A Systematic Literature Review On Reliability: 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 **-**In the present scenario reliability and fault are two major component for evaluating software quality at design phase. Many researchers and authors are putting input in this domain for error free software development. This paper reviewed their work and present a comprehensive study on reliability and fault.

Keywords: *-Reliability Factor, fault factor, Quality factor, design phase***.**

I INTRODUCTION

Software designing practices have risen as the most recent point of talk among Information Technology experts. It has been felt by the information framework researchers that more research and improvement are required at the front of software building hones. US original Standish disorder report (1995) anticipated that in an example of IT anticipate just 16% has their goals met, somewhere in the range of 31%, were just utilized by a few clients for a few purposes and that 53% were never at any point introduced or fizzled/pulled back after establishment. An imperative result from the Standish report was the way that sentiment regarding why these IT anticipates fizzled positioned "deficient necessities" and "absence of client association" at the best of the rundown. Another face of this angle is unwavering quality of Software item or capacity of item to act in the event that of blunder and disappointment. In order to software quality can be measured in terms of attributes of the system.

II. Software Quality

Software quality is turning into a vital part in software design, helping the designer to deal with the many-sided quality of extensive frameworks. While designing, the planner ought to dissect the framework prerequisites before conferring the assets to it. The investigating process encourages us to guarantee the high quality of engineering design. For as long as decade, there were numerous examining techniques are utilized, which thus to dissect just the perspectives of single partner. Thusly, there are numerous restrictions that prompt basic circumstance in the improvement process. They expounded this circumstance to extreme measure of time to play out the total examination. The extent of finding the key structural choice is exceptionally troublesome.A quality attribute is the property of a software framework. A quality necessity is a prerequisite that is put on a software framework by a partner; a quality attribute is the thing that the framework as a matter of fact shows once it has been actualized. Amid the improvement of the engineering it is consequently critical to approve that the engineering has the required quality attributes, this is generally done utilizing at least one engineering assessments. In order to reliability and fault is an important attributes of quality. So focusesare on the following quality attributes.

2.1 RELIABILITY AT DESIGN STAGE

With the growing errors, it is hard to maintain the criticality of software and increasing inadequacy at desired quality level and reliability become more essential, complicated and expensive [3]. In the highly competitive software industry, customer pressure causes Companies to go faster the speed to deliver software products [6]. Reliability has two components, namely, completeness and complexity. The assessment of the reliability of software is of major concern for buyers of large scale software systems found in fast moving domains. As software system grows more complex and start on increasingly to return human decision-makers in every facet of our life, software quality and reliability requires careful attention [1].The proposed work for reliable software is an important research area, specifically in the early design stage of the software development. In fact, reliability should be designed and built into taking the fault factors and the system at the earliest possible stages of software development. Reliability targeted design isthe most economical advance to minimize the life-cycle efforts of the product or system. One can achieve better product or software reliability at less effort by the utilization of these techniques. Otherwise, the majority of software life-cycle processes are locked in phases other than design and development.Reliability is the essential property of referring ‘how well software meets its requirements’ & also ‘the probability of fault, errors free operation for the specified period of time in a specified environment [20].Assess reliability at a later stage leads to the late arrival of desired information, leading to late decisions about changes in design [28].Therefore, early assessment of reliability in the software development process may improve quality and reduce fault, efforts and costs. Reliability Metrics are used to know the possibility of failure or failure rate of software where system error is occurs. Software reliability quantification has attracted immense interest from researchers as well as software practitioners since the early 1990’s. Traditional methods for quantifying the software-reliability such as reliability growth models estimates reliability on the basis of the defects observed during validation testing, where operational patterns represent how actually the product would be used [26]. A consolidated chart for the

Reliability factors identified by variousexperts is concluded in given table No.1.

Authors	Years	Correctness	Completeness	Testability	Reliability	Complexity
Surbhi [33]	2017	√	√		√	√
Monika [13]	2016	√	√	√	√	√
K. Anil [31]	2015		√	√	√	√
H. B. Yadav [30]	2015	√	√		√	√
G. P. Jaiswal [29]	2015	√		√	√	√
R. A. Khan [32]	2012	√	√	√	√	√
D. Yuan [28]	2011		√	√	√	
W. Kong [27]	2009	√	√	√	√	
Hooshma[25]	2008	√		√	√	√

Table 1 Critical view of quality issues

2.2 FAULTS AT DESIGN STAGE

All factors that are fault which determines the probability of occurrences attacks and also play a striking responsibility with other factors of quality. Capability of fault is signified by the hiatus of unsecure software [10].The faults are found during the testing and the failure is when the system stops working. Fault attacks can be deployed in software which generally helps to avoid, detect and correct faults [13]. During development of software, faults and flaws are introduced either from the implementation or from the design of the software. It relies on software attributes that how much system’s software is protected; exactly the fault decides for how longer software will be protected. Fault is not only the factor that makes things hard to understand but with enough difficulty anything can become harder to understand. In this paper, we conduct a study regarding impact of fault to quality estimation and their efficiency. There is need to develop a new advance to deal with a word of fault in software systems.The required goal of reliability is to introduce measures and procedures that preserve effectiveness, functionality and fault. The goal of increasing the reliability of software is not just to detect fault but more importantly, to detect faults at design stage as soon as they are introduced [2]. Therefore, reducing the price and overall time to fix the bug and producing higher reliable software each build of the release of development cycle. Many experts and researchers in the area recommended that fault is the ability of a software system or component to continue normal operation despite the software fault, errors and the similar is concise in table 1 and also shown the fault at design stage in table 2.

Table: 2 A Critical Observation Table: consider of fault contributions byvarious Experts / Researchers

Sr. No.	Expert/ Researchers	Contribution with Fault
1.	W. T. Pomales (2000) [2]	Demonstrated Software Fault Tolerance
2.	Saridakis and Titos (2004) [7]	Provided a new concept about Design Pattern for Fault
3.	Y. Jiang (2007) [9]	Highlighted the fault detection techniques used Life cycles
4.	E. J. Weyuker (2008) [3]	Enhanced to defect prediction models for software
5.	Dhiman (2009) [12]	Provided Algorithm Based Classification Approach for Finding Fault Prone
6.	A. Nugroho (2010) [5]	Provided design metrics for predicting fault-prone classes in a Java system
7.	D. Mougouei et.al (2012) [10]	Introduced a measurement model for evaluating the degree of fault
8.	Ahmet Okutan (2013) [14]	A Novel Regression Method provided for Software Defect Prediction
9.	Monika (2016) [13]	Provided Method for fault prediction using ck metrics
10.	Zunnun Khan (2017) [34]	Described Fault Criteria at design Stage.

III. CRITICAL OBSERVATIONS

After successful completion of the systematic literature review some important. Critical observations are as follows.

- A complete estimation of software reliability at an initial stage that is design phase in the software development process is highly recommended by researchers.

- Estimation of software reliability at design phase maygreatly improve the software quality, user satisfaction, and reduce effort of rework.
- For minimizing effort in measuring fault of objectoriented design, one needs to identify a minimal set of commonly accepted set of the reliability factors early in design phase for object oriented development process, which have positive impact on reliability estimation.

IV. CONCLUSION

- Numbers of various approaches have been proposed in the systematic literature review for software fault and Reliability.
- Numbers of different software reliability are presented in different researches in different perspectives by practioners /researchers.

Fault is a factor which affect on reliability most. Its feature depends on some quality attributes and affects directly. By controlling the design characteristics and their relation it is possible to develop a product which is more reliable and reduce fault. This paper will give a view to show the effect of reliability factors construct fault and impact of software quality.

REFERENCES

1. S. S. Gokhale, "Architecture-Based Software Reliability Analysis: Overview and Limitations", IEEE Transactions on Dependable and Secure Computing, vol. 4, no. 1, (2007). Natasha Sharygina , James C. Browne, and Robert P. Kurshan, "A Formal Object-Oriented Analysis for Software Reliability: Design for Verification", 2011.
2. Wilfredo Torres-Pomales, "Software Fault Tolerance: A Tutorial", NASA/TM-2000-210616, October 2000.
3. E. J. Weyuker, T. J. Ostrand, R. M. Bell, Do too many cooks spoil the broth? Using the number of developers to enhance defect prediction models, Empirical Software Engineering, (2008).
4. N. Fenton, S. L. P. fleeger and R. Glass, "Science and Substance, A Challenge to Software Engineers", IEEE Software, July 1994.
5. Nugroho, M. R. V. Chaudron, E. Arisholm, Assessing UML design metrics for predicting fault-prone classes in a Java system, IEEEWorking Conference on Mining Software Repositories, 2010.
6. Natasha Sharygina , James C. Browne, and Robert P. Kurshan, "A Formal Object-Oriented Analysis for Software Reliability: Design for Verification", 2011.
7. Saridakis and Titos, "Design Pattern for Fault Containment", 2004.
8. Nikora, J. Munson, Building high-quality software fault predictors, Software: Practice and Experience, 2006.
9. Jiang, B. Cukic, T. Menzies, Fault Prediction using Early Lifecycle Data, The 18th IEEE International Symposium on Software Reliability (ISSRE '07) , 2007.
10. D. Mougouei& M. Almasi, "A Goal based Modeling Approach to Develop QualityRequirements of Fault Tolerant Quality Critical Systems", IEEE, 2012.
11. D. Wahyudin, A. Schatten, D. Winkler, A. M. Tjoa, S. Biffl, Defect Prediction using Combined Product and Project Metrics A Case Study from the Open Source "Apache" MyFaces Project Family, Proceedings of the 34th Euromicro Conference on Software Engineering and Advanced Applications, IEEE Computer Society, 2008.
12. Dhiman, S.K. and Goyal, A.,"A Genetic Algorithm Based Classification Approach for Finding Fault Prone Classes", World Academy of Science, Engineering and Technology, Vol. 60, pp.485- 488, 2009.
13. Monika, "Prediction of Fault-Proneness using CK Metrics", International Journal of Computer Science and Information Technology Research, Vol. 4, Issue 3, pp: (114-118), Month: July - September 2016.
14. Ahmet Okutan and OlcayTanerYildiz, "A Novel Regression Method for Software Defect Prediction with Kernel methods", Proceedings of the international conference on pattern recognition and applications, pp. 216 – 221, 2013.
15. ShatnawiR.,"Improving Software Fault- Prediction for Imbalanced Data" IEEE Innovations in Information Technology (IIT), 2012.
16. NavpreetKaurȦ* and Aman Paul, "Fault Prediction Modeling using Object-Oriented Metrics: An Empirical Study", International Journal of Current Engineering and Technology, Vol.4, No.1, February 2014.
17. Seliya, N. and T. Khoshgoftaar, "Predicting Faults in High Assurance Software", IEEE High-Assurance Systems Engineering (HASE), 2010.
18. Jintao zeng, Jinzhong Li, Xiaohui Zeng, Wenlang Luo "A Prototype System of Software Reliability Prediction and Estimation". IITSI 2010.
19. RazeefMohd,MohsinNazir,September 2012,"Software Reliability Growth Models: Overview and Applications",Journal of Emerging Trends in Computing and Information Sciences,Vol.3, No.9, pp-1309-1320.
20. Goel, B. And P.K. Bhatia, " Investigating of High and Low Impact Faults in Object-Oriented Projects", ACM SIGSOFT Software Engineering Notes, 2013.
21. Hans Erik Eriksson, Magnus Penker, Brain Lyons and David Fado;"UML 2 Tool kit", Willey publication, 2007.
22. EQ-Mine: Predicting short-term defects for software evolution, Fundamental Approaches to Software Engineering, 2007.
23. F. Garcia, M. Bertoa, C. Calero, A. Vallecillo, F. Ruiz, M. Piattini, M. Genero, Towards a consistent terminology for software measurement, Information and Software Technology 48 (2006).
24. A Web Service Reliability Prediction Using HMM and Fuzzy Logic Models, Open Access Procedia Computer Science, Volume 93, 2016.
25. Hooshmand and A. Isazadeh, Software Reliability Assessment Based on a Formal Requirements Specification, In Proceedings

of the Conference on Human System Interactions, Publisher IEEE Krakow, Poland, pp. 311–316, 2008.

26. Fiondella, L.; Gokhale, S.S., “Software reliability model with bathtub-shaped fault detection rate” Reliability and Maintainability Symposium (RAMS), 2011 Proceedings - Annual , 24-27 Jan. 2011,pp: 1 – 6, ISBN: 978-1-4244-8857-5
27. W. Kong, Towards a Formal and Scalable Approach for Quantifying Software Reliability at Early Development Stages, Ph.D. Thesis, University of Maryland, 2009.
28. D. Yuan and C. Zhang, Evaluation Strategy for Software Reliability based on ANFIS, In (ICECC-11) Proceedings of the IEEE International Conference on Electronics and Communications and Control, pp. 3738–3741, 2011.
29. G. P. Jaiswal and R. N. Giri, A Fuzzy Inference Model for Reliability Estimation of Component Based Software System, International Journal of Computer Science and Technology, vol. 3(3), pp. 177–182, 2015.
30. H. B. Yadav and D. K. Yadav, A Fuzzy Logic based Approach for Phase-wise Software Defects Prediction using Software Metrics, Information and Software Technology, vol. 63, pp. 44–57, 2015.
31. K. Anil and D. Namrata, Reliability Estimation of Object-oriented Software: Design Phase Perspective, International Journal of Advanced Research in Computer and Communication Engineering, vol. 4(3), pp. 573–577, 2015.
32. Yadav and R. A. Khan, “Reliability Quantification of Object-Oriented Design: Complexity Perspective”, In (ICCSEA 2012), Proceedings of the 2nd International Conference on Computer Science, Engineering and Applications, New Delhi, India; vol. 1, pp. 577–585, 2012.
33. SurabhiSaxena1 ,Dr.Devendra Agarwal2. “A Systematic Literature Review on Software Reliability Estimation Model for Measuring the Effectiveness of Object Oriented Design”, International Journal of Advanced Research in Computer and Communication Engineering, 2017.
34. Zunnun Khan, “Fault Criteria at Design”, IJSRD, 2017.