Complexity estimation model: Fault perspective

[1]Fateh Bahadur Kunwar,
M.Tech Scholar
Department of Computer Science and Engineering, KNIT, Sultanpur

[2]Dr.Neelendra Badal
Professor
Department of Computer Science and Engineering, KNIT, Sultanpur

ABSTRACT

Complexity is major issue for software quality. Software quality highlights must be synchronized in every steps of software development process. The quality parameters are directly affecting the reliability attributes like Fault, Correctness, Complexity and Consistency. Object oriented design is accepted and established conceptions in today's software development process. The aim of this research work is to empirically explore the relationship between faults factors at design level and software complexity. The proposed paper introduces a **Metric based "Complexity Assessment Model" CAM^{DF}** to reduce fault in design phase using logistic regression technique.

I INTRODUCTION

Measuring complexity of object oriented software ahead of schedule in the development procedure, especially at design stage extraordinarily decrease the general development cost and reduce, and moreover architects and engineers for creating superb maintainable and reliable software product within time and budget. In a technical report McCall et al. (1977) introduced a complexity definition of fault affecting software reliability [24]. Software Quality model is a basic to gained information with the goal that moves can be made to enhance the execution. Different specialists have proposed unique software quality models to help measure the quality of software items. The Software industries are giving careful consideration to authoritative fault in any software framework is extremely normal and complex issue. Software complexity constantly increases with fault handling purpose [1]. Software products with high complexity generally develop software system with faults issues. Such change can be estimated quality, expanded customer satisfaction and reduce cost of quality. Software metrics and quality models assume a significant part in estimation of software quality [9]. Various positively understood characteristics models are utilized to produce quality software. Keeping a software framework from mistakes is such a difficult task. Software metrics are the quantitative estimation of the complexity of the software, so they are great possibility for directing the determination of testing systems [8]. The proposed model for complexity assessment point outs the possible manipulate of design constructs and also discusses the impact of complexity with fault issues at design.

II ADDRESSING COMPLEXITY AND FAULT AT DESIGN STAGE

During the systematic literature has been observing an important number of

reliability studies. For measuring and estimating software complexity have effect on it which need to be identified before entered the development stages. Many complex programs are in genuine word is effectively facilitating the humankind. Indeed, even the space carries program or the complex structure of fly military aircraft and so on. According to Booch, complexity can be determined either at the development stage or after deployment. Development use of object oriented technology forces the development of objects oriented software metric in software development life cycle [6]. Considering the paradigm of object oriented approach, one of the issues is to estimate the complexity of it with fault construct and what fault parameters at design level are being affected. Many researchers in the area recommended that fault issues should be done at design stage and their view is summarized in table1.

Table 1 Critical Analysis of fault factors by various Experts

	Reliability Factors				
Experts	**Correctness**	**Fault**	**Completeness**	**Consistency**	**Complexity**
Rizvi (2016) [20]	√		√	√	
G.P. et. al. (2015) [19]		√	√		√
Dalal (2014) [17]	√		√	√	√
Pandey (2013) [18]	√	√		√	√
Abdullah (2013)[13]	√	√	√		√
A. Yadav (2012) [21]	√		√	√	√
Zainab (2011) [15]	√	√	√	√	√
Vennila (2011)[16]		√	√	√	√
Al-Qutaish (2010) [12]		√		√	√
R. A Khan (2009) [14]	√		√		√

III RESEARCH OBJECTIVES

In order to achieve the goal for addressing Complexity for reliability index at design phase the following objectives are set:

- ✓ To review and critically examine the literature on complexity, quality specification, fault factors, design phase, verification and validation.
- ✓ To identify quality goals with fault issues.
- ✓ To develop framework for addressing reliability to reduce fault at design stage.
- ✓ To validate and test the proposed framework.
- ✓ To implement the framework to ensure the software specification that contains the reliability specification which helps to improve the fault free application and reduce the fault of the software product.
- ✓ To validate the proposed model
- ✓ To proposed complexity index with guidelines

IV MODEL DEVELOPMENT

Assessment of complexity can be performed through measurement and there are distinctive methodologies and approaches accessible which are either hypothetical or best hones as for actualize complexity. Complexity measurement will push the product designer to accomplish the Reliability objectives and split down the cost of reuse. A fault show is expected with a specific end goal to evaluate complexity at design time in order to illuminate the relationship amongst fault issues and complexity. The key of model has been utilized to characterize the subjective highlights of design measurements that can evaluate through prerequisite points of view. Figure 1 show the importance of study in order to establish a contextual relationship between faults attributes and reliability factors such that reliability can be quantified with available set of design.

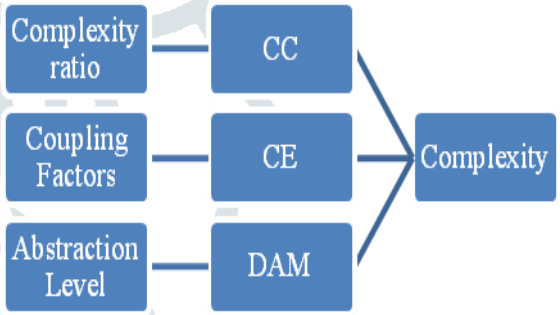

Figure 1 Correlation among Fault Factor and Reliability Factors

V EMPIRICAL VALIDATION

The relationship between the Reliability factors and fault attributes are based on virtual importance of individual factors which shows a major effect on Reliability at design time that directly correlate the quality traits and is proportionately evaluated. The coefficient is acquired with the help of multiple linear regression line. Multiple regression equation is established with association shown among the data variables as dependent data and multiple independent data. The proposed multivariate model takes the following form:

Where

• Y is dependent variable

• X1, X2, X3 ... Xn are independent variables.

• $\alpha_1, \alpha_2, \ldots \alpha_n$ are the regression coefficient of the respective independent variable.

• α_0 is the regression intercept.

The data utilized for establishing complexity show is taken from [22] that have been gathered through expansive business question arranged systems.

$$Y=\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X3 + \ldots\ldots \alpha_n X_n \quad \text{Eq (1)}$$

Standard values of complexity have taken from [23]. The connection between unwavering quality factor and blame elements has been set up as portrayed in Figure 1. According to the mapping, Metrics are chosen from [22] as independent variable to develop the complexity appraisal display by means of SPSS, estimations of coefficient are computed and complexity show is detailed as given beneath.

Table III Complexity Computed Table

Project	Standard Complexity	Cyclomatic Complexity	Coupling Efferent	Direct Access Measurement
P_1	0.976	0.6667	3	0.5
P_2	0.925	1.6667	2	0
P_3	0.919	1.1	12	0.25
P_4	1.258	1.7083	15	0.916667
P_5	1.413	1.9470	2	1.00000

DEVELOPED EQUATION^{Complexity} = 0.678 + 0.158 CC - 0.00462 CE + 0.426 DAM

Eq (2)

The model summary of conscious data is mentioned in table 4 which imparts the statistical explanation of used data and signifies the high value of R Square represents that model is highly correlated.

Table IV Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.985[a]	.971	.958	.013298
a. Predictors: (Constant), CC, CE, DAM				

In order to validate proposed complexity Estimation model, the value of metrics are available by using [22] data set for following 10 projects in table V.

Table 5 Complexity Data Table

Project	Cyclomatic Complexity	Coupling Efferent	Direct Access Measurement	Calculate Index
P_1	1.31	3.00	1.00	1.30
P_2	1.52	3.00	.92	1.29
P_3	1.98	8.00	.70	1.25
P_4	1.36	.00	.60	1.15
P_5	1.13	5.00	1.00	1.26
P_6	1.38	20.00	.95	1.21
P_7	2.22	7.00	.60	1.25
P_8	1.40	7.00	1.00	1.29
P_9	.80	2.00	1.00	1.22
P_{10}	1.00	3.00	.92	1.22

VI STATISTICALSIGNIFICANCE

For any approval of proposed model it is mandatory to check the standard of correctness. A 2-tail student sample test has been applied to analyze the dissimilarity between two population means. In table 6 the 2-t test examination of complexity model values are shown.

Table 6 T –test (one sample) analysis

Paired Samples Statistics					
		Mean	N	Std. Deviation	Std. Error Mean
Pair 1	Calculate Index	1.2441	10	.04673	.01478

Table 7 Test Examine View

One-Sample Test						
	Test Value = 1.24					
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference	
					Lower	Upper
cal	.277	9	.788	.00410	-.0293	.0375

Null hypothesis (H0): The value of Complexity index is not similar

H0: μ1-μ2 = 0

Alternate hypothesis (HA):): The value of Complexity index is similar

HA: μ1-μ2 ≠ 0

In the above hypothesis μ1 and μ2 are treated as sample means of population.

Mean value and Standard Deviation value have been calculated for specified two samples and represented in table 7. The hypothesis is tested with zero level of significance and 95% confidence level. The p value is 0.788. Therefore null hypothesis directly discards and the alternate hypothesis is accepted. The developed equation used for complexity assessment is accepted.

VI Conclusion

A Model (CAM^{DF}) has been created to evaluate Complexity from necessity point of view at the initial phase of improvement of software. It appraises the Complexity with reflect to fault parameters which are influenced according to their effect. A multiple linear relapse technique is done to measure the model. The early measurement implies the nature of software at the early phase of SDLC. Thus measurement of Complexity reduce the fault at the initiation of the software i.e., at design stage. The predictable model has been approved and factual investigation suggests the acceptance of the model.

REFERENCES

1. Marian Cristescu, Laurenţiu Ciovica, "Estimation of the Reliability of Distributed Applications", Informatica Economică vol. 14, no. 4/2010, pp: 19-29.
2. Goel, B. And P.K. Bhatia, 2013. Investigating of High and Low Impact Faults in Object-Oriented Projects, ACM SIGSOFT Software Engineering Notes, 38: 6.
3. Subramanyam, R., Krishnan, M.S., "Empirical analysis of CK metrics for object -oriented design complexity: Implications for software defects", IEEE Transactions on Software Engineering, vol. 29, no. 4, pp.297-310, April 2014.
4. Shatnawi R., 2012. Improving Software FaultPrediction for Imbalanced Data, IEEE Innovations in Information Technology (IIT), pp: 54-59.
5. Bansiya, Jagdish, and Carl G. Davis. "A hierarchical model for object-oriented design quality assessment." Software Engineering, IEEE Transactions on 28.1 (2002): 4-17
6. G. Booch, "Object-Oriented Design and Application", Benjamin/cummings, Mento Park, CA, 1991.
7. Madeyski, L. And M. Jureczko, 2014. Which process metrics can significantly improve defect prediction models? An empirical study, Software Quality Journal, pp: 1-30
8. Venkata U.B challagula,Farokh B.Bastani,"Emperical Assessment of Machine Learning based Software Defect prediction Techniques ",10 th IEEE International Workshop on Object Oriented Real Time Dependable Systems,2008.
9. Shreya Gupta, Ratna Sanyal,"Advanced object oriented metrics for process measurement", The Sixth International Conference on Software Engineering Advances, Barcelona, Spain, pp.318-324, October 2011.
10. Brij Mohan Goel, Pradeep Kumar Bhatia, "Investigation of Reusability Metrics for Object-Oriented Designing", Proceeding of NCETCIT, GVM IT&M, Sonipat, pp. 104-110, May 2014.
11. Z. A. Rana, S. Shamail, M.M. Awais, Ineffectiveness of Use of Software Science Metrics as Predictors of Defects in Object Oriented Software, 2009WriWorld Congress on Software Engineering, Vol 4, Proceedings, IEEE Computer Society, 2009.
12. Al-Qutaish, R. E., "Quality Models in Software Engineering Literature: An Analytical and Comparative Study," *Journal of American Science*, 2010. Vol. 6, no. 3, pp. 166-175.
13. Abdullah, Dr, Reena Srivastava, and Dr. M. H. Khan. "Testability Estimation of Object Oriented Design: A Revisit."International Journal of Advanced Research in Computer and Communication Engineering, Volume 2, Issue 8, Pages 3086 -3090, 2013.
14. Yadav, A. and Khan, R.A. 2009b. Measuring Design Complexity–An Inherited Method Perspective. ACM Software Engineering Notes, 34(4), 1-5.
15. Zainab Al-Rahamneh, Mohammad Reyalat, Alaa F. Sheta, Sulieman Bani-Ahmad, Saleh Al-Oqeili,, "A New Software Reliability Growth Model: Genetic-Programming-Based Approach", Journal of Software Engineering and Applications, 2011, Vol: 4, PP:476-481 doi:10.4236/jsea.2011.48054.
16. Vennila G, Anitha P, Karthik R, Krishnamoorthy P. A study of

evaluation information to determine the software quality assurance, International Journal of Research and Reviews in Software Engineering. 2011;1(1):1-8.

17. Dalal, S. R., Lyu, M. R., and Mallows, C. L. 2014. Software Reliability. John Wiley & Sons.
18. Pandey, A. K., and Goyal, N. K. 2013. Early Software Reliability Prediction. Springer, India.
19. Jaiswal, G.P. and Giri, R. N. 2015. A Fuzzy Inference Model for Reliability Estimation of Component Based Software System. International Journal of Computer Science and Technology, 3(3), 177-182.
20. Rizvi, S. W. A., Singh, V. K., and Khan, R. A. 2016. The State of the Art in Software Reliability Prediction: Software Metrics and Fuzzy Logic Perspective. Advances in Intelligent Systems and Computing, Springer, 433, 629-637
21. Yadav, A., and Khan, R. A. 2012a. Reliability Quantification of an Object-Oriented Design: Complexity Perspective. Proceedings of the Second International Conference on Computer Science, Engineering and Applications (ICCSEA 2012), May 25- 27, 2012, New Delhi, Advances in Intelligent and Soft Computing, Springer, 166, 577-585.
22. M. Jureczko & L. Madeyski, "Towards identifying software project clusters with regard to defect prediction", IEEE, 2010.
23. Nidhi Gupta et. al. , "Reliability Measurement of Object Oriented Design: Complexity Perspective", International Advanced Research Journal in Science, Engineering and Technology Vol. 2, Issue 4, April 2015
24. McCall, J. A., Richards, P. K., Walters, G. F. "Factors in Software Quality, Volumes I, II, and III," 1977. US Rome Air Development Center Reports, US Department of Commerce, USA.
25. Pradeep Kumar Singh et. al., "A Critical Analysis on Software Fault Prediction Techniques", World Applied Sciences Journal 33 (3): 371-379, 2015.
26. Zunnon Khan, "Fault Criteria at Design Phase", IJSRD, Vol. 5, Issue 05, 2017.