IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⚟ Impact Factor 7.12 ⚟ Vol. 10, Issue 1, January 2023

DOI: 10.17148/IARJSET.2023.10134

Software Quality Reliability Estimation Model: Fault Perspective

Mr. Fateh Bahadur Kunwar[1], Dr. Mayank Patel[2], Dr. Narendra Singh Rathore[3]

Assistant Professor, CSE, GITS, Udaipur, India[1]

Professor, CSE, GITS, Udaipur, India[2]

Campus Director, CSE, GITS, Udaipur, India[3]

Abstract: Reliability is very important factor for judging software quality. A number of researchers have proposed qualitative and quantitative approaches to measure and improve the reliability of software. In systematic literature review, a plethora of reliability improved models have been used to evaluate it. The factors of the reliability model along with the part of faults issues at design stage have been presented in this paper. "Reliability Estimation Model" has been developed by establishing the co- relationship among reliability factors with fault issues and reliability, mitigating the relationship with the help of statistical observation. This paper helps to designer for developing of reliability estimation model to access and evaluate reliability for software quality.

Keywords: Software Quality factors, Design Metrics, Fault Perspective, Software Reliability

I. INTRODUCTION

The ISO/IEC 9126 standard depicts a model for software quality that analyzes the general idea of quality into 6 fundamental characteristics i.e: usefulness, reliability, usability, efficiency, maintainability, and portability [1]. These qualities are additionally subdivided into 27 sub-attributes. Besides, the standard gives a consensual stock of measurements that can be utilized as pointers of these attributes. The ID and meaning of software quality attributes of the ISO quality model gives a helpful edge of reference and standardized wording which encourages correspondence concerning software quality [2]. The characterized measurements give direction to a later assessment of these attributes in light of exertion and time spent on exercises identified with the product item, for example, affect examination, blame amendment, or testing. Shockingly, the recorded measurements are estimated by implication and need prescient power [3]. Talking casually, they anticipate tomorrow's climate to be the same as today's. As a rule, the proposed measurements for surveying the maintainability qualities are not estimated straight forwardly regarding the matter of support, i.e. the framework's source code and documentation, in any case, in a round about way on the execution of the support movement by the specialized staff.

II. RELIABILITY: QUALITY FACTOR

In software engineering, it is broadly acknowledged that the quality of at last conveyed item is profoundly subject to the early choice in the development procedure. A precise assessment of software quality relies upon Reliability estimation, which relies upon the factor that can influence reliability [4]. Delivering high-quality, enterprise-grade software on-time and under budget requires balancing several competing priorities, and the complexity and completeness of such software is directly related to its reliability. The importance of reliability is not novel [5]. The "reliability factor," as it is known in the industry, has inspired a great deal of software advancements aimed at improving product quality. At the time of configuration, dependability plays a significant role in determining an application's overall quality, and it also serves as the foundation for locating reliability lists used to place industrial projects [6]. The essential standard for assessing quality factor is reliability. On the off chance that a segment isn't reusable then the entire idea of segment based software development falls flat.

III. FAULT FACTORS: DESIGN STAGE

Authorities being to blame in any software system is a common and difficult problem, and the software industry is giving it significant thought. Keeping a software framework from mistakes is such a troublesome assignment. Faults must be checked at the beginning periods of the item development as it spares time and cost. Similarly it helps in diminishing complexity at design times, which additionally influences other quality characteristics. In the event that complexity is less it suggests that the exertion required to test a program would be less and item would be more dependable [7].

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⋇ Impact Factor 7.12 ⋇ Vol. 10, Issue 1, January 2023

DOI: 10.17148/IARJSET.2023.10134

An incredibly common and challenging problem in any software system is authoritative fault, and it is receiving significant scrutiny from the software industry. Error-proofing a software architecture is a tedious task. Since software metrics are quantifiable estimates of programme complexity, they are excellent candidates for regulating the selection of testing methods [8]. In the realm of question arranged dialects, software metrics has been utilized for a long time to furnish designers with extra data about their software quality. Software metrics can screen the quality of software. Keeping a software framework from fault is such a troublesome undertaking. Quality of software is progressively imperative for software[9]. For enhancing quality we ought to give more spotlights on testing to those segments of code which have largest number of faults.

IV. RELIABILITY ESTIMATION MODEL

The metric-based estimating model takes into account concerns of completeness and complexity during the design phase, and for this reason, a reliability estimation model has been established. Completeness, complexity, and dependability are all described in detail in Fig. 1, along with the estimating method itself.

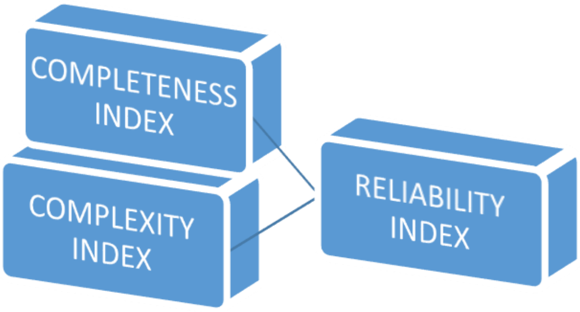

Fig : 1 Relationship view

When estimating dependability during the design process, it's important to give due consideration to a number of different criteria, each of which is based on a different connection. Class diagrams may be used to determine the values of these complexity and completeness problems. Multiple regression was used to get the coefficients and regression intercept in equation 2 in order to establish a metric-based model for reliability assessment. With reliability as the dependent variable, the identified reliability components will be measured as independent variables. Accurate reliability estimates are crucial for calculating the design's dependability index, which is a measure of high-quality software. Multivariate regression equation is given in Equation (1) which is as follows

$$Y=\alpha_0 + \alpha_1 X_1 + \alpha_2 X_2 + \alpha_3 X3 + \ldots\ldots.. \alpha_n X_n \qquad (1)$$

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.

When taking into account the fault problem during the design phase of software development, it has been widely agreed that comprehensiveness and complexity are the most important elements influencing reliability estimate. To ensure dependability, the previously listed criteria were taken into account throughout the design process. By applying the statistical analysis, Complexity Estimation Model [10] and Completeness Estimation Model were developed and that is given in Equations (2) and (3) respectively.

Complexity = 0.678 + 0.158 CC - 0.00462 CE + 0.426 DAM ……………………. (2)

Completeness = 0.695 + 0.0139* CE + 0.046 *MFA ……………………. (3)

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

ISO 3297:2007 Certified ⋇ Impact Factor 7.12 ⋇ Vol. 10, Issue 1, January 2023

DOI: 10.17148/IARJSET.2023.10134

Table I: Reliability Calculation

Project	Standard Complexity	Standard Completeness	Standard Reliability
P_1	0.976	0.783	0.250
P_2	0.925	0.687	0.258
P_3	0.919	0.879	0.690
P_4	1.258	0.924	0.800
P_5	1.413	0.834	0.625

The values of "Complexity, Reliability and Completeness" have been used from equation [1, 2, and 3]. Table 1 displays the results of a correlation analysis performed in SPSS, and the model of Reliability Estimation is then built as shown in Equation (4).

RELIABILITY = - 1.60 + 0.182 COMPLEXITY + 2.34 ……………………… (4)

The results of the model evaluation, which is essential when running multiple regressions, are shown in Table 2. R, the multiple correlation coefficient, is shown in the table to indicate the degree to which a given dependent variable is affected by a given set of independent variables. The "R square" coefficient of determination is promising.

Table 2: Model Summery

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.999[a]	.998	.998	.00676
Predictors: (Constant) Completeness, complexity				

Table 3's equations [1, 2] provide the values of completeness and complexity that may be used to verify the generated model. The proposed model has been put through experimental statistical analysis to determine the reliability index value of the projects, and then reliability estimations have been generated.

Table 3: Reliability Data

Project	Complexity	Completeness	Cal_Reliability	Std_Reliability
P_1	1.300	.767	.431	.421
P_2	1.290	.781	.462	.438
P_3	1.250	.780	.453	.466
P_4	1.150	.761	.390	.290
P_5	1.260	.779	.452	.490
P_6	1.210	.762	.408	.396
P_7	1.250	.749	.380	.376
P_8	1.290	.757	.406	.382
P_9	1.220	.750	.377	.383
P_{10}	1.220	.789	.468	.472

DOI: 10.17148/IARJSET.2023.10134

V. COMPARTIVE STUDY

[3] built a model for early-stage reliability estimation of object-oriented software. The complexity and fullness of the proposed model are used to calculate dependability. The values of reliability index as are composed to standard values and also compared the correlation with model summary as shown in fig 2.

Table 4: Comparative Analysis

S. No	Reliability Model	Correlation	Impact analysis
1	Wajahat A. Rizv, [3]	0.936	High
2	Reliability Proposed Model	0.999	Maximum

VI. CONCLUSION

Measurement models for complexity, completeness, and dependability have been created to track the index values throughout the design process. The bulk of the research should be dedicated to comparing the study's three main aspects: dependability, completeness, and complexity. The implementation can benefit in measuring reliability index for the software or even to be used in different session. The assessment result proves that the method can be used as a structure and effort to measure reliability of software. In conclusion, by measuring the quality of software using multiple regression models, it could assist designers to improve their reliability.

REFERENCES

[1] Fateh Bahadur Kunwar and Neelendra Badal "Complexity estimation model: Fault perspective" Journal of Emerging Technologies and Innovative Research (JETIR), September 2018, Volume 5, Issue 9, (ISSN-2349-5162).

[2] Fateh Bahadur Kunwar and Neelendra Badal "Completeness estimation model: Fault perspective" Journal of Emerging Technologies and Innovative Research (JETIR), September 2018, Volume 5, Issue 9, (ISSN-2349-5162).

[3] Syed Wajahat A. Rizvi et.al. "Early Stage Software Reliability Modeling using Requirements and Object-Oriented Design Metrics: Fuzzy Logic Perspective", International Journal of Computer Applications (0975 – 8887) March 2017 Volume 162 – No 2.

[4] Yadav A., and Khan, R. A., '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.

[5] Kong, W., "Towards a Formal and Scalable Approach for Quantifying Software Reliability at Early Development Stages", Ph.D. Thesis University of Maryland, 2008.

[6] Sen, S., Patel, M., Sharma, A.K. (2021). Software Development Life Cycle Performance Analysis. In: Mathur, R., Gupta, C.P., Katewa, V., Jat, D.S., Yadav, N. (eds) Emerging Trends in Data Driven Computing and Communications. Studies in Autonomic, Data-driven and Industrial Computing. Springer, Singapore. https://doi.org/10.1007/978-981-16-3915-9_27

[7] Hooshmand, A. and Isazadeh, A., "Software Reliability Assessment Based on a Formal Requirements Specification", Proceedings of the Conference on Human, System Interactions, Publisher IEEE Krakow, Poland, 2008, P 311-316.

[8] Offutt, J. and R. Alexander : A fault Model for Subtype Inheritance and Polymorphism. In 12th International Symposium, Software Reliability Engineering, Nov 27-30, 2001, IEEE, pp. 84- 93.

[9] Ameta, U., Patel, M., Sharma, A.K. (2021). Scrum Framework Based on Agile Methodology in Software Development and Management. In: Mathur, R., Gupta, C.P., Katewa, V., Jat, D.S., Yadav, N. (eds) Emerging Trends in Data Driven Computing and Communications. Studies in Autonomic, Data-driven and Industrial Computing. Springer, Singapore. https://doi.org/10.1007/978-981-16-3915-9_28

[10] A. Mishra, D. Agarwal and M. H. Khan, "Confidentiality Estimation Model: Fault Perspective" International Journal of Advanced Research in Computer Science (IJARCS), June 2017 Volume.8 Issue. 4.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License