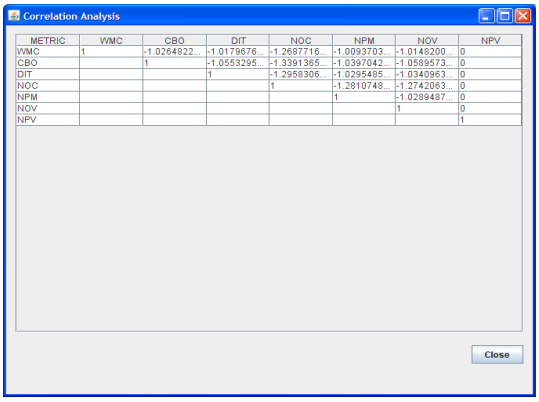

Figure 3. Correlation Table

5. Conclusion

In this paper, a framework has been presented for assessment of quality software using correlation analysis. The system is the software measurements with the help of Object-Oriented metrics, WMC, CBO,DIT, NOC, NPV, NPM, NOV that are evaluated using the software system. Software quality is also evaluated by using Object-Oriented metrics in the form of an equation. The correlations between these metrics are calculated with the help the correlation formula. By estimating the result, the quality of software such as reusability, reliability, understandability…etc is predicted. The future work can be exploded by adding another Object-Oriented metrics such as cohesion. The fault - pronesses prediction capabilities of the metrics will be investigated. And then the relations between the metrics and size metrics can be measured.

6. References

[1] L.J.Kazmier, "Theory and Problem of Business Statistics with Computer Applications", 2nd Edition, 1998.

[2] Lee,Y.,b.Laing and F.Wamg , " Some Complexity Metrics for Obkect-oreinted programs Based on Inforamtion Flow " Proceeding Computer ,March 1993,PP 302-310.

[3] Lorenk ,Mark and Jeff kidd, Object-oriented Software Metrics, Prentices Hall, Englewood Cliffs,N.J,1994

[4] L.Brand, K.E Eman and S. Morasca,"Theoretical and Empirical Validation of Software Product Measures," ISERN Technical report 95-03,1995.

[5] Hudi,R.C.Hoskins and A.Huill, "software Metrics for Object-Oriented design ,"IEEE,April,1994.

[6] McCabe & Associated, McCabe. "Object-Oriented Software Metrics", Prentice-Hall, Englewood Cliffs, N.J, 1994.

[7] S.R Chidamber and C.F Kemmerer, "Towards A Metrics Suite for Object-Oriented Design", Proceeding

[8] Swe Zin Aung, " A Study of Correlation between Object-Oriented Design Metrics and Size Metrics" Proceedings of 3rd International conferences on Computer Application 05, March9-10, Myanmar, PP 745-751.

[9] Seyyed Mohsen Jamali, " Object-Oriented Metrics" Department od Computer Engineering, Shariff University of Technology, January2006, Tehran Jran.

[10] S.Hemry and F.Kafura,"The Evaluation of Sotware System 's Structure Using Quantitative Software Metrics",Software Practice and Fxperience14,PP 561-573 30,000 lines of C code, June 1984.