International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 6 Issue 1, January-2017, Impact Factor: 1.544

Analysis of Impact of Faults on OO Software using Univariate Logistic Regression Coefficient& Standard Error

Dr. Brij Mohan Goel

Assistant Professor, Vaish College of Engineering, Rohtak, Haryana, India.

ABSTRACT

This paper assesses the ability of OO metrics to identify fault-proneness in OO software systems. The faults (bugs) in the class per metric value in various OO designs are compared. We have used three projects from the NASA data set to access the applicability of object-oriented CK metrics. The CK metrics are used to predict the bugs in the class. This enables us to calculate the fault-proneness using OO metrics. Further, we can make the necessary corrections in the object-oriented design using the information obtained from the metrics. This will increase the quality and reliability of the OO software systems.

Keywords: CK Metric, Regression, Object Oriented, Fault.

1. INTRODUCTION

Currently, the Object-Oriented (OO) paradigm is used extensively in the development of software systems. The OO metrics can be used to access the quality of these OO software systems. Many metrics and metric suits have been proposed/developed by the researchers to access the software quality. The properties of object-oriented designs can be measured using object-oriented metrics. With the help of metrics, more accurate estimations of project milestones can be attained. We can develop software projects with nominal faults. Maintenance of the project, budgets etc is taken care by project based metrics.

The design based metrics keep track of design performance and its complexity. They also keep track of size and robustness of object-oriented designs. The OO design is a new technology as compared to structural development. The metrics that are used to evaluate structural development may not affect the design that use Object oriented language. For example, the "Lines of Code" metric is not of much use in object-oriented design but is extensively used in structural development. A study shows that we can save an estimated 42% on account of corrective maintenance by using object-oriented metrics [1].

2. OBJECT ORIENTED METRICS

Software metric is the measurement of an individual characteristic of a program's efficiency or performance and also used to measures the attributes of software products and processes. Presently, most of the software is developed using Object-Oriented (OO) languages. The Object-Oriented paradigm varies drastically as compared to procedural development as the metrics that can be applied on procedural systems cannot be applied on OO software.

The CK are the first one to propose the suites of metrics for OO design measures [2]. The CK metrics claims that the given measures help users in understanding design complexity. It also helps in detecting design flaws and in predicting certain project outcomes. Further, they claim that it also helps in predicting external software qualities such as software defects, testing, and maintenance effort. The use of CK metrics and other corresponding measures are increasingly growing in the software industry. The CK metrics suite is one of the OO design complexity measurement systems that support the measurement of the external quality parameter of a software package. The literature contains many metrics that depends on the internal structural analysis of OO components i.e. inheritance, coupling, cohesion etc.

3. OBJECTIVE

These days most of the software industry is developing software using Object-oriented design. To predict the quality of these systems, many OO design metrics have been developed. The objective has been set for this paper is "To study fault prediction using various object-oriented metrics".

4. LITERATURE SURVEY

Predicting software fault proneness has been a critical issue in software engineering. A lot of studies have been conducted on this critical issue worldwide. For the purpose of this thesis a literature survey was carried out. Some of these studies have been reviewed here:

N. Rajkumar et al. [3] established the relationship, at class level, among object oriented metrics and fault proneness. The fault was taken as a function of DIT, CBO, WMC, NOC, RFC, and LCOM. All these metrics were of the CK metric suite. The machine learning technique of neural networks was used for problem definitions. It involves prediction and classification. The learning of the rules and membership functions from data used the neuro-fuzzy based system approach. Further, the authors showed that for prediction of fault proneness, conceptual relations between classes could be an excellent metric. The authors showed some of the coupling and inheritance measures through multivariate analysis. They showed that it is possible to derive accurate models to predict the classes that contain most of the faults. They showed that when predicting fault prone classes, the best model shows 80% of correct classifications and finds over 90% of faulty classes.

Madhu Rohila et al. [4] has used the technique ofbugs per class per metric are measured with the help of object-oriented CK metrics. Thereafter the object-oriented design can easily be correct out. They evaluated

Total number of bugs with respect to WMC = Σ (WMC of the Class *bugs per class value)
Total number of bugs with respect to DIT = Σ (DIT of the Class *bugs per class value)
Total number of bugs with respect to CBO = Σ (CBO of the Class *bugs per class value)
Total number of bugs with respect to RFC = Σ (RFC of the Class *bugs per class value)

They compared the bugs per class per metric value in different object-oriented designs. They employed hypothetical examples to access the applicability of object-oriented CK metrics to predict the bugs in class. They showed that DIT metric is best metric in predicting the fault-proneness of classes. They showed that once they have calculated the fault-proneness with the help of object-oriented metric, then the object-oriented design can easily be corrected that will affect the quality and reliability of the object-oriented design.

Johny Antoony P., [5] has determined for a software project, the threshold value of software reliability. They further validated the same. The threshold for reliability is determined using the relationship between the reliability at the class level with CK Metrics. First of all they estimated the threshold for Reliability of the software by selecting the CK metric suite. Next, the OO Metrics in the literature were used as the threshold values of CK metrics. Further, as per the principle and experience they showed that if the too low metric values may signify poor utilization of the benefits of OO technology and too high values may signify more complexity. A new threshold is proposed for the CK Metrics. Then, the CK metrics values were assigned with corresponding weighted values. After that, the relationship was established between Reliability and CK metrics to calculate the threshold for the reliability i.e.

Reliability α 1/WMC,
Reliability α 1/RFC,
Reliability α 1/DIT
Reliability α 1/LCOM
Reliability α 1/CBO

To validate the threshold values, a special tool named Java Class Analyzer was used to collect the CK metric values applications/projects on class level. The calculated reliability is checked to lie within the thresholds. Their results showed that the Reliability value that lies within the thresholds will have less number of defects and has more reliability. They have analyzed 16 projects. From them, the 8 are working properly and 8 are more prone to errors. They proved that the designers can improve the reliability and quality of the whole system by maintaining the threshold values of the metrics WMC, CBO, DIT, RFC, LCOM, and NOC.

5. DATA SET USED FOR FAULT PREDICTION

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.

6. METHODOLOGY AND FAULT PREDICTION METHOD

6.1 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.

7. STATISTICS FOR UNIVARIATE LOGISTIC REGRESSION ANALYSES

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

- **Regression Coefficient –** It is the constant that represents the rate of change of one variable (say y, dependent variable) as function of changes in the other variable (say x, independent variable).
- **Standard Error –** It is ameasure of the statistical accuracy of an estimate. The smaller values of the standard error indicate that the observations are closer to the fitted line and are better.

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.

8. 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.

If the absolute value of the regression coefficient is large then the impact is stronger. The strong impact (+ve or -ve, coefficient sign) of the independent variable depends on the likelihood of the fault being found in the class.

Table 1: Result of Univariate Logistic Regression for InterCafe1

Metric	Regression Coefficient	Standard Error
WMC	0.0226	0.535
RFC	0.0097	0.543
CBO	0.0897	0.514
LCOM	0.0015	0.679
DIT	0.0361	0.721
NOC	0.0673	0.724

From Table 1, we have noted that all metrics except DIT and NOC have low value of standard error, this means classes with low WMC, RFC, CBO and LCOM are less fault-prone. All metrics have positive regression coefficients, which means classes are less fault-prone.

Table 2: Result of Univariate Logistic Regression for TermoProjekt1

Metric	Regression Coefficient	Standard Error
WMC	0.0689	0.612
RFC	0.0131	0.594
CBO	0.0485	0.650
LCOM	0.0052	0.649
DIT	0.1409	0.658
NOC	0.1037	0.706

Table 2 shows that all metrics except DIT and NOC have low value of standard error, this means classes with low WMC, RFC, CBO and LCOM are less fault-prone. All metrics have positive regression coefficients, which means classes are less fault-prone.

Table 3: Result of Univariate Logistic Regression for Zuzel1

Metric	Regression Coefficient	Standard Error
WMC	0.0510	0.875
RFC	0.0275	0.707
CBO	0.0363	0.994
LCOM	0.0034	0.914
DIT	0.2501	0.836
NOC	-*	-*

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

Table 3 shows that all metrics except CBO and LCOM have low value of standard error. The value of NOC is statistically not calculated, because of zero value in Zuzel1. This means classes with low WMC, RFC and DIT are less fault-prone. All metrics have positive regression coefficients except NOC, which means classes are less fault-prone.

Table 4: Comparison of Regression Coefficient Values Project-wise

Metric/Project	InterCafe1	TermoProjekt 1	Zuzel1
WMC	0.0226	0.0689	0.0510
RFC	0.0097	0.0131	0.0275
CBO	0.0897	0.0485	0.0363
LCOM	0.0015	0.0052	0.0034
DIT	0.0361	0.1409	0.2501
NOC	0.0673	0.1037	0.2501

* Consider NOC=0.2501(as equal to DIT) in Zuzel1, the value of NOC is statistically not calculated, because of zero value in Zuzel1.

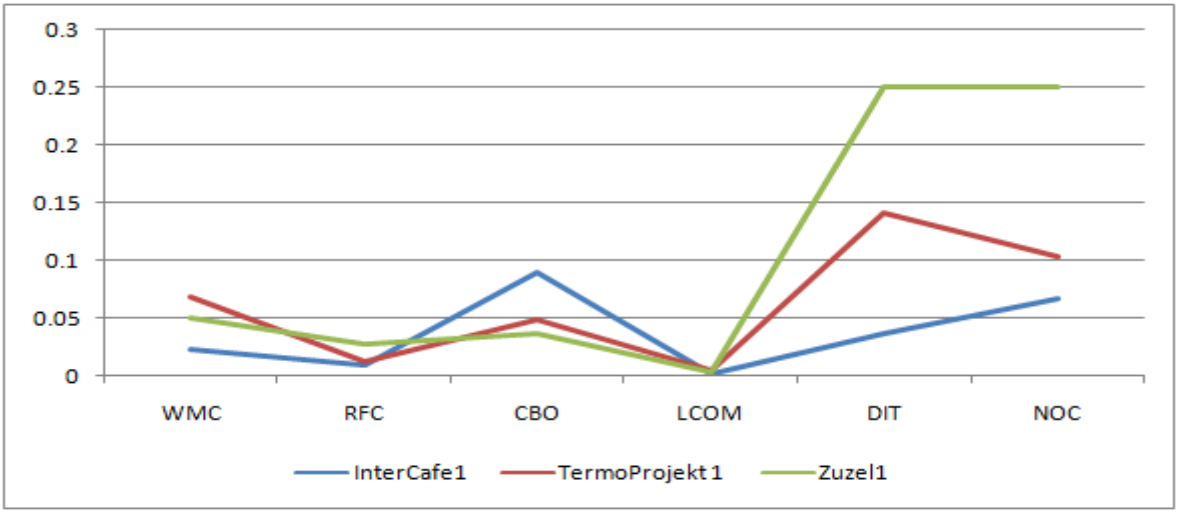

Figure 1: Line chart to show the Comparison of Regression Coefficient values project-wise

From Table 4 and Figure 1, the regression coefficient value for the project InterCafe1 is low. The regression coefficient value for the project Zuzel1 is high as average of all metrics except metric CBO. When the absolute value of the coefficient is large, this indicates that the impact is stronger. The strong impact (+ve or –ve, coefficient sign) of the independent variable depends on the possibility of the fault being found in the class. This means, the project InterCafe1 has low impact of faults in the classes. TermoProjekt1 has medium impact of faults in the classes and Zuzel1 has high impact of faults in the classes.

Table 5: Comparison of Standard Error Values Project-wise

Metric/Project	InterCafe1	TermoProjekt1	Zuzel1
WMC	0.535	0.612	0.875
RFC	0.543	0.594	0.707
CBO	0.514	0.650	0.994
LCOM	0.679	0.649	0.914
DIT	0.721	0.658	0.836
NOC	0.724	0.706	0.836

* Consider NOC=0.836 (as equal to DIT) in Zuzel1,The value of NOC is statistically not calculated, because of zero value in Zuzel1 Nasa data file.

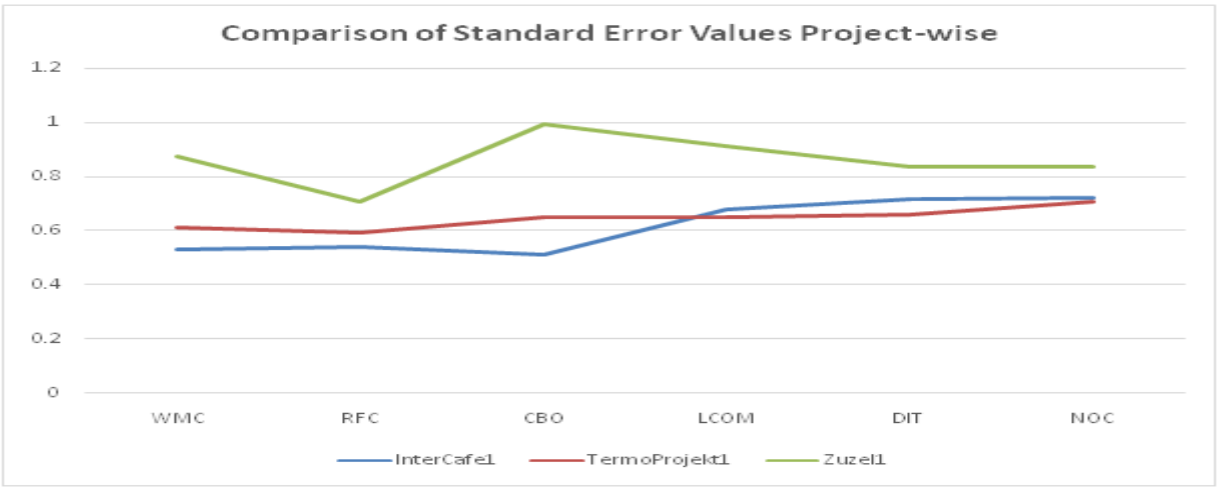

Figure 2: Line chart to show the Comparison of Standard error values project-wise

From Table 5 and Figure 2, the standard error for project Zuzel1 is high. The standard error for the project InterCafe1 project is low except LCOM, DIT and NOC metrics. The standard error for TermoProjekt1 is medium except LCOM, DIT and NOC metrics. The low value of standard error is indicates the low impact of the faults in the class. The high value of standard error indicates the high impact of faults in the class. Thus, project InterCafe1 has the low impact of faults in the classes. The project TermoProjekt1 has the medium impact of faults in the classes and the project Zuzel1 has the higher impact of faults in the classes.

The complete effect of the values comparing the values of Regression Coefficient, Standard error, 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

In real-life systems, the impact of fault varies drastically on the operation of a software system. It is wise and helpful for the developers to use OO design metrics that helps them to discover the fault-proneness of classes. This must be done while considering the impact of faults in the classes. The authors use the public domain data set InterCafe1, TermoProjekt1 and Zuzel1 available in the NASA repository. 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 classes, 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]. Sarker M., “An overview of object-oriented design metrics”, Thesis, Umea University, Sweden, pp.9-10, June 2005.
[2]. Chidamber S. and Kemerer C.: “A Metrics Suite for Object-oriented Design”, IEEE Transactions on Software Engineering, vol. 20, no. 6, pp. 476-493, June 1994.
[3]. N. Rajkumar et al., “Fault Prediction of Object Oriented Design using a Hybrid ANFIS Prediction Model”, International Journal of Advanced Engineering Technology, Vol. VII, Issue –III, pp. 179-183, 2016.
[4]. Madhu Rohilla1, P. K. Bhatia, “Prediction of fault-Proneness Using CK Metrics”, International Journal of Emerging Technology and Advanced Engineering, Volume-3, Issue-8, 2013.
[5]. JohnyAntoony P., “Predicting Reliability of Software Using Thresholds of CK Metrics”, International Journal of Engineering Research & Technology, Vol. 2, Issue 6, 2013.
[6]. S. R. Chidamber and C. F. Kemerer, Towards a metrics suite for Object- Oriented design, vol. 26, ACM, 1991.
[7]. T. Menzies, B. Caglayan, Z. He, E. Kocaguneli, J. Krall, F. Peters, and B. Turhan, The Promise Repository of Empirical Software Engineering Data, June 2012.
[8]. Goel, Brij Mohan, and Pradeep Kumar Bhatia. "Investigating of high and low impact faults in object-oriented projects", ACM SIGSOFT Software Engineering Notes, 2013.
[9]. V.R. Basili, L.C. Briand, and W.L. Melo, “A Validation of Object-Oriented Design Metrics as Quality Indicators,” IEEE Trans. Software Eng., vol. 22, no. 10, pp. 751-761, Oct. 1996.
[10]. L.C. Briand, J. Wust, J.W. Daly, and D.V. Porter, “Exploring the Relationships between Design Measures and Software Quality in Object-Oriented Systems,” J. Systems and Software, vol. 51, no. 3,pp. 245-273, 2000.