IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

A Software Complexity Prediction Model using Coupling Metrics: A Statistical Analysis

M. Kavitha[1], Dr. S.A. Sahaya Arul Mary[2]

Research Scholar, Dept. of Computer Science, Manonmaniam Sundaranar University, Tirunelveli[1]

Head of CSE, Saranathan College of Engineering, Trichy[2]

Abstract: OO programming has become the most popular technology in software development environment as it is been proven that the maintenance of OO software is comparatively lesser than the other programming languages. But still the burden of software maintenance is not completely eradicated. One popular software maintenance approach is the reduction of software maintenance cost by imposing the software evaluation metrics during the development phase of the life cycle. Software metrics helps in identifying the potential problem areas in the code. Many novel metrics have been proposed and only few are validated. The objective of this research is to experimentally explore the two novel OO coupling metrics namely Subclass Coupling Factor (SCF) and Temporal Coupling Factor (TCF) to evaluate their ability to predict the complexity of the built software through statistical validation.

Keywords: OO metrics, software maintenance, SCF, TCF and software complexity.

I. INTRODUCTION

Software metrics have been the part and parcel of the software evaluation process for assessing the quality of any software before its delivery. Software may assessed by variety of quality attributes such as complexity, maintainability and reusability. Among them, complexity and maintainability plays a vital role for assessing the worthiness of software as it increases the scope of future enhancements. The more the complexity of the software increases is the more the scope of software maintainability decreases. There are plenty of software metrics presented in the literature for assessing the complexity and maintainability of software, with which the important one is the coupling metric.

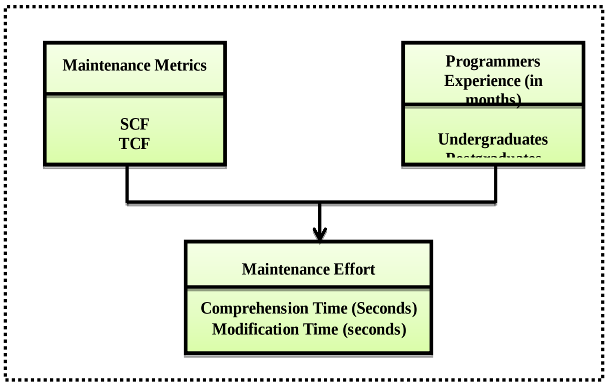
Coupling refers to the degree of dependency between software modules and also measures the strength of relationships of modules within one another [1]. Software with high cohesion and low coupling is often desirable as the low coupling indicates that the built software consists of well-structured programs. Software with high coupling indicates the incurrence of high complexity in the software which would in turn affects software maintainability. The types of software can be majorly categorized into procedural and object oriented programming. There are two types of object-oriented couplings possible as

- Subclass coupling: designates the dependency between the base and derived classes.
- Temporal coupling: when two or more actions are assimilated into one module just because they occur in a particular instance of time. For instance, processes whose activities are orderly sequenced or whose outputs are carry forward and waiting for a response to a prior request.

The primary objective of this paper is to statistically analyze the two OO coupling metrics namely SCF and TCF to prove their ability in predicting the complexity of the built software.

II. EXPERIMENTAL STUDY FRAMEWORK

The framework of this study suggests the complexity involved in the design, code, maintenance and programmers ability to understand the software for corrective and maintenance efforts. In this paper, the maintenance effort is taken as a dependent variable and the maintenance metrics, programmers experience are taken as the independent variables. The design complexity and the maintenance of software in terms of its casual relationships are depicted in Figure 1. This paper investigates the underlying relationships between the predictive maintenance time. The study also extended to predict the maintenance efforts. To prove this, an experimental study is conducted with the students of Bharathidasan University, Tiruchirappalli, India.


Maintenance Metrics
SCF
TCF
Programmers Experience (in months)
Undergraduates
Maintenance Effort
Comprehension Time (Seconds)
Modification Time (seconds)

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Software maintainability refers to ease of comprehending and modification of software such as time taken to understand and modifying the software code [2]. In this paper, the maintenance effort is interpreted as the time taken to comprehend and making modification in the existing programs.

A). Maintenance Metric

Coupling stands as an important factor to evaluate the quality of the software in terms of maintenance. Hence, two object oriented coupling metrics have been proposed as our earlier contributions called Subclass Coupling Factor (SCF) [3] and Temporal Coupling Factor (TCF) [4] which explicates the design complexity of the software. The description of the metrics is discussed in this section.

i). Subclass Coupling Factor

The proposed Subclass Coupling Factor metric (SCF) measures the direct and indirect subclasses of the individual class so as to calibrate the complexity of the whole module. SCF metric adopts the concepts of CF metric as its base and performs the union operation with the intersected sets as the results. Hence, the complexity of the whole module can be weighed to assess the quality of software modules.

$$C_L = C_{i=1}^{n}[C_{j=1}^{n} C_i(C_j)(is\_subclass(C_i, C_j)) \quad \dots (1)$$

CL is the class labels of sets Ci .. to Cn where n is the total number of classes in a module, Ci ,Cj represents the ith and jth class respectively and Ci represents the sets of all subclasses of ith class. The subclass elements are added onto the set Ci iff and only if Cj is the subclass for Ci .

$$\mathrm{CO}(\mathrm{C}_{\mathrm{CL}}) = \mathrm{C}_{i=1}^{n}[\mathrm{C}_i \cup \mathrm{alldirectandindirectsubclasses}(\mathrm{C}_i)]$$

CO is the complexity of each class.

$$SCF = \frac{\sum_{i=1}^{n} allelementsin\, C_i}{\sum_{i=1}^{n}(n-i)} \quad \dots (2)$$

The complexity values of SCF should range from 0 to 1, where 0 represents low subclass coupling and 1 represents high subclass coupling. A high subclass coupling is an alarm for the programmers as it implicitly depicts the high complexity in program design. Low subclass coupling is desirable as it reduces the code complexity.

ii). Temporal Coupling Factor

The calibration of temporal coupling is measured by analyzing the sequences of methods that processes an individual variable throughout all the classes in a module. The TCF value is obtained by performing the fraction of cumulative usage of all variables by the methods of the module with the total number of possibilities of temporal coupling within the module. An adjacency matrix between all coupling classes is created to trace the series of actions that are invoked when the variables are initialized. Let Mod be a module with several of interconnected classes with well-defined methods a coupling classes adjacency matrix CAMv×m can be derived using the notion CAM [i,j]

$$CAM[i,j] = \begin{cases} 1\ if\ variable\ i\ is\ used\ in\ method\ j \\ 0\ if\ varible\ i\ is\ not\ used\ in\ method\ j \end{cases} \quad \dots (3)$$

From this matrix, the series of actions that is performed with all variables are identified to be processed further. The accumulation of all series of actions with regard to all variables is then performed to compute the fraction with the possible temporal coupling in the module. The TCF is calculated as:

Let ‘a’ be the sum of all variables that is processed by the methods of the module which is initialized with 0.

$$TC = \frac{\sum_{i=0}^{v}\sum_{j=0}^{m} a=a+1(if\ AM\ [i,j]=1)}{(CCV \times CCM)-(VCC \times MPC)} \quad \dots (4)$$

where ‘v’ is the total number of variables and ‘n’ is the total number of methods in the module. ‘CCV’ is the number of coupling class variables, ’CCM’ is the number of coupling class methods, ‘VCC’ is the number of variables in the child class and ‘MPC’ is the number of methods in the parent class.

$$TCF = \frac{\sum_{i=0}^{tcg} TC}{Total\ number\ of\ Coupling\ Groups} \quad \dots (5)$$

TCG is the total number of coupling groups between the classes.

The maximum and minimum value of TCF can be 0 and 1 respectively. The module that results the TCF value closer to 1 signifies a high temporal coupling, the TCF value that is closer to 0 signifies the low temporal coupling and the TCF value that centres around 0.5 denotes the medium level temporal coupling. Modules with high and medium temporal coupling may be focused for further depreciation to a low level for reducing the complexity in the module.

B. Programmer Experience

The second dependent variable used in the experimentation is the samples experience with object oriented and other computer programming languages to investigate on how well the knowledge of computer programming helps the samples to handle lower and higher complexity programs.

C). Maintenance Effort

The maintenance effort is often predicted by maintenance task which is generally classified into comprehensive, corrective and adaptive [5]. This paper focuses only on the corrective and adaptive maintenance tasks at it assures the code oriented reliability in software maintenance. In our study corrective refers to the time taken by the samples to modify a segment of code in software modules and perfective refers to the time taken by the samples to understand a software module. Hence, the variable maintenance effort refers to the time taken by the samples to comprehend or modify the code in a module.

III. RESEARCH HYPOTHESIS

Based on the independent variables of framework shown in Figure 1, the primary objective of this research is to investigate the following research question

- Is there a relationship between the maintenance metrics and the programmer experience with corrective maintenance?
- Is there a relationship between the maintenance metrics and the programmer experience with the perfective maintenance?
- Is the metric values has the ability to predict the complexity of the software?

There are number of statistical tests to assess the relationship between the independent and dependent variables. In this study we have used the three statistical tests such as correlation, analysis of variance (ANOVA) and regression for yielding the solution of the research question.

A). Correlation
Correlation shows whether and how strongly two variables are related or dependent with each other in terms of the extent to which the variables have a linear relationship with each other [6]. Correlation expects an increase in one variable also let the increase in other or vice versa.

B). ANOVA
ANOVA contains a collection of statistical models used to analyse the differences among group means and their associated procedures. In ANOVA, observed variance in a particular variable is partitioned into components attributable to different sources of variation. In its simplest form, ANOVA provides a statistical test of whether or not the means of several groups are equal, and therefore generalizes the t-test to more than two groups [7]. ANOVAs are useful for comparing (testing) three or more means (groups or variables) for statistical significance.

C). Regression
Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analysing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables [8]. More specifically, regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

IV. EXPERIMENTATION AND RESULTS AND DISCUSSION

The experiment is steered over duration of one semester and conducted upon 58 students who had at least one to five years of programming language experience in which they had minimum three months of OO experience. The samples were chosen from both under and post-graduation students of computer science with 42 and 16 respectively. The details of the samples were collected through a questionnaire. There were two lower and higher complexity versions of stock market program, where each pair depicts a perfective and corrective maintenance tasks. The samples were split into two groups consists of 29 members each and were asked to yield the results of the programs by understanding the flow of classes in a module (Perfective) , and also instructed to modify the segment of code (Corrective). The starting and ending time to understand and modification of the samples was noted.
The objective of this study is to experimentally analyse the evaluation of coupling metrics by assessing their ability to find complexity of the program. Thus, Single Factor ANOVA test was conducted to determine the difference of mean maintenance time within and between the groups of high and low complexity versions of perfective and corrective tasks. Table 1 shows that the F Critical value of both perfective and corrective maintenance task is lesser than F-measure and the variance of low and high complexity mean maintenance time within the groups is minimum than the variance between the groups for both the maintenance tasks. Hence, the null hypothesis is rejected.

Table 1.Results of Analysis of Variance (ANOVA)

Maintenance Task	Mean Maintenance Time (MMT)				F Crit	F-Measure	P
	Low Complexity	High Complexity	Difference Within Groups (MS)	Difference Between Groups (MS)			
Perfective	178.37	313.55	9221.25	264937.93	4.012	28.73	<0.0001
Corrective	201.14	336.74	11942.94	395907.08	16.03	44.27	<0.0001

The graphical representations of the comparison of time complexity between the module versions, mean square difference and F-measure in Figures 2,3 and 4 respectively.

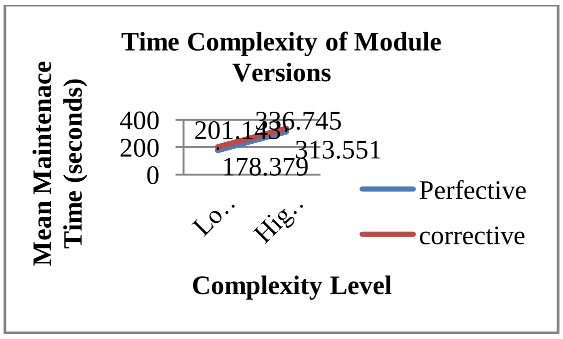


Figure 2.Time Complexity of Module Versions

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

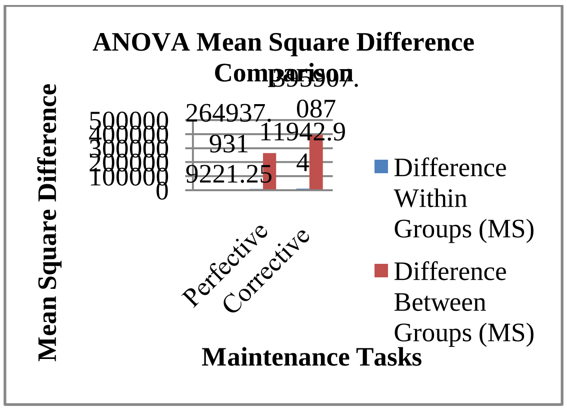


Figure 3. ANOVA Mean Square Difference Comparison

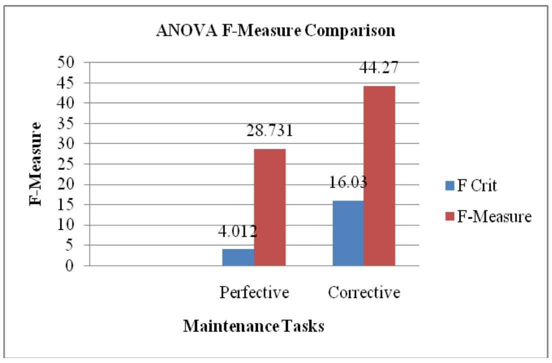


Figure 4. ANOVA F-Measure Comparison

This section analyses the results of regression analysis test conducted to examine the significance of OO coupling metrics with respect mean maintenance time. Table 2 depicts the coupling metrics value of low and high module versions of maintenance tasks and the average mean maintenance time taken by them.

Table 2: Regression Analysis ofOO metrics with MMT

Maintenance Task	Module Version	SCF	TCF	Mean Maintenance Time (MMT)
Perfective	Low	0.33	0.25	178.3793
	High	0.667	0.724	313.5517
Corrective	Low	0.4	0.312	201.3793
	High	0.694	0.856	336.5517

Table 3 shows the results of the multiple linear regression as per the values of Table 2. The correlation between the independent and dependent variable Multiple R (0.99999839) denotes a strong linear relationship them. The coefficient of determination R^2 (0.999997) determines close to 99% of the variation in the dependent variable is explained by independent variables. The Adjusted R Square (0.9999) adjusts the number of terms in a model, and increases only when meaningful values are added. Standard Error (0.246063) depicts the standard deviation error. The significance of F (00179457172937816) explains there is only 1% of chance that the regression output is merely a chance occurrence. Moreover, the p-value of the independent variables are not greater than 0.05, Hence, the null hypothesis is rejected for the multiple linear regression.

Table 3.Regression Statistics

Multiple R	0.99999839
R Square	0.999997
Adjusted R Square	0.99999
Standard Error	0.246063
Significance F	0.00179457172937816

From the results of Table it is clearly understood that the value of the coupling metrics clearly depicts the complexity of the software code. High metric values denote that the software is highly complex and maximizes the maintenance cost. Low metric values denote that the software is less complex and minimize the maintenance cost.

V. CONCLUSION

The primary objective of this paper is to statistically explore the validation of the two proposed OO coupling metrics: Subclass Coupling Factor (SCF) and Temporal Coupling Factor (TCF). For the statistical validation a planned laboratory experimented was conducted to achieve the research objectives of the paper. Three statistical tests such as ANOVA, correlation and multiple linear regression are employed to quantitatively analyse the experimental data. The results have proven that the coupling metrics are found to be useful in predicting the complexity of the software by rejecting the null hypothesis.

REFERENCES

[1] https://www.infoq.com/news/2009/04/coupling
[2] Rising. L.S, "Information Hiding Metrics for Modular Programming Languages," PhD dissertation, Arizona State Univ., 1992.
[3] S.A. Sahaya Arul Mary, ‚M.Kavitha, "A Quality Based Novel Subclass Coupling Factor Metric for Evaluating Software", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016.
[4] M. Kavitha, S.A. Sahaya Arul Mary, "An Analysis of Novel Temporal Coupling Factor Metric for Evaluating the Quality of Software", IJCTA, 9(27), 2016, pp. 211-218.
[5] Lientz. B.P and Swanson. E.B, ―Software Maintenance Management. Reading‖, Mass: Addison-Welsey, 1980.
[6] Xu, Jie, Danny Ho, and Luiz Fernando Capretz. "An empirical study on the procedure to derive software quality estimation models." arXiv preprint arXiv:1507.06925 (2015).
[7] van der Meulen, Meine JP, and Miguel A. Revilla. "Correlations between internal software metrics and software dependability in a large population of small C/C++ programs." In Software Reliability, 2007. ISSRE'07. The 18th IEEE International Symposium on, pp. 203-208. IEEE, 2007.
[8] A. Aloysius, L.Arockiam, "Maintenance Effort Prediction Model Using CognitiveComplexity Metrics", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 11, November 2013.