Software Defects and Object Oriented Metrics - An Empirical Analysis

Dr. M.P. Thapaliyal
Department of Computer Sc,
H.N.B. Garhwal University, Srinagar Garhwal

Garima Verma
Master of Computer Application Dept,
Dehradun Institute of Technology, Dehradun

ABSTRACT

Object Oriented design, today, is becoming more popular in software development environment. Object oriented measurements are being used to evaluate and predict the quality of software. Because of the need of Object Oriented software and high quality demand, traditional metrics cannot be applied. One of the reasons for this is that the traditional metrics measures are generally used in the structured programming paradigm where the design structure and data structure are measured independently. But an Object Oriented metric is able to treat function and data as combined integrated object.

In this paper we have evaluated two metrics Weighted Method per Class (WMC) and Coupling between Object Classes (CBO) of Chidamber and Kemerer metrics Suite. We have done an empirical study and tried to find out the nature of relationship of these metrics with defects. In other words, it has been investigated whether these metrics are significantly associated with defects or not. For this we have taken samples of 50 Java classes of different projects developed by the final year B. Tech students under the guidance of faculty members having 4-10 years experience in object oriented programming. We have deliberately taken different projects & tried to check if these metrics can really be reliable measurements for predicting defects when applied to inherently different projects.

Keywords:

Object Oriented Paradigm, Java.

1. INTRODUCTION

Object oriented design, today, is becoming more popular in software development environment and Object Oriented design metrics is an essential part of software environment. The main objective of analyzing these metrics is to improve the quality of the software [1], [2], [6]. Detection and removal of defects prior to the customer delivery is extremely important in the software development. Therefore, finding out the relation of defects with various measurement techniques in the process of software development becomes imperative. Does the size of a class in software development affect the defect proneness of software? Is there any effect of higher number of methods in a class on defects? These are some of the basic questions that need to be answered at the design time only. In this paper first we have discussed the Chidamber and Kemerer metrics suite first, then the methodology & analysis to answer above mentioned questions.

1.1 CK Metrics Suite

1.1.1 Weighted Method per Class (WMC) – WMC measures the complexity of a class [3]. WMC is a predictor of how much time and effort is required to develop and maintain the class. A large number of methods also mean a greater potential impact on derived classes, since the derived classes inherit the methods of the base class. If we analyze the WMC then we will find that high WMC leads to more faults which increases the density of bugs and decreases quality.

1.1.2 Depth of inheritance Tree (DIT) – DIT metric is the length of the maximum path from the node to the root of the tree. The deeper a class is in the hierarchy, the more methods and variables it is likely to inherit, making it more complex [2]. High depth of the tree indicates greater design complexity. Thus it can be hard to understand a system with many inheritance layers [4]. A high DIT has been found to increase faults and many methods might be reused.

1.1.3 Number of Children (NOC) – It is equal to the number of immediate child classes derived from a base class. NOC measures the breadth of a class hierarchy. A high NOC indicates several things like- High reuse of base class, base class may require more testing, improper abstraction of parent class etc.

1.1.4 Coupling Between Objects (CBO) - Two classes are coupled when methods declared in one class use methods or instance variables defined by the other classes. Multiple accesses to the same class are counted as one access. Only method calls and variable references are counted. An increase of CBO indicates the reusability of a class will decrease; also a high coupling has been found to indicate fault proneness. Thus, the CBO values for each class should be kept as low as possible.

1.1.5 Response for a Class (RFC) – The RFC is the count of the set of all methods that can be invoked in response to a message to an object of the class or by some method in the class. This includes all methods accessible within the class hierarchy. Pressman, states that since RFC increases, the effort required for testing also increases because the test sequence grows. If RFC increases, the overall design complexity of the class increases and becomes hard to understand.

1.1.6 Lack of Cohesion (LCOM) - LCOM measures the dissimilarity of methods in a class by instance variables or attributes. A highly cohesive module should stand alone; high cohesion indicates good class subdivision. LCOM measures the amount of cohesiveness present, how well a system has been designed and how complex a class is [3]. LCOM is account of the number of method pairs whose similarity is not zero. Lack of cohesion or low cohesion increases complexity, thereby increasing the likelihood of errors during the development process. If LCOM is high methods may be coupled to one another via attributes and then class design will be complex. So, designer should keep cohesion high, that is, keep LCOM low.

2. REVIEW OF RELATED LITERATURE

Amjan Shaik et al. (2010) have done statistical analysis for object oriented software metrics on CK metric suite by validating the data collected from the projects of some students. Metrics data provided quick feedback for software designers and managers. They found out that if appropriately used; it could lead to a significant reduction in cost of the overall implementation and improvement in quality of the final product.

Dr. B.R. Sastry et al. (2010) tried to implement software metrics with aid of GUI & also analyzed relationships of metrics to determine quality and quantity of software attributes measured with regard of object oriented software development life cycle.

C. Neelamegan et al. (2009) did survey on four object oriented quality metrics. In this study, the metrics for object oriented design focused on measurements that were applied to the class and design characteristics.

Dr Kadhim M. Breesam (2007) validated a set of metrics empirically that could be used to measure the quality of an object oriented design in terms of the using class inheritance.

Ruminate Subramanyam et al. (2003) conducted an empirical analysis on subset of C & K metrics suite in determining software defects.

Victor R. Basili et al. (1996) conducted empirical analysis on Object Oriented metrics. The aim of the paper was to access these metrics as predictors of fault prone classes.

3. RESEARCH OBJECTIVE & HYPOTHESIS FORMULATION

3.1 Research Objectives

The main objective of this paper is to understand the relationships between some selected metrics with software defects at a class level. We have taken one traditional metrics i.e. Size and two metrics of CK metrics suite i.e. WMC & CBO. We have tried to find out empirically how the size, WMC & CBO of the class affects the defect proneness in a software, especially when applied to inherently different software.

3.2 Hypothesis Formulation

Based on the literature we have formulated following three hypotheses for our study:

H1: Larger the class size, higher shall be the number of defects in software

H2: Larger the values of WMC in classes, higher shall be the number of defects

H3: Larger the values of CBO in classes, higher shall be the number of defects

4. METHODOLOGY

The primary purpose of the study was to find out the impact of three selected metrics, one traditional i.e. size & two object oriented metrics i.e. WMC & CBO, on defects found in software designs.

To achieve above purpose we have taken data from various projects developed by the students of B.Tech final year under the guidance of experienced faculties for diversified applications. The experience of the faculties varied from 4-10 years in object oriented programming. Also, all the students who were involved in developing these projects had been through the various programming languages since third semester. We have selected only those projects which have used Java as primary language for the development of the project. The projects considered for the study were of variety of nature such as memory management, customer management, Online Banking Transaction, Shopping website, etc. In total we have taken 50 classes for this study & calculated the above said metrics for them. Due to the unavailability of the source code of all the projects, we have collected data through discussion with project guide, student teams & documentations maintained by them. Also, we have checked the SRS prepared by the students & tested their project by applying various cases.

SIZE metric is based on the information given by the students while WMC & CBO is taken out from the documentation & bit part of the source code given by them. To analyze the values of defects, views of various faculty members and some students during testing.

The measurement of metrics is as follows:

1. Defects - Counted at class level
2. Size - All lines in the source code except blank lines & comment lines
3. WMC - Number of methods in the class

4. CBO - Number of classes to which a given class is coupled

Past literature indicates that relationship between defects; Size and Metrics are not linear [6]. However, we have tried to test & verify that in this study. Therefore, we applied multivariate linear regression but found out that the relationship was non-linear.
To find out a non-linear relationship, we generally used BOX-COX transformation for the above mentioned dependent & independent variables [13].
$y = (y^{\lambda} - 1)/\lambda$

We know that λ value close to zero indicates need for logarithmic transformation, the value close to 1 indicates linear form of dependent variable & value close to -1 indicates that a reciprocal transformation is appropriate. Since in our study the λ value was close to -1 but we also had many values 0 therefore we have 1 + defects as dependent variable.
Our empirical model for our analysis is as below-

$$1/(1+\text{Defects}) = a + \beta 1\ (\text{SIZE}) + \beta 2\ (\text{WMC}) + \beta 3\ (\text{CBO}) + e \quad \ldots\ldots\ldots\ldots \quad (1)$$

In the above equation, *a* is the constant, β1, β2 & β3 are the coefficients of SIZE, WMC & CBO respectively. These coefficients capture the effect of each one these variables. Here, e is an error term.
Since Defects have a reciprocal relationship with each independent variable therefore higher values of each coefficient would imply association with lesser number of defects.

5. ANALYSIS OF THE DATA

The above mentioned model has been empirically tested using Ordinary Least Square regression for Java classes selected from projects. To eliminate the problem of heteroskedasticity, weighted least square procedure was used.

The basic descriptive statistics of the variables involved is shown below in the table no. 1.

Table (1): Descriptive Statistics

As visible from the above table, the total number of classes taken in this study is 50 and we calculated size, defects, WMC & CBO for these classes. The mean values for defects, WMC & CBO for these classes may be higher primarily due the reason that the projects which have been taken for the study have been developed by final year students pursuing B.Tech courses.

The following table exhibits correlation amongst independent variables & the degree of correlation. In our study, the correlations amongst these variables have been found to moderate but significant.

Variables	**N**	**Minimum**	**Maximum**	**Mean**	**Std. Deviation**
Size	50	3	200	90.22	43.234
Defects	50	0	6	2.46	1.606
WMC	50	1	13	5.94	2.714
CBO	50	0	17	4.30	3.309

Table (2): Correlations amongst independent values

		Size	WMC	CBO
Size	Pearson Correlation	1		
	Sig. (2-tailed)			
WMC	Pearson Correlation	.539(**)	1	
	Sig. (2-tailed)	.000		
CBO	Pearson Correlation	.703(**)	.548(**)	1
	Sig. (2-tailed)	.000	.000	

** Correlation is significant at the 0.01 level (2-tailed).

To check the problem of multicollinearity amongst independent variables i.e. SIZE, DEFECTS & CBO, we checked data by using the conditions as specified in Belsley et. al.(1980) In all cases the maximum condition indices were well below the critical value of 20 as suggested by Belsley.

The following table exhibits weighted least square estimates for our empirical model (1). Because of the problem of heteroskedasticity in our model we used the procedure of weighted least square estimates by applying square root of class size as weight.

Table (3): Weighted Least Square Estimates (Square root of class size used as the Weight)

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		Std. Error
1	(Constant)	.710	.091*		7.842	.000
	Size	-.005	.001*	-.724	-4.174	.000
	WMC	-.011	.015	-.104	-.702	.486
	CBO	.012	.015	.140	.800	.428
	Adjusted R2	.347				

A Dependent Variable: DF, *Significant at 5%

6. RESULTS AND DISCUSSION

a. The impact of Class Size on Defects

From the table 3 it is visible that the standardize beta values for SIZE is -.724 & is found to be significant at 5 % significance level. In other words, the impact of size on defects is negative and significant. But since in our model we had taken dependent variables as 1/(1+defects), therefore we interpret the results as increase in the class size leads to increase in defects & this supports our hypotheses number one i.e. H1. Our findings are for this variable are consistent with earlier findings of El Emam et al. (2001) who found out that size does confound the effects of metric on defects.

b. The impact of WMC on Defects

The weighted least square model results show that WMC (with value of -.104) has again a positive relationship with number of defects indicating that increase in the number of methods is going to increase the number of defects. However, the relationship between defects & WMC has been found insignificant at 5 % level. In other words, our hypotheses number two i.e. H2 is rejected for the model.

c. The impact of CBO on Defects

The standardized beta value for the metric Coupling between Objects in our empirical models is .140 indicating an inverse relationship between CBO & defects. Also, this value has been found insignificant at 5 % level. Therefore, we reject our third hypotheses i.e. H3.

7. CONCLUSION

In this paper we have made an effort to find out the nature of relationships between the Defects in software designs developed in Java and some selected metrics such as class SIZE, WMC and CBO. Also, how do all these metrics impact the defects proneness of software have been analyzed? [6][12] Various studies in the past have shown a positive and significant relationship between the number of defects and SIZE, WMC and CBO. Most of these studies had used either a single project or controlled similar projects to collect data for these variables. But in this study, we tried to test these relationships to different projects and have found some inconsistencies in results.

Though, class size has been found to have a positive & significant impact on defects, which was consistent with earlier findings, but other two variables i.e. WMC & CBO have been found to have insignificant impact on defects which is not consistent with earlier findings. One the possible explanation for this could be the difference in the nature of projects which have been used in our study to collect metric data related to these variables.

One the important conclusions that may be drawn from this study is that even most recognized metrics that are used to predict the defect proneness of software may not be sufficient enough when applied to diverse kind of projects. We may need some more metrics to enhance the predictability in defect proneness of software designs.

ACKNOWLEDGEMENT

We want to thank Mr. Hemraj Verma who helped in training us on statistical tool SPSS for analysis, Mr Ashok Kumar for his valuable suggestions to improve the contents of the paper and students of B.Tech final year cooperated us in our data collection and findings.

REFERENCES

1. Dr B.R. Sastry, M.V. Vijaya Saradhi, “Impact of software metrics on Object Oriented Software Development life cycle”, International Journal of Engineering Science and Technology, Vol 2 (2), pg 67-76, 2010.
2. Amjan Shaik, Dr C.R.K. Reddy, Dr A Damodaran, “Statistical Analysis for Object Oriented Design Software security metrics”, International Journal of Engineering Science and Technology, Vol 2 (5), pg 1136-1142, 2010.
3. C. Neelamegan, Dr. M. Punithavalli, “A Survey- Object Oriented quality metrics”, Global journal of Computer Sc. And Technology, Vol 9, no 4, 2009.
4. Dr Kadhim M. Breesam, “Metrics for Object Oriented design focusing on class Inheritance metrics”, 2nd International conference on dependability of computer system IEEE, 2007.
5. K.K. Agarwal, Yogesh Singh, Arvinder Kaur and Ruchika Malhotra, “Empirical Study of Object-Oriented Metrics”, Journal of Object Technology, Vol 6, pg 121-138, 2006.
6. Ramanath Subramanyam and M.S. Krishnan, “ Empirical Analysis of CK metricsfor Object Oriented Design Complexity: Implications of Software defects” IEEE transactions on Software Engineering, vol 29, no 4, 2003.
7. K. El Emam, W. Melo, and J.C. Machado, “The prediction of faulty classes using Object-Oriented design metrics”, J. Systems and software, vol 56, pp 63-75, 2001.
8. L.C. Briand, J. Wuest, J.W. Daly and D.V. Porter, “Exploring the relationship between design measures and software quality in Object Oriented Systems”, J. Systems and Software, Vol 51, no 3, pp 245-273, 2000.
9. Glasberg, D. Emam, K.E. Melo, W., & Madavji N., “Validating Object Oriented Design metrics on a commercial java application”, National Research Council 44146, 2000.
10. M.H. Tang, M.H. Kao, and M.H. Chen, “An Empirical study on Object Oriented Metrics”, Proc. 6th International software metrics symposium, pp 242-249, 1999.
11. Abreu, F.B.E. and Melo, W., “Evaluation the impact of OO design on Software Quality”, presented at 3rd International software metrics symposium, Berlin, 1996.
12. Victor R. Basili, walcelio L, Melo, “A validation of Object Oriented Design Metrics ass Quality Indicators”, IEEE Tracsactions on software engineering, Vol 22, No. 10, 1996
13. G. Box and D. Cox, “An analysis of transformations, “, J. Royal Statistical Soc., Series B, pp. 211-264, 1964.

AUTHORS

Dr. Thapliyal is Phd and working as Reader in Department of Computer Science, Garhwal University, Srinagar and have published several papers in National and International Journals. He has been supervising PhD research scholars in the field of computer science for several years.

Garima Verma has done M.Phil in Computer Science and Post Graduation in Computer Application. Presently, she is working as Asstt. Professor in the Department of Computer Application at DIT, Dehradun. She has got more than 5 years of industry experience and 6 years of teaching experience and published 5 research articles till now in different refereed journals. Also, she has attended and presented papers in several conferences and workshops.