International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 4 Issue 11, November-2015

Evaluation of Effect of Faults on OO Software using Univariate Logistic Regr. p-value & R^2

Dr. Brij Mohan Goel

Assistant Professor, Vaish College of Engineering, Rohtak

ABSTRACT

Software industry is developing software using Object-oriented design. To predict the quality of these systems, many OO design metrics have been developed. There is no fool proof evidence on the significance of these metrics. The object oriented metrics will be adopted to identify a limited set of measureable attributes that have a significant impact on prediction of faults and quality attributes. In this paper three projects from the NASA data set to access the applicability of object-oriented CK metrics are used to evaluate the effect of faults on object-oriented software.

Keywords: OO Metrics, CK Metrics, Faults, Regression.

1. INTRODUCTION

It is necessary to predict faults in software development life cycle. This activity will reduce the software failure and identify the modules that are fault prone. This activity must be carried out in the early phases of the development life cycle. Fault prediction not only gives an insight to the need for increased quality of monitoring during software development but also provides necessary tips to undertake suitable verification and validation approaches that eventually lead to improvement of efficiency and effectiveness of fault prediction. Effectiveness of a fault prediction is studied by applying a part of previously known data related to faults and predicting its performance against other part of the fault data. Several researchers have worked on building prediction models for software fault prediction to improve the quality of software systems.

2. LITERATURE SURVEY

Jagmohan Mago et al., [1] proposed a system based on fuzzy logic to assess the quality of OO design, uses the CK metric suite and Mamdani Inference Engine. They presented a decision making system that is based on fuzzy inference mechanism and is proposed by Mamdani. In the proposed, system, they consider all the crisp values of six input metrics and each input metric was defined by three membership function i.e. LOW, MEDIUM and HIGH. Each of the inputs is mapped to a membership value in the interval [0, 1]. The value zero is used to indicate the complete non membership and value one indicate the complete membership and value in between were used to represent intermediate degrees of membership. The first step was the fuzzification of all inputs by transforming the crisp values into fuzzy values. All the input metrics were considered and combined with AND operator. The MIN/MAX membership operator used to determine the degree of membership in Mamdani inference Engine. The technique of defuzzification was Centroid which transformed the fuzzy values to linguistic variable.The results obtained from the system are compared with number of industrial software tools such as Analyst4j and ViZZAnalyzer. They showed that the results produced by the system were better than the results produced by the software tools. This was validated and verified by the human experts such as professors and the developers in the field.

Aman Kumar Sharma et al., [2] selectedCK metric suite to measure the quality of the OSS software. They use six parameters- Number of Children, Weighted Methods per Class, Depth of Inheritance Tree, Coupling between Objects, Lack of Cohesion in methods, Response for Class to measure the quality. They use empirical study to evaluate the quality of OSS i.e. JasperReport and LlamaChat. The results obtained were in accordance with the theoretical results i.e. low WMC, high DIT, low RFC and high cohesion. They showed that the lower the values of WMC, the less will be the complexity. The values obtained for RFC lied between 0 and 50 which were in accordance with theoretical results depicted for good quality software. The DIT values were more than NOC values. The reason was that the depth in the inheritance hierarchy is better than the breadth in the inheritance hierarchy. LCOM values for both the OSS's were decreasing with the updated versions. The result showed that the latest versions were more cohesive and high cohesion is an important factor contributing towards high quality software.

Sharma Aman Kumar, KaliaArvind, Singh Hardeep [3] studied the metrics identification for measuring object oriented software quality. This study presents a review of quality metrics suites namely, MOOD, CK and Lorenz & Kidd, and then selects some metrics and discards other metrics based on the definition and capability of the metrics. This study used three object oriented software metrics suite comprising of CK Suite, MOOD Suite and Lorenz and Kidd Suite. All the metrics suites evaluated in the study were from the object-oriented domain. The work of CK suite was seminal in defining metrics, binding scope of metrics, class level based and validating quality. The MOOD suite is well defined, project level based, mathematically computable. The metrics collected from a given design can be judged by the thresholds provided by the MOOD suite. But, the Lorenz and Kidd suite is neither validated in the existing studies nor the metrics of Lorenz and Kidd suite are capable to measure software quality. The Lorenz and Kidd metrics are statistical measures for software in terms of counting: the number of methods under various categories, the number of variables, etc. The Lorenz and Kidd metrics seems to be ineffective for measuring software quality. From among the suites analyzed the study has recommended metrics which are useful in evaluation of software quality. The metrics namely, WMC, RFC, DIT, NOC and CBO are suitable for evaluation of software quality from the CK Suite and whereas from the MOOD suite the appropriate metrics are MHF, AHF, MIF, AIF and PF. Based on the comparison and analysis the study concluded that the mentioned list of metrics is the most complete, comprehensive and supportive.

3. METRICS USED IN THE STUDY

The objective here is to establish the relationship between fault proneness and OO metrics at the class level. The CK metrics that corresponds to inheritance, size, coupling, and cohesion are used in this study and will act as independent variables. These OO metrics are used as independent variables. These metrics can be used in a model, to predict the fault proneness, in the initial stages of development of the software. The CK metrics that are used in this paper are shown below.

- WMC: Weighted Methods per Class
- NOC: Number of Children
- DIT: Depth of Inheritance
- RFC: Response for a Class
- CBO: Coupling between Objects
- LCOM: Lack of Cohesion

The fault (Bug) is used as a dependent variable. Every CK metric is used as an independent variable. Thus a function needs to be established between fault of a class and CK metrics (WMC, NOC, DIT, RFC, CBO, LCOM) suite. The fault (bug) is a function of WMC, NOC, DIT, RFC, CBO and LCOM [4]. It can be represented by using the equation [4].

$$\text{Faults} = f(\text{WMC, NOC, DIT, CBO, RFC, LCOM})$$

4. OBJECTIVE

The objective has been set for this paper is “To investigate the impact of faults on object oriented software to improve the quality”.

5. METHODOLOGY AND FAULT PREDICTION METHOD

Data Analysis Methodology

The purpose of the study is to analyses the six Object Oriented CK metrics. This study evaluate whether the CK metrics are valuable for predicting fault-prone classes or not, when the impact of faults is considered. What makes the class fault prone depends on the context in which that class is used. In case of low impact faults, a class is known fault prone if there exist at least one low impact fault. In the case of medium impact faults, a class is known fault prone if there exist at least one medium impact fault. In the case of high impact faults, a class is known fault prone if there exist at least one high impact fault. All these cases are considered when we investigate the fault-proneness prediction capabilities of the metrics.

When there is a fault of any impact in the class then the class is said to be fault-prone, this means there exists at least one fault of any impact in the class. The author of this thesis distinguished the three types of prediction models: low impact, medium impact and high impact fault models. Further, the classes are divided into two categories namely fault-prone and not fault-prone. The type i.e. fault-prone or not fault-prone is taken as the dependent variable [5]. The various metrics are taken as the independent variables. The model helps us to assess the metrics to predict fault-proneness in the classes.

Here, we will discuss the data set that is being used to predict the faults in the classes by using object oriented metrics. The Fault (Bug) is chosen as the dependent variables and the metrics are chosen as the independent variables for the fault prediction.

Collection of Data

Many metric and metric suites have been defined and used for fault prediction. These metrics and metric suits can also be used for reusability, estimation of efforts and maintenance. This study uses the most popular CK metric suite [6] for predicting the faults. The NASA [7] datasets, available in public domain, are used to assess the impact of fault-prediction.

Logistic Regression Model

Logistic regression is a standard statistical method of modelling. This method takes one value as the dependent variable from the two different values. This method is appropriate for constructing classification model for software quality. This is possible as the classes are of two types namely fault-prone and not fault-prone [8].

The univariate logistic regression model is a special case of the multivariate logistic regression model. In univariate model, we have only one independent variable [8, 9, 10]. All the observations are statistically independent, when a logistic regression model is build.

We have used univariate logistic regression in our study.Univariate regression analysis examines the effect of every metric independently. This means it identifies that metrics that are significantly associated to fault-proneness of the classes.

6. STATISTICS FOR UNIVARIATE LOGISTIC REGRESSION ANALYSES

The following statistics are collected for each of the given metric for univariate logistic regression analysis:

- **p-value –** It is known as calculated probability**.** The p-value is associated with the statistical hypothesis and tells us whether the resultant coefficient is significant or not.We use the α=0.05 significance level.
- **R^2 -** It is the proportion of the variance in the dependent variable that is explained by the variance of the independent variables. The accuracy of the model is better if the effect of the model's explanatory variables is high.

We use the SPSS software package for analysis and reporting the above said values. The outcomes of univariate logistic regression analyses for low impact, medium impact and high impact faults have been described.

7. ANALYSIS

Table 1 shows the outcome of the univariate analysis to evaluate the fault-proneness prediction in InterCafe1 data set. Table 2 shows the outcome of the univariate analysis to evaluate the fault-proneness prediction in TermoProjekt1 data set. Table 3 shows the outcome of the univariate analysis to evaluate the fault-proneness prediction in Zuzel1 data set.

Table 1: Result of Univariate Logistic Regression for InterCafe1

Metric	p-value (significance)	R^2
WMC	0.0001	0.457
RFC	0.0000	0.441
CBO	0.0000	0.498
LCOM	0.0699	0.125
DIT	0.5672	0.013
NOC	0.7184	0.005

From Table 1 we have noted that three of the six metrics have very significant p-value (< 0.002). The p-value of metric DIT (0.5672) and metric NOC (0.7184) are not significant. The p-value of metric LCOM (p-value=0.0699) is also not significant, but has less value than DIT and NOC. So DIT and NOC are more insignificant than LCOM.

WMC, RFC and CBO have the largest R^2 value, this shows that WMC, RFC and CBO metrics are the best predictors. LCOM has the slightly largest R^2 value, this shows that LCOM is not the best predictor. DIT and NOC have the smallest R^2 value, this shows that DIT and NOC are not the best predictors.

Table 2: Result of Univariate Logistic Regression for TermoProjekt1

Metric	p-value (significance)	R^2
WMC	0.0006	0.256
RFC	0.0002	0.298
CBO	0.0085	0.161
LCOM	0.0084	0.161
DIT	0.0147	0.140
NOC	0.5651	0.008

Table 2shows the outcome of the univariate analysis to evaluate the fault-proneness prediction in TermoProjekt1 data set.It is noted that WMC and RFC metrics have very significant p-value (< 0.001). Further, p-value of DIT and NOC metric (i.e. 0.0147 and 0.561 respectively) are not very significant.BothCBO and LCOM metrics have smaller value than DIT and NOC metrics. This means DIT and NOC metrics are more insignificant than CBO and LCOM metrics.

WMC and RFC metrics have the largest R^2 value. The CBO and LCOM metrics have the second highest R^2 value. The DIT and NOC metrics have significantly smaller R^2 value than the other metrics, this shows that these are less valuable.

Table 3: Result of Univariate Logistic Regression for Zuzel1

Metric	p-value (significance)	R^2
WMC	0.0023	0.295
RFC	0.0000	0.540
CBO	0.1145	0.090
LCOM	0.0085	0.230
DIT	0.0006	0.356
NOC	**-***	**-***

*The value of NOC is zero in all columns in Zuzel1.

Table 3 shows the outcome of the univariate analysis to evaluate the fault-proneness prediction in Zuzel1 data set. It is noted that three of the six metrics have very significant p-value (< 0.003). The CBO, LCOM and NOC metrics are not significant.

Statistics cannot be computed for NOC, because the values of independent variable NOC are constant in NASA data set with dependent variable BUG.

The value of R^2 is largest for the metricRFC. WMC, CBO, LCOM and DIT metrics have the second largest R^2 value. The R^2 value of NOC metric is undefined due to its Zero value. This showsthat the NOC metric is less useful.

Table 4: Comparison of p-values Project-wise

Metric/Project	InterCafe1	TermoProjekt1	Zuzel1
WMC	0.0001	0.0006	0.0023
RFC	0.0000	0.0002	0.0000
CBO	0.0000	0.0085	0.1145
LCOM	0.0699	0.0084	0.0085
DIT	0.5672	0.0147	0.0006
NOC	0.7184	0.5651	0.0006

* Consider NOC=0.0006 (as equal to DIT) in Zuzel1, The value of NOC is statistically not calculated, because of zero value in Zuzel1.

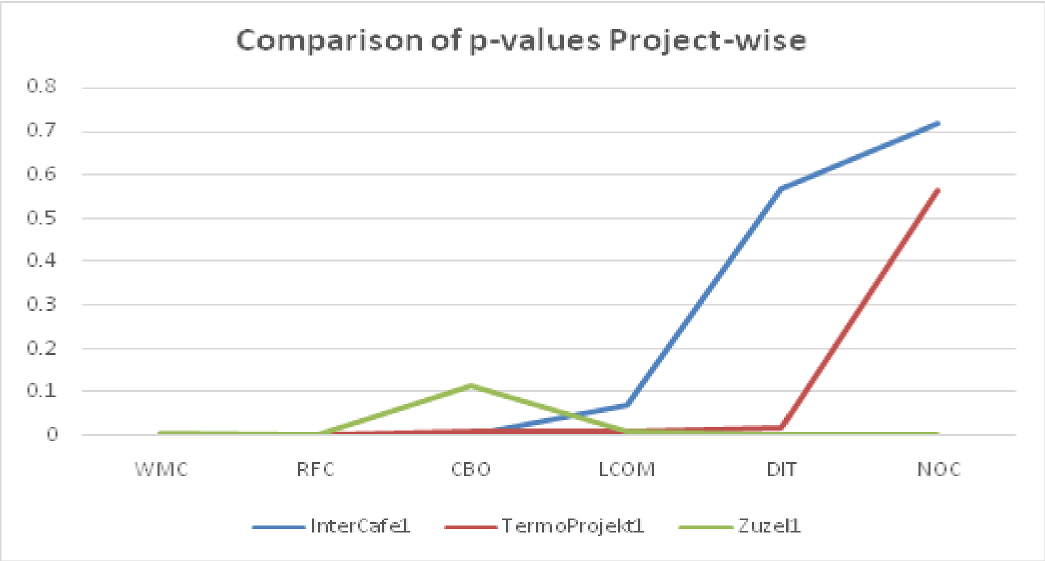

Figure 1: Line chart to show the Comparison of p-values project-wise

From Table 4 and Figure 1,the p-value of WMC, RFC and CBO metrics is less than 0.002 means having significant values for the project InterCafe1. The p-value of WMC, RFC metrics is less than 0.003 means having significant values for the project TermoProjekt1. The p-value of WMC, RFC and DIT metrics is less than 0.003 means having significant values for the project Zuzel1. To see the comparison p-values, project InterCafe1 values are more significant. So project InterCafe1 has the low impact of faults in the classes.

Table 5: Comparison of R^2 Values Project-wise

Metric/Project	InterCafe1	TermoProjekt1	Zuzel1
WMC	0.457	0.256	0.295
RFC	0.441	0.298	0.540
CBO	0.498	0.161	0.090
LCOM	0.125	0.161	0.230
DIT	0.013	0.140	0.356
NOC	0.005	0.008	0.356

* Consider NOC=0.356 (as equal to DIT) in Zuzel1, The value of NOC is statistically not calculated, because of zero value in Zuzel1.

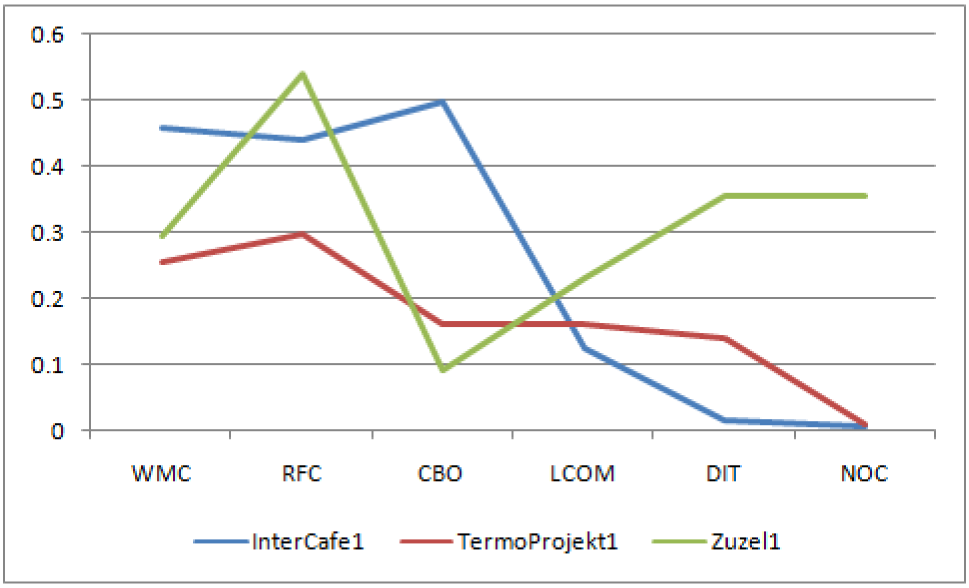

Figure 2: Line chart to show the Comparison of R^2 values project-wise

From Table 5 and Figure 2, the value of R^2 of WMC, RFC and CBO metrics for the InterCafe1 project is high means best predictor at these metrics. The value of R^2 of RFC metric for the TermoProjekt1 project is good means best predictor at this metric. The value of R^2 of WMC, RFC and DIT metrics for the Zuzel1 project is high means best predictor at these metrics.

The complete effect of the values comparing the values of p-value and values of R^2 for univariate analysis to evaluate the fault-proneness prediction is as follows. The effect is in terms of low/medium/high impact of faults. The project InterCafe1 has low impact of faults in the classes. The project TermoProjekt1 has medium impact of faults in the classes. The project Zuzel1 has high impact of faults in the classes.

CONCLUSION

Public domain data set InterCafe1, TermoProjekt1 and Zuzel1 available in the NASA repository are used in object-oriented software. The authors performed the statistical univariate logistic regression p-value and R^2 analyses to evaluate the fault-proneness prediction. This analysis is performed considering the OO metrics with regard to low, medium and high, impact faults. The authors analyzed six OO design metrics from CK metrics suite. The result with respect to fault-proneness prediction in terms of Low/Medium/High impact of faults is that the InterCafe1 project has low impact of faults in the clssses, the impact of faults in the classes of TermoProjekt1 project has medium impact and the impact of faults in the classes of Zuzel1 project has high impact.

REFERENCES

[1] JagmohanMago et al., “Fuzzy Model to Analyze and Interpret Object Oriented Software Design”, International Journal of Electrical, Electronics and Computer Engineering Vol. 1, No. 1, pp. 41-46, 2012.
[2] Aman Kumar Sharma, ArvindKalia and Hardeep Singh, “Empirical analysis of object oriented quality suites”, International Journal of Engineering and Advanced Technology, vol. 1 no. 4, 2012.
[3] Sharma Aman Kumar, KaliaArvind, Singh Hardeep, “Metrics Identification for Measuring Object Oriented Software Quality”, International Journal of Soft Computing and Engineering (IJSCE), Volume-2, Issue-5, November 2012.
[4] Love Kumar, Software Fault Prediction using Object-Oriented Metrics, e-thesis, Department of Computer Science and Engineering, National Institute of Technology Rourkela, India, 2014.
[5] Goel, Brij Mohan, and Pradeep Kumar Bhatia. "Investigating of high and low impact faults in object-oriented projects", ACM SIGSOFT Software Engineering Notes, 2013.
[6] Schröter A, Zimmermann T, Premraj R and Zeller A, “If your bug database could talk”, In Proceedings of the 5th International Symposium on Empirical Software Engineering, vol. 2, pp. 18-20, 2006.
[7] Lech Madeyski and Marian Jureczko, “Which process metrics can significantly improve defect prediction models? An empirical study”, Software Quality Journal, vol. 23(4), pp. 393-422, 2015.
[8] Rumbaugh, J.,Blaha, M., Premerlani,W., Eddy F. And Lorenses, W: Object-oriented modeling and design, Prentice Hall, 1991.
[9] Booch G., Object-Oriented Analysis and Design with Applications, Second ed. Redwood City, Calif.: Benjamin/Cummings, 1994.
[10] Banker, Rajiv D., Kauffman, Robert J., Kumar, Rachina.: "An Empirical Test of Object-based Output Measurement Metrics in a CASE Environment." Journal of Management Information Systems 8,3 (Winter 1991): 127-150.