Software Bugs Identification and Prediction Approaches and their Interrelationships with Levels of Inheritance: An Empirical Analysis

Varuna Gupta, Tarun Kumar Singhal

Article Info
Volume 82
Page Number: 5750 - 5755
Publication Issue:
January-February 2020

Article History
***Article Received**: 18 May 2019*
***Revised**: 14 July 2019*
***Accepted**: 22 December 2019*
***Publication**: 28 January 2020*

Abstract:
The software applications are experiencing the challenges of ever-growing complexity caused by the increase in the number of bugs. The software development process has been adversely affected due to the wastage of resources caused due to the bugs. It is imperative to identify and predict bugs to facilitate the software development process and prevent rework. The researchers have successfully attributed metrics (product, process and project) as the dominant causes of the bugs. Through this work, the researchers aspire to spread awareness among the developers regarding prominent causes of the bugs while facilitating a smooth software development process.

Additionally, the researchers have correlated the effectiveness in the processes related to prediction of severities of the bugs corresponding to the levels and the levels of inheritance. The software quality is predominantly and adversely affected by product metricsoriented software bugs with significant correlation. Different levels of inheritance are associated with the severities of the bugs with a correlation that the severities are bound to rise with an increase in the levels of inheritance. The outcomes of this research work are of particular significance to software developers, software quality expertsandresearchers.The outcomes of this research work while promoting a set of practices to rationalize the inheritance suggest a lookout on severities of the bugs to roll out an acceptable quality of software applications. The outcomes of this research work also suggest rationalizing the efforts to identify and predictthe bugs.

Keywords: *Software Bugs, Bug Prediction, Reasons of Bugs, Neural Networks, Levels of Inheritance.*

I. INTRODUCTION

Quality Assurance (QA) is an activity that performs an essential role in software engineering. QA administers the software project to ensure that the final product or the software application meets the expectations of the end users. To judge the software quality, testing (quality assurance activity) is required. Testing is a time and efforts taken process. Several bugs remain uncoveredeven after rigorous testing using test cases. Main reason of these uncovers bugs is lack of time. So, for saving cost, time and efforts, there is a need to focus on this area. According to Gupta and Singhal (2015) have focused on size related software project metrics. They have noticed that detection of bugs using traditional waysis efforts intensive and costly. In this work, main focus was on awareness of important reasons of bugs in software application. The developer and the tester need to rationalize the coding efforts at an opportune time so that reasons causing bugs are essentially identified. Historical data is always for the prediction. The process of software development attaches high importance to prediction of software bugs.Severalbug indicators as the model input can be used with prediction modelsfor predicting the number of bugs. A comprehensive list of bug indicatorsusing prediction modelscan significantly improve the quality of software products. There is a growing need for improving the quality of software applications with judicious utilization of bug prediction approaches.

To avoid the possible harms of software products, bug severity needs to be identified. Artificial Neural Network (ANN) is required toenhance the prediction effectiveness. An attempt has been made forcorrelating levels of inheritance using the neural network with the severities of the bugs.

Objectives

1. To identify important reasons of software bugs that impact software quality.

2. To predict bug severitytypes like non trivial, major and critical through neural networkalgorithms.

3. Toexamine the effectiveness of neural network algorithms for predictionof severitiesthrough object-oriented projects.

II. LITERATURE REVIEW

Various studies have highlighted a number of reasons concerning software bugs. Jureczko and Spinellis (2000), Bansiya and Davis (2002) have also worked on software bugs and declarednumber of reasonsbehind software bugs. Somestudieslike Lorenz and Kidd (1994)have highlighted size and complexity relatedreasons, whereas others have focused on intrinsic dynamics of the project. Lorenz and Kidd (1994); Mende and Koschke (2009) have stated that for increasing the accuracy, number of parameters to identify the bug reasons can be increased. In their study they have proved that increasein the number of parameters with data from relevant entities may illustrate statistically significant results. This study has included more number of parameters to identify main reasons of bugs and light is thrown on reasons attached to product, process and project.

Most researchers have worked with open source projects (secondary datasets) for the studies. So, to get more accurate results, there is a need to work with primary data also (Industry data) for the analysis. This primary data will include the reasons together with expert opinions.

According to Wahyudin&Biffle (2008) and Sunghun (2011), CK Metrics have been extensively used in the prediction of bug proneness. The organizations have been using numerous statistical techniques for making the prediction. However, SRGM oriented prediction has not been holistically examined, supported by different bug indicators and machine learning methods. Software reliability growth modelling consistently focuses on the prediction of bugs(Catal and Diri(2007)). SRGM is considered one of the most effective tools for ensuring software reliability through prediction, estimation, control and assessment (Gupta and Singhal (2015)).Therefore, this research work has considered SRGM aspects with different bug indicators to arrive at precise outcomes. The prediction of software bugs is mostly dependent on the contribution of bug indicators. Therefore, a model under SRGM has been put forward with bug indicators to enhance the effectiveness of prediction of bugs. Different initiatives have been undertaken towards improvement in OO design (Qinbao 2011); Ebru and Parvinder 2009). OO metrics have been deemed as the most useful metrics in the prediction of bugs (Jureczko and Spinellis 2010).

A significant number of researchers have carried out a significant volume of work dedicated to the severities of bugs (Zhou and Leung (2006)). Developer related factors have been attributed to the capability of bug prediction models while concentrating on product and process metrics (Graves et al. 2000 et al.; Gupta V. et al. 2015). Several experiments have tried to use OO metrics through machine learning methods or neural networks for predicting the number of bugs (Tomaszewski et al. 2007;Purao and Vaishnavi 2003;Graves et al. 2000 andGupta V. et al. (2015). Other research works have used DIT as indicators of bugs (Lorenz and Kidd 1994, Qinbao 2011).However, these have not predicted the severities of bugs. Therefore, the gaps, as mentioned above, necessitate the need for including inheritance related indicators in the neural networks for prediction of the severities of the bugs.

III. NEED OF THE STUDY

The researchers have highlighted the following needs after conducting an extensive review of existing research literature:

- The prevention of software failures needs to be achieved by paying more attention topotential bug-prone areas.
- The quality of the project needs to be improved through rigorous shortlisting of bug indicators.
- The bug indicators caused with levels of inheritance need to be associated with the severities of bugs.
- The severities of bugs need to be predicted in association with an increase in the level of inheritance through artificial neural networks.

IV. ANALYSIS AND RESULTS

The work has been carried out in two parts for analysis.

A. Root Causes of Software Bugs

Total 627 (out of 1000) responses were received from various IT professionals. Exploratory Factor Analysis (EFA) was done on the responses received through survey. Rotated component metricsin Table 1.1 is depicting product-oriented reasons,Table 1.2 is depicting process-oriented reasons and Table 1.3 is depicting project-oriented reasons. Further, factors (three in number) were retained after applying factor analysis on 20 variables.

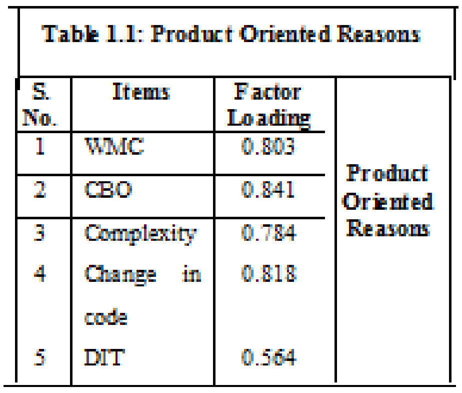

Table 1.1: Product Oriented Reasons

S. No.	Items	Factor Loading	
1	WMC	0.803	Product Oriented Reasons
2	CBO	0.841	
3	Complexity	0.784	
4	Change in code	0.818	
5	DIT	0.564	

Table 1.2: Process Oriented Reasons

S. No.	Items	Factor Loading	
6	Number of Committers	0.921	Process Oriented Reasons
7	Modified Lines	0.774	
8	Previous Bugs	0.869	
9	Number of Revisions	0.737	

Table 1.3: Project Oriented Reasons

S. No.	Items	Factor Loading	
10	Less Number of Planned Test Cases	0.780	Project Oriented Reasons
11	Less Number of Planned Milestone	0.722	
12	Workflow	0.755	
13	Uncover Problem	0.641	
14	Less Potential Risk	0.554	

Table 1.1is depicting the five root causes. Factor 1 is showing software product oriented reasons that become visible at the time of software development. Table 1.2, is depicting the four different root causes (variable 6 to 9)which is showing reasons related to process. Factor 2 depicts software process activities. Table 1.3 is depicting another five root causes (variable 10 to 14) as Factor 3 represent project-related problems.

Table 1.4: Reliability Test

Factors	Cronbach's Alpha
Product Oriented	0.874
Process Oriented	0.870
Project Oriented	0.788

Table 1.4 shows Cronbach's alpha values for software reasons related to product, process and project.

Various models based on WMC,CBO, LoC and DIT have been proposed. However, models using DIT are found to be more significant in prediction of the bugs (Gupta V and Singhal TK 2017).

B. Part 2: Predicting Bug Severity with Levels of Inheritance

In this section, severity prediction has been done with the help of diverseinheritance's levels (from level 1 to level 5) and types. The typesof inheritance is considered as multilevel, multiple, hierarchical and hybrid. Figure 1.1 is depicting the proposed neural network structure involving four inputs with one output. In this work, four training functions with 10 and 20 neurons have been utilized to measure the accuracy of these training functions. Parameters of accuracy comparison of these four training functions are mean square error (MSE), R on training,testing, and validation.

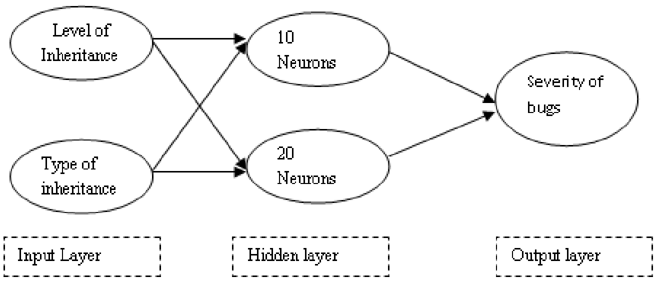

Figure 1.1: Structure of Neural Network (WFS)

Work Force Scheduler (WFS)dataset contains 172 entries related to bug severities and levels of inheritance. One input represents different levels of DIT (from level 1 to level 5) and another input is types of inheritance. Table 1.5 and Figure 1.2are depicting the status of level -1 of inheritance and number of non-trivial, major and critical bugs at this level.

Table 1.5 Predicting Bug's Severity at Level-1 of Inheritance

DIT Level - 1			
Training Function	nontrivial Bugs	Major Bugs	Critical Bugs
LM(10)	26	0	0
LM(20)	26	0	0
BFG(10)	26	0	0
BFG2(0)	26	0	0
GD(10)	26	0	0
GD(20)	26	0	0
GDM(10)	26	0	0
GDM(20)	26	0	0

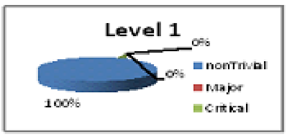

Figure 1.2 Bug's Severity at Level-1

Table 1.5represents that more non-trivial bugs exist at level-1 of inheritance than major and critical bugs.Figure 1.2 is depictingthe growing impact of predicted non-trivial bugs over major and critical bugs at level -1 through graphical representation.

Table 1.6 and Figure 1.3are representing estimated values of non-trivial, major and critical bugs at level -2 of inheritance.

Table 1.6 Predicting Bug's Severity at Level-2 of Inheritance

DIT Level -2			
Training Function	nontrivial Bugs	Major Bugs	Critical Bugs
LM(10)	0	24	0
LM(20)	0	24	0
BFG(10)	0	24	0
BFG2(0)	20	4	0
GD(10)	0	24	0
GD(20)	0	24	0
GDM(10)	3	21	0
GDM(20)	4	20	0

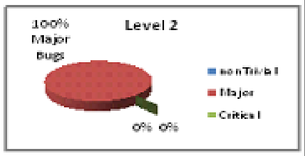

Figure 1.3 Bug's Severity at Level-2

Table 1.6is representing that the number of major bugs are more than non-trivial and critical bugs at level-2 of inheritance. Figure 1.3 is depicting the growing impact of predicted major bugs in comparison of non-trivial and critical bugs at level -2 through graphical representation.Table 1.7 and Figure 1.4are representing the estimated values of non-trivial, major and critical bugs at level -3 of inheritance.

Table 1.7 Predicting Bug's Severity at Level-3 of Inheritance

DIT Level -3			
Training Function	nontrivial Bugs	Major Bugs	Critical Bugs
LM(10)	0	41	3
LM(20)	0	41	3
BFG(10)	0	41	3
BFG2(0)	0	0	44
GD(10)	0	41	3
GD(20)	0	41	3
GDM(10)	3	0	44
GDM(20)	4	0	44

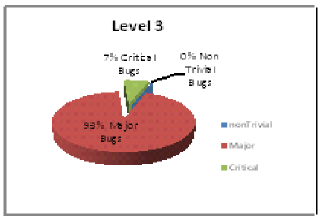

Figure 1.4: Bug's Severity at Level-3

Table 1.7representshigher number of major bugs at level-3 of inheritancethan non-trivial and critical bugs. Figure 1.4, is depicting the growing impact of predicted major bugs in comparison of other bugs at this level through graphical representation.
Table 1.8 and Figure 1.5are representing the estimated values of non-trivial, major and critical bugs at level -4 of inheritance.

Table 1.8: Predicting Bug's Severity at Level-4 of Inheritance

DIT Level -4			
Training Function	nontrivial Bugs	Major Bugs	Critical Bugs
LM(10)	0	2	63
LM(20)	0	2	63
BFG(10)	0	0	65
BFG2(0)	0	0	65
GD(10)	0	5	60
GD(20)	0	5	60
GDM(10)	0	0	65
GDM(20)	0	2	63

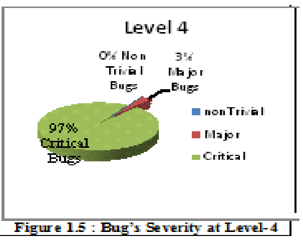

Figure 1.5 : Bug's Severity at Level-4

Table 1.8representshigher number of critical bugs at level-4 of inheritance than non-trivial and major bugs. Figure 1.5is depicting the growing impact of critical bugs in comparison of other bugs at this level through graphical representation.
Table 1.9 and Figure 1.6are representing the estimated values of non-trivial, major and critical bugs at level -5 of inheritance.

Table 1.9: Predicting Bug's Severity at Level-5 of Inheritance

DIT Level -5			
Training Function	Nontrivial Bugs	Major Bugs	Critical Bugs
LM(10)	0	0	13
LM(20)	0	0	13
BFG(10)	0	0	13
BFG2(0)	0	0	13
GD(10)	0	10	3
GD(20)	0	10	3
GDM(10)	0	0	13
GDM(20)	0	0	13

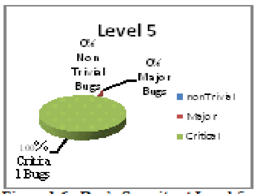

Figure 1.6 : Bug's Severity at Level-5

Table 1.9representshigher number of critical bugs at level-5 of inheritance than non-trivial and major bugs. Figure 1.6 is depicting the growing impact of critical bugs in comparison of other bugs at this level through graphical representation.

After analyzing the above tables, It is imperative to conclude that severities of the bugs are intricately correlated with levels of inheritance. Table 1.10has produced analytical results of two neural network algorithms (i.e. Quasi Newton and Gracient Descent) using four functions of these algorithms (trainlm, trainbfg, traingd and traingdm)for observing MSE, R for training, testing and validation.

Table 1.10: Accuracy Measurement of Training Functions

Algorithms	Training Functions	Neurons	Best validation MSE	R on training	R on testing	R on validation
Quasi Newton	Trainlm	10	0.105	0.766	0.715	0.843
		20	0.119	0.763	0.673	0.838
	Trainbfg	10	0.174	0.726	0.816	0.865
		20	0.166	0.746	0.740	0.855
Gradient Descent	Traingd	10	0.216	0.683	0.692	0.652
		20	0.290	0.721	0.721	0.792
	Traingdm	10	0.177	0.767	0.611	0.801
		20	0.276	0.712	0.662	0.823

The hidden layer has 10 and 20 neurons calibrated for the parameters. To achieve value of MSE close to zero, the network is trained. The trainlm function renders MSE at close to 0.1 using 10 and 20 neurons. The relationship between bugs severities of and the inheritance levels is being depicted in Figure 1.7.

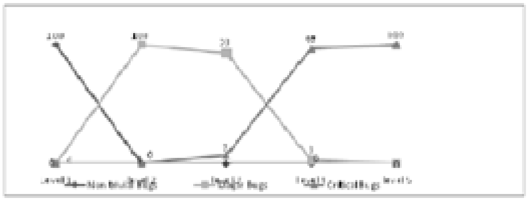

Figure 1.7:Correlationbetween Bug Severities and Inheritance Levels

A significantly strong relationship is depicted between non-trivial bugs and level 1 of inheritance. With level 2 and level 3 of inheritance, major bugs are found associated. Finally, level 4 and level 5 are found to be associated with critical bugs. The general interpretation suggests increase in the severities of the bugs with the increase in the levels of inheritance. Table 1.11 is depicting the correlation of bugs severities with levels of inheritance.

Table 1.11: Impact of Type of Inheritance on the Severities of Bugs

Level of Inheritance	Type of Inheritance	Severities of bugs
Level 1	Single Inheritance	Non- Trivial
Level 2	Multilevel Inheritance	Major
Level 3	Multilevel Inheritance	Major, Critical
Level 4	Multilevel , Hierarchical Inheritance	Critical
Level 5	Hierarchical , Multiple Inheritance	Critical

CONCLUSION AND FUTURE SCOPE

The product-oriented, process-oriented and project-oriented reasons majorly contribute in creation of bugs. The software developers and quality practitioners need to pay sufficient attention to product-oriented reasons for curbing the bugs in development environment.Also, the severities of the bugs are found correlated with increase in levels of inheritance. The inheritance levels need to be suitably rationalized to effectively control the bugs severities.

REFERENCES

1. Jureczko, M., Spinellis D., "Using Object-Oriented Design Metrics to Predict Software Defects", In Models and Methods of System Dependability, Pp., 69-81, 2010.
2. BANSIYA J., and DAVIS C. G., "A Hierarchical Model for Object-Oriented Design Quality Assessment", IEEE Trans. on Software Engineering, Vol., 28, No. 1, Pp. 4-17, 2002.
3. Lorenz M. and Kidd J., "Object-Oriented Software Metrics.", Prentice-Hall, 1994.
4. Mende T., Koschke R., "Revisiting the Evaluation of Defect Prediction Models.", Proc. of PROMISE, 2009.
5. Ebru A ,Erdem U., and Parvinder S., "Software Maintenance Severity Prediction with Soft Computing Approach', Proceedings of World Academy of Science, Engineering and Technology , Vol. 38, pp. 139-143, 2009.
6. Qinbao Song, Zihan Jia, Martin Shepperd, Shi Ying, and Jin Liu, "A General Software Defect-Proneness Prediction Framework", IEEE Transactions on Software Engineering, Vol. 37, No. 3, May/June 2011
7. Zhou Y. and Leung H., "Empirical analysis of object-oriented design metrics for predicting high and low severity faults", IEEE Transaction Software Engineering, vol. 32, no. 10, pp. 771-789, 2006
8. Sunghun Kim, Hongyu Zhang, Rongxin Wu and Liang Gong, Dealing with Noise in Defect Prediction', CSE'11, Waikiki, Honolulu, HI, USA, ACM 978-1-4503-0445-0/11/05, 2011.
9. Purao S. and Vaishnavi V. K., "Product metrics for object-oriented systems", ACM Computing Surveys, Vol. 35, No. 2, Pp. 191-221, 2003.
10. Tomaszewski, P., Håkansson, J., Grahn, H., & Lundberg, L., " Statistical models vs. expert estimation for fault prediction in modified code–an industrial case study", Journal of Systems and Software, Vol. 80, No.8, pp. 1227-1238, 2007.
11. Gupta V., Ganeshan N. and Singhal T.K., "Developing Software Bug Prediction Models using various Software Metrics as the Bug Indicators", IJACSA, vol 6, no 2, pp. 60-65, 2015.
12. Gupta V., Ganeshan N. and Singhal T.K., "Determining the Root Cause of Various Software Bugs through Software Metrics", In the proceedings of 9th International Conference on "Computing for Sustainable Global Development " IEEE, pp. 45-49, 2015.
13. Wahyudin D., Ramler R. and Biffl S., A framework for Defect Prediction in Specific Software Project Contexts. Proc. of the 3rd IFIP CEE-SET, Pp. 295-308, 2008.
14. Catal C. and Diri B., 'Software Defect Predication Using Artificial Immune Recognition System', Proceedings of the 25th IASTED International Multi – Conference Software Engineering (2007), Innsbruck, Austria, ISBN Hardcopy:978-0-88986-641- 61/CD:978-0-88986-643-0.
15. Graves T.L., Karr A.F. , Marron J.S. and Siy H.. "Predicting fault incidence using software change history," IEEE Trans. Softw. Eng., vol.26, no.7, 653–661, 2000.

Dr. Varuna Gupta is working as an AssistantProfessor of Computer Science with Dr. A.P.J. Abdul Kalam Technical University, India. She holds MCA degree from Dr. A.P.J. Abdul

Kalam Technical University and MPhil and Ph.D. Degrees from Christ University Bangalore.Her research interests include Software Testing, Software Metrics, Bug Prediction and Search-Based Software Engineering.

Dr. Tarun Kumar Singhalis an assertive yet humble academician with over 25 years of credible experience in teaching, research, consulting, advisory and training & development domains. He is presently working as Professor at Symbiosis Centre for Management Studies Noida (Constituent of Symbiosis International (Deemed University), Pune). His research interests include Social Media, Internet of Things,