

# Completeness estimation model: Fault perspective

<sup>1</sup>Fateh Bahadur Kunwar,  
M.Tech Scholar

Department of Computer Science and  
Engineering, KNIT, Sultanpur

<sup>2</sup>Dr. Neelendra Badal  
Professor

Department of Computer Science and  
Engineering, KNIT, Sultanpur

## ABSTRACT

Reliability is an important software quality factor that is more effective at an in the development software. It is more effective in the case of object oriented design approaches. Completeness is a significant key attribute to reliability analysis. Its evaluations provided the maintainable and highly reliable software. Completeness is a reason of important impact to software developers. This paper evaluated on the need and importance of completeness for reliable software. A model has been proposed for completeness assessment with fault issues at design level by establishing multiple linear regressions. Finally the completeness model has been validated using experimental examine.

## I INTRODUCTION

Quality has turned out to be more vital with our expanding reliance on software. In the last decades the interest for quality in software products has been progressively underscored. PC industry has been conveying exponential change in cost, execution; however the issues with software are not declining. Software still conveyed is late, goes past spending plan

and is brimming with lingering faults. According to the most current IBM report, 31% of the tasks get crossed out before they are finished, 53% over-run their cost estimates by a normal of 189%, and for every 100 activities, there are 94 restarts. A key issue of software industry is its absence of capacity to create bug free software [1, 4].

On the off chance that software engineers are asked to formally express that the created software is bug free, no software would have ever been discharged. Goal of software designing is to make fantastic software in time and inside spending plan. In the event that a product is meeting its prerequisites, we may state it is a prevalent quality product [2, 7]. The entire thing is estimated regarding prerequisites and in the event that it matches, product is a quality product. Software has turned out to be imperative to headway in every aspect of human undertaking. The capacity of software just is not any more agreeable to manufacture large projects. There are significant issues in the cost, opportuneness, upkeep and quality of numerous software products. Software

designing has the objective of taking care of these issues by creating great quality, reliable, viable software, on time, inside spending plan [3]. As indicated by software building standards, if the procedure for improvement of any software is correct, the possibility of accomplishment of the software ventures is significantly expanded.

To achieve this target, think about needs to center in a trained way around both the quality of the product and on the procedure used to build up the product. Notwithstanding, because of increment in cost of testing and support of software, objective is presently changing to convey quality software. Software reliability is a vital and important movement of software advancement life cycle for delivering improbable software [8, 9]. Reliability is imperative and reliable assignment.

The time spent and exertion required for software completeness is extremely critical and expends around 40% to half of the aggregate cost for the whole advancement

life cycle. The most imperative issue among reliability is that before amending a program (faults), the developer should first follow and comprehends it and it is conceivable with the assistance of its completeness [10, 11]. In view of the fact, it is clear that completeness holds a significant position as part of reliability.

## II. IMPACT OF CORRELATION

The figure1 portrays the quantification procedure of completeness display keeping in mind the end goal to build up a multivariate model for completeness and object oriented design faults develops. The estimations of these metrics can be effortlessly distinguished by class diagram metrics. This metrics (Coupling Efferent and Measure Functional Abstraction) will assume the part of autonomous while privacy will be taken as independent variable. The quantifiable appraisal of completeness is extremely useful to accomplish unwavering quality record of software outline for reliable the product.



**Fig 1 Correlation between Fault factors and Quality factors**

### III MODEL DEVELOPMENT

It is clear from review study that completeness isn't another term; rather it has been in advocate among the research experts at different gatherings, however there is no generally acknowledged exhaustive and finish model or structure accessible to evaluating the secrecy through given blame variables at configuration stage, that propel to create '**Completeness Assessment Model**' (CAM<sup>DF</sup>) utilizing deficiency characteristics and approach in view of its interior outline property or configuration diagram. This model utilized the low level outline metrics in particular Cyclomatic Complexity, Cohesion among methods, to portray a scope of estimation for software and characterized as far as plan trademark and furthermore accommodating for quantitative assessment of degree to which framework, segment or process hold a given property. Keeping in mind the end goal to set up a model for completeness, multiple linear regression strategies have been utilized. The proposed multivariate model takes the accompanying structure:

$$Y = \alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X_3 + \dots + \alpha_n X_n$$

*Eq (1)*

Where

- Y is dependent variable
- X<sub>1</sub>, X<sub>2</sub>, X<sub>3</sub> ... X<sub>n</sub> are independent variables.
- $\alpha_1, \alpha_2, \dots, \alpha_n$  are the regression coefficient of the respective independent variable.
- $\alpha_0$  is the regression intercept.

The information utilized for building up completeness model is taken from [5] that have been gathered through extensive industry protest arranged frameworks. Using SPSS software estimations of every single free factor (metrics), regression catch and coefficient of the particular autonomous factors are figured and shown in table 1. Completeness esteems have chosen from [6] as standard esteems. Based on this approach, the multiple linear regression privacy models have been created that is given in equation 2.

Table 2 shows the calculated index for completeness. Model summaries (table 3) are presented the correlation history of developed model between dependent variables and in dependent variables.

Table 1 Calculated Table

Standard Completeness	CE	MFA
0.783	3	0.937500
0.687	3	0.822222
0.879	9	0.750000
0.924	4	0.921053
0.834	10	0.962500
0.735	2	0.987179

Formulated Completeness<sup>DF</sup> = 0.695 + 0.0139\* CE + 0.046 \*MFA Eq (2)

Table.2 Correctness Data Table

Project	Calculate Completeness	CE	CAM
P <sub>1</sub>	.767	2.000	.969
P <sub>2</sub>	.781	3.000	.970
P <sub>3</sub>	.780	3.000	.949
P <sub>4</sub>	.761	2.000	.841
P <sub>5</sub>	.779	3.000	.721
P <sub>6</sub>	.764	2.000	.902
P <sub>7</sub>	.749	1.000	.875
P <sub>8</sub>	.757	2.000	.750
P <sub>9</sub>	.750	2.000	.581
P <sub>10</sub>	.789	4.000	.841

Table 3 Model Summary

Model Summary				
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.981 <sup>a</sup>	.963	.952	.003077
a. Predictors: (Constant), CE MFA				

#### IV VALIDATION

It is essential to check the strength of proposed model for acceptance. A one sample t test has been initiated to test the difference between one population means i.e., computed correctness values. Hypothesis is depending on the p value rule.

If P value  $\geq 0.05$  then Null accepted otherwise Alternate.

Null hypothesis (H<sub>0</sub>): The value of **Completeness** index is similar.

Alternate hypothesis (H<sub>A</sub>): The value of **Completeness** index is not similar.

H<sub>0</sub>:  $\mu_1 - \mu_2 = 0$  versus H<sub>A</sub>:  $\mu_1 - \mu_2 \neq 0$

Table 4 T –test (one sample) analysis

		Paired Samples Statistics				
		Mean	N	Std. Deviation	P Value	Std. Error Mean
Pair 1	Calculate Completeness	.76795	10	.014066	0.836	.004448

The hypothesis is tested with zero level of significance and 95% confidence level. The p value is 0.836. Therefore null hypothesis is accepted. The developed equation used for **completeness** assessment is accepted.

#### V CONCLUSION

Completeness is a standout amongst the hugest components for estimating reliability of objects oriented programming design. Concentrate built up a completeness estimation demonstrate that sets up the connection among completeness, blame issue at design properties and object oriented

measurements. This paper demonstrates the noteworthiness of completeness as a key factor of reliability and its association with different object oriented design properties. Completeness estimation show in design stage has been created and approved hypothetically and additionally empirically utilizing trial experiment with. For test approval a few extensive business ventures has been utilized. The connected approval on the completeness display presumes that created demonstrate is exceedingly dependable, up to similar index.

## References

1. Heimdahl M. P. E. and Leveson N. G., "Completeness and Consistency in Hierarchical State-Based Requirements", IEEE transactions on Software Engineering, Vol 22, No. 6, June (1996).
2. Marinescu, Radu, and Daniel Ratiu. "Quantifying the quality of object-oriented design: The factorstrategy model", Reverse Engineering, 2004. Proceedings. 11th Working Conference on. IEEE, 2004.
3. 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.
4. ISO 9001:2005, Quality management system Fundamentals and vocabulary, 2005.
5. M. Jureczko & L. Madeyski, "Towards identifying software project clusters with regard to defect prediction", IEEE, 2010.
6. P.Nikhat, S. Kumar and M. H. Khan, "Model to Quantify Integrity at Requirement Phase", Indian Journal of Science and Technology, Vol. 9, Aug 2016.
7. Anshul Mishra, D. Agrwal & M. H. Khan, "Integrity Estimation Model: Fault Perspective", International Journal on Recent and Innovation Trends in Computing and Communication, Vol 5, Issue 5, pp 1246-1249, May 2017.
8. M. Al-Kuwaiti and S. Hussein, "A Comparative Analysis of Network Dependability, Fault Tolerance, Reliability, Security and Survivability", IEEE, Vol. 11, No. 2, 2009.
9. K. K. Aggarwal, Y. Singh and J. K. Chhabra, "An Integrated Measure of Software Maintainability", Proceedings Annual Reliability and Security Symposium, pp 235-241 2002.
10. R. Haesen, M. Snoeck, Implementing consistency management techniques for conceptual modelling, in: Proceedings of the 3rd International Workshop, Consistency Problems in UML-based Software Development III – Understanding and Usage of Dependency Relationships, held at UML 2004, 2004, pp. 99–113.
11. E. Arisholm, L.C. Briand, and A. Foyen, "Dynamic Coupling Measurement for Object-Oriented Software," IEEE Trans. Software Eng., vol. 30, no. 8, pp. 491-506, Aug. 2004.