

# International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: [www.ijarcsms.com](http://www.ijarcsms.com)

## *Reliability Quantification of Object Oriented Software: A Revisit*

**Abhinav Kumar Chand<sup>1</sup>**

M.Tech. Scholar, Department of Computer Science  
& Engineering, G.I.T.M.  
Lucknow – India

**Namrata Dhanda<sup>2</sup>**

Associate Professor & Head, Department of Computer  
Science and Engineering, G.I.T.M.  
Lucknow – India

*Abstract: Reliability is one of the most significant quality indicators. It's perfect measurement frequently support and increase the development process. A truthful measure of software quality fully depends on reliability measurement. A number of reliability theories have been published till date and the reliability notion has been developed with different research states. On the other hand, reliability has always been a mysterious theory and its truthful measurement or evaluation is a complex exercise. Researchers and practitioners have always argued that reliability should be considered as a key characteristic in order to assure the quality software. At this paper we will talk about some of the essential theories given by some researches in their paper and we will relate those all research with our paper in order to encourage our effort.*

*This paper shows the outcome of a systematic literature review conducted to gather information on software reliability quantification of object oriented design. In this review paper, our endeavor is to discover the existing known comprehensive and complete model or framework for measuring the reliability of object oriented design at an initial stage of development life cycle.*

*Keywords: Software reliability, reliability quantification, Software metrics, Object Oriented Design, Software Quality.*

### I. INTRODUCTION

Software Engineering has turn into extremely essential discipline of study, practice and research. Everyone are working hard to decrease the problems and to meet the purpose of developing high-quality reliable software that is delivered on time, within budget, and furthermore satisfies the requirements. Software has become significant to expansion in almost all areas of human endeavor. The skill of programming only is no longer enough to make large programs. There are serious problems in the price, timeliness, maintenance and reliability of many software products. Software engineering has the purpose of solving these problems by producing feature- quality reliable software within time, financial plan. Software reliability is the probability that the software will work without failure for a specified stage of time. Failure means the program in its functioning has no met user requirements in several way. Software reliability concerns itself with how well the software functions to meet the requirements of the customer. Reliability represents a user oriented view of software quality. Reliability is associated with unexpected failures of products or services and understanding why these failures occur is key to improving reliability. The main reasons why failures occur include:

- The product is not fit for purpose or more specifically the design is inherently incapable.
- The item may be overstressed in some way.
- Failures can be caused by wear-out
- Failures might be caused by variation.
- Wrong specifications may cause failures.

- Misuse of the item may cause failure.
- Items are designed for a specific operating environment and if they are then used outside this environment then failure can occur.

It is clear that to ensure good reliability the causes of failure need to be identified and eliminated. Without a doubt the objectives of reliability engineering are:

- To relate engineering information to stop or decrease the likelihood or frequency of failures;
- To recognize and correct the causes of failure those do occur;
- To decide ways of coping with failures those do occur;
- To apply methods of estimating the likely reliability of innovative designs, and for studying reliability facts.

## II. SOFTWARE RELIABILITY

Software reliability is a key attribute to software quality [3]. Reliability is the property of referring 'how well software meets its requirements' & also 'the probability of failure free operation for the specified period of time in a specified environment' [1]. Software reliability defines as the failure free operation of computer program in a specified environment for a specified time [4, 5, 6 and 7]. Unreliability has a number of unfortunate consequences and therefore for many products and services is a serious threat. Software Reliability is a vital to trait of software quality, jointly with functionality, usability, performance, serviceability, ability, installability, reliability, and documents. Software reliability is inflexible to achieve, since the complexity of software tends to be elevated. Although any system with a high level of complexity, including software, will be inflexible to reach a positive level of reliability, system developers tend to move forward complexity into the software layer, with the fast growth of system size and affluence of doing so by upgrading the software.

For instance poor reliability can have implications for:

- Safety
- Competitiveness
- Profit margins
- Charge of repair and maintenance
- Delays further up supply chain
- Reputation
- Good will

### *Growing the software reliability and quality*

#### *a) Faults and Failures:*

It is the departure of the outside results of program operation from requirements. So failure is self-motivated. It depends upon the process and activities. Failure is not the similar object as a 'Bug' or 'fault'. There is a group of difference among these two terms. A fault is a defect in a program which arises when programmer makes an error and causes failure when executed under meticulous conditions.

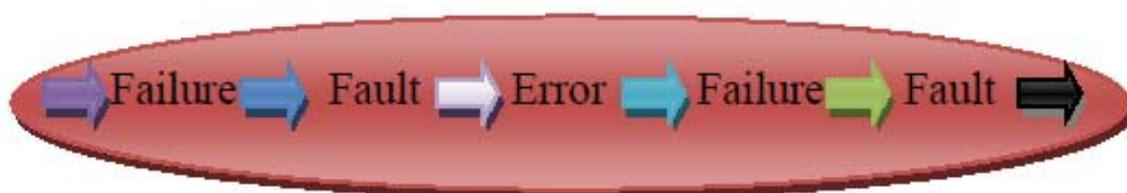


Figure 1. Fault Behavior

**b) Increasing Reliability**

Reliability can be increased by preventing the above understood errors and developing quality software throughout all of the stages of software development life cycle. To do this, we have to make sure that whether the requirements are undoubtedly specifying the functionality of the final product or not. Amongst the phases of the software reliability, the second one i.e. useful life is the most important one and so the software product must be maintained carefully. So we have to guarantee that the code generated can maintain reliability to avoid any additional errors.

**III. RELIABILITY ESTIMATION AT DESIGN PHASE**

As reliability is an attribute of software quality, we can say that reliability fully depends on software quality. So to construct a high reliable software there is a necessitate to measure the attributes of quality that are applied at each development cycle mainly at design phase. Software metrics are used to quantify these applicable attributes. Measuring software reliability in the early hours in the development life cycle, particularly at the design phase, may facilitate designers to incorporate essential improvement and corrections for improving reliability of the final software. If we can predict the reliability at the early stages of the software development, the cost of the software can be economical. Reliability uniqueness indicate the product's ability to be changeable, maintainable, and updateable. Early estimation of reliability helps to utilize its attributes more competently to control/improve the quality of software. The design phase quantification of software reliability is more practicable for software development and maintenance economically.

- Reliability is a assess of doubt and for that reason quantifying reliability means via statistics and probability theory
- Reliability is quality over time
- Reliability be required to be planned into a product or service
- Most central characteristic of reliability is to recognize grounds of failure and remove it in design if probable or else ecognize ways of accommodation.
- Reliability is describe as the capability of an item to carry out a mandatory function without failure under stated circumstances for a stated phase of time
- The expenditure of unreliability can be harmful to a company

S. No.	Authors /	Year	SDLC Phase
01	Abdullah	2014	Design Phase
02	P.Nikfard	2013	Design Phase
03	P.Malla	2012	Design Phase
04	Nazir et al.	2010	Design Phase
05	R A Khan	2009	Design Phase
06	Jerry et al.	2005	Design Phase
07	S.Mouchawrap	2005	Design Analysis
08	Jungmayr	2004	Design Phase
09	Wang	2003	Design Phase
10	Jungmayr	2002	Design Phase
11	Bach	1999	Design Phase
12	Binder	1994	Design Phase
13	J Voas et al.	1992	Design Phase

TABLE 1. Quality Estimation at Design Phase consider by various expert [8]

## IV. OBJECT ORIENTED DESIGN CHARACTERISTICS

Procedural-oriented languages centre of consideration on procedures, all the way through function as the basic unit. You need to primary figure out all the functions and after that consider about how to represent data. The object-oriented languages main point on components that the user perceives, by means of objects as the elementary unit. You outline all the objects by putting all the data and operations that demonstrate the user's communication with the data. Object-Oriented technology has a group of compensation:

- *Simplicity in software design* as you could assume in the problem space relatively than the machine's bits and bytes. In OOD you are dealing with high-level concepts and abstractions. Easiness in design direct to more dynamic software development process.
- *No difficulty in software maintenance*: object-oriented software is easier to understand, as a result easier to test, debug, and maintain.

Object Oriented Programming has great advantages over other programming styles: The object-oriented technology is very well-liked in software development atmosphere in recent years. More and more organizations are launching object oriented technique and languages into their software development practices [14]. Object Oriented Software procedure is an approach of structuring software as a group of distinct objects reflecting real-world entities and mapping them into design constructs to characterize relationships and functionality powerfully. It is a sign of an accepted view of the domain and handles inherent complexity improved. Object oriented method presents forthcoming benefits over traditional software development approach. The function of object oriented technology to systems development has brings a lot of reimbursement and benefits as well as new demanding issues [29].

The object-oriented technology is more trustworthy to design the software in order to make available the product of higher quality. The recognition of the object-oriented approach is credible to produce improved and cheaper software [1]. Three important concepts formulate a distinction the object-oriented approach from conventional software engineering: Coupling, Encapsulation, Cohesion and Inheritance & Polymorphism [1]. These concepts play major role of design properties in object-oriented software product quality assessment. in conclusion, show the light on the function of a variety of object oriented design properties such as polymorphism, encapsulation, inheritance, coupling and cohesion on quality attributes such as reliability, effectiveness, understandability and reusability in order to assess the object oriented software product quality.

OO Design Properties →	Cohesion	Coupling	Encapsulation	Inheritance	Abstraction
Author/Study ↓					
MC Gregor et al. (1996)		√	√		
Bruce & Shi(1998)		√	√		
B.Pettichord(2002)		√			
Baidry et al.(2002)		√			
M Bruntik (2004)			√		

<b>S.Mouchawrab (2005)</b>	√	√		√	
<b>I.Ahson et al.(2007)</b>	√	√	√	√	
<b>Nazir et al.(2010)</b>	√	√	√	√	
<b>Suhel et al.( 2012)</b>	√	√	√	√	√
<b>Khan et al. (2012)</b>	√	√	√	√	
<b>Nikfard &amp; Babak(2013)</b>		√	√	√	

TABLE 2. OO Design Characteristics Contributing in Reliability Quantification at Design Phase: A Close Appear

## V. RESEARCH METHODOLOGY

A systematic literature review is a practice of recognizing, estimating and understanding the existing research result significant to a particular research area or subject [2]. The study in research area has mainly divided into two categories primary and secondary studies. Primary study is an individual studies contributing to the research and secondary study is a systematic review of other research related to the research area, topic or observable fact of interest [2]. The enthusiasm for choosing systematic literature review as methodology of study are to sum up the existing body of knowledge regarding the research of concern, to recognize the gap in current research and to present framework/ background for further examination. In this perspective, Study select the systematic review to sum up the existing concepts of reliability factors and measurement in software engineering and apply that knowledge to build up a reliability assessment framework/model for reliability estimation.

The justification for selecting this methodology is:

1. Systematic literature review's healthy defined methodology helps to decrease the bias for selecting primary studies.
2. Its systematic process enables consistency in study selection and quality estimation of primary studies.
3. Its result serves as input for advance framework creation

The systematic literature review has the following steps [2].

1. Data source collection
2. Search approach development
3. Search string creation
4. Study selection criteria recognition

## VI. SYSTEMATIC LITERATURE REVIEW

A number of reliability concepts have been published till date and the reliability concept has been grown with different research areas. At this study we talk about a number of the essential theories specified by some researches in their paper and we will relate those all research through our study in order to encourage our work. Various studies below provide some inspiration regarding the related work on this area.

Stage one: 1950-1967 Subject develop period; software reliability did not attract concentration.

Stage two: 1968-1987 Subject configuration periods, Software Engineering was recognized and developed; mathematical models of software reliability began to come into view.

Stage three: at this time Subject developing period, Software Reliability Engineering was put onward, software reliability transits from hypothetical research to engineering application. More and more importance has been attached to software reliability. Software Reliability Security Design Criteria in the year 1997 as well [3].

### 1) *Theoretical research*

The first document on software reliability is the Birth and Death Process raised by Hudon in 1967 [9], this model exported Weibull Distribution depend on Mean Time between Failures (MTBF) [10].

From 1970s to early 1980s, the research of software reliability mostly determined in the assessment and selection of models, a few well-known models got used and improved. In this era, software reliability models had little uniqueness such as correctness verification-oriented, using the stochastic modeling technique, introducing statistical study technique to fault data, measuring software reliability by setting variables [11]. In 1972, Jelinski and Moranda planned the renowned Jelinski-Moranda model based on software failure rate, which appertains MTBF model and uses time dimension and maximum likelihood estimation [12-13]. Additional models projected later were mostly the improvements of J-M model by making the difficult hypothesis reasonable so that the model would be more practical [14]. In 1975, Littlewood set software reliability model designed at modular program [18], pointing that transmission and control among modules follow the Markov procedure and it could be consider a white-box model. In 1979,

Goel and Okumoto established a software fault model via plain Non homogeneous Poisson Process (NHPP) [19], and made some development to the earlier models relating software failure process [11]

Since 1990s, fast advancement has been made on the research of software reliability; research comprises software reliability plan, reliability testing and organization, collection of reliability facts, reliability prediction and reliability troubles of hardware- software-hybrid systems [10, 12]. Nowadays, software reliability modeling is still a hotspot.

### 2) *Engineering practice*

Software Reliability Models have come into observation as people try to understand the characteristics of how and why software fails, and try to calculate software reliability.

More than 200 models have been developed in view of the fact that the before 1970s, but how to measure software reliability still remains largely unsolved. No particular model completely represents software reliability

Despite the fact that the improvement position of engineering practice in software reliability is not so increase as theoretical research, it has made some development.

Besides, more and more applications are complete in production field concerning software reliability, particularly in aerospace field and automobile industry. So far, reliability and safety management method that faces the whole software life cycle systematically has been firstly formed at abroad.

In 2013, Johny et al. to assess quality, author advised the quality attributes must be taken into consideration and measured in the planning and design of the software[21].

In 2012, khan et al. developed a framework for measuring reliability of OOD at initial stage of development life cycle. Reliability assessment framework for OO design complexity viewpoint has been developed in this work. The planned framework associates the object oriented design property with complexity and in addition correlates complexity through reliability. He claimed no such framework has been presented in the literature that estimates software reliability of OO design by taking complexity into concern [22].

In 2011, Author made utilize of complexity measures, cohesion and coupling measures as the independent variable. The validation has shown a number of recognized metrics can be usefully employed for the evaluation of reliability [20]

In 2010, According to Quyoun Aasia, Software reliability is an essential component of software quality. It is self-motivated and stochastic. He classifies reliability into modeling, measurement and enhancement and then examined dissimilar modeling method and metrics for software reliability [23].

In 2009, Michael et al. through use of object oriented metrics to formulate quality assessment of the software product. They provide helpful comment to the management to keep the software process and product extra reliable [24].

In 2008, Leslie Cheung, Roshanak Roshandel, Nenad Medvidovic, proposed a framework for predicting reliability of software components at architectural design. Author identified reliability parameters and concludes the effects of reliability components. They mentioned the system to overcome the lack of failure by using defect analysis and categorization techniques, lack of operational outline information. [15].

In 2007, Claes Wholin introduces three traditions to estimate the parameters of the model. Author explained that factor can be evaluated by comparing historical data to earlier data. Parameters can be estimated using information from the present project. He described that by unifying three approaches given by Claes Wholin with two parameters given by G.O model, total number of failure is extremely dependent on recent project [16]

In 2006, Myron Hecht explained that for terrestrial and space elements software becomes a more significant cause of operational failures. So there is a growing want for collection of valid software failure data that can be correctly used to evaluate and improve dependability [17].

Figure 2. Shows facet of reliability models. Software reliability models comprise estimation and prediction of software [1].

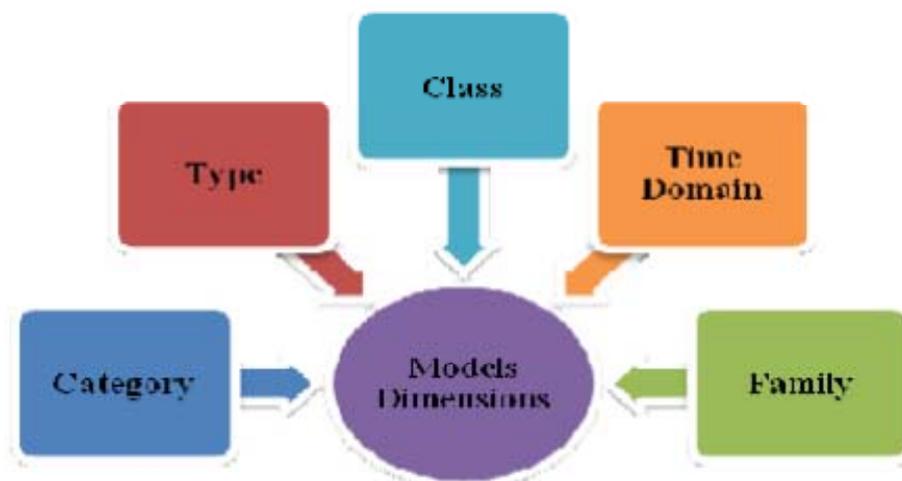


Figure 2: Dimensions of model these dimensions include the following kind of models characteristics as shown in table 1:

CHARACTERISTICS OF MODELS DIMENSION

Dimensions	Characteristics
Group	Finite & Infinite Failures
Kind	Poisson & Binomial
Class	Limited Failures
Time Domain	Execution time & Clock time
Family	Countless Failures

After a reassessment tour study found that a variety of methods or techniques are available in the literature for improving and estimating software reliability. A review of the reliability estimation of object oriented design shows that greatest effort center of attention at the later stage of software development life cycle. On the other hand, current developments in software design management repeatedly advocate integrating software reliability at design phase. This in turn will help the designers to improve quality and security of software and significantly reduce the overall development cost and rework of development life cycle for producing high quality maintainable, testable and reliable software.

## VII. CRITICAL OBSERVATIONS

Above successful accomplishment of the systematic literature review, a number of key explanations can be enumerated as follows.

Subsequent successful completion of the systematic literature review a number of important explanations can be enumerated as follows. Reliability is coupled with unpredicted failures of products or services and Analyzing, understanding why these failures occur is key to improving software reliability. The main reasons why failures take place include:

*The product is not robust for purpose or more especially the design is inherently incapable due to lack of analyzability.*

*Failures can be caused by wear-out*

- Failures might be caused by deviation.
- Wrong stipulation may basis failures.
- Misuse of the item may grounds failure.

## VIII. CONCLUSION

For any software industry, achieving software reliability is the central task. Attain Software reliability is tough for the reason that the complexity of the software be liable to be high. Reliability is a feature of superiority quality and software quality can be measured .So reliability measurement fully depends on those software characteristics that have positive impact on reliability quantification especially at initial phase of development cycle. So at each development phase, some quality attributes are applied and the reliability and quality of the software can be improved by applying software metrics at each of these software development stages.

## IX. CONTRIBUTION

The most important contribution of this study is in the field of object oriented design reliability improvement and measurement. Study has conducted a systematic review in this field. The unconnected reliability factors and measurement are recognized. Overall contribution is listed as follows:

- Conduct systematic literature review to collect details based on reliability improvement and estimation of object oriented software.
- A comprehensive step by step improvement of the systematic review process is described. It will help to further researchers as an orientation for undertaking SLR.
- Recognition of key papers related to the reliability improvement and evaluation.
- Finding of reliability factors and measurement in the current domain of OOD.
- Categorization and kind of different concepts about the software reliability in the present software engineering field.

## References

1. R.A Khan, K. Mustafa and S.I Ahson, "Operation Profile- A key Factor for Reliability Estimation", Universities press, Gautam Das & V P Gulati, CIT, 2004, pp 347-354.
2. M. Xie, Software Reliability Modelling, World Scientific Publishing Co. Ltd., 1991.
3. J. Peter Rooney, "Customer Satisfactions", In Proceedings of annual Reliability and Reliability Symposium, 24- 27 Jan. 1994 pp.376 – 381.

4. J. D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Bell Lab, IEEE, 1984, pp. 230-238.
5. M. L. Shooman, "Yes, Software Reliability Can be Measured and Predicted", Division of Computer Science, Polytechnic University, 1987, pp.121-122.
6. G. Junhong, Y. Xiaozong, L. Hsongwei, "Software Reliability Nonlinear Modeling and Its Fuzzy Evaluation", 4th WSEAS Int. Conf. on non-linear analysis, non-linear systems and chaos, Sofia, Bulgaria, October 27-29, 2005 , pp. 49-54.
7. N. F. Schneidewind, "Software Reliability Measurement", the R & M Engineering Journal, Vol. 23 No. 2, June 2003, pp. 1-10.
8. Abdullah, Dr. Reena Srivastava, and M. H. Khan."Modifiability: A Key Factor To Reliability", International Journal of Advanced Information Science and Technology, Vol.26, No.26, Pages 62-71 June 2014.
9. Hudon G R. Program Errors as a Birth and Death Process. Report SP- 3011, SsntaMonica, CA:System Development Corporation, 1967.
10. Yun Liu, Wei Zhao. Research and Progress in Software Reliability. Microcomputer Development, 2003(2):1-15.
11. Bangqing Qiu. Model Study and Overview of Software Reliability Abroad. Quality and Reliability, 1994(1): 14-18.
12. Z. Jelinski, P. Moranda. Software Reliability Research: Statistical Computer Performance Evaluation. N.Y. and London: Academic Press, 1972: 465-484.
13. Yong Cao. Software Reliability Model Based on Fractal and Mathematic Mechanization of Program Correctness Proof. University of Electronic Science and Technology of China, Doctoral Dissertation, 2010.
14. Ming Han. Software Reliability Models. Beijing University of Technology M.E.Dissertation, 2007.
15. W. Farr, "Software reliability modeling survey", Naval Surface Warfare Center, Technical Foundation, 1996, pp. 73-109.
16. L. Cheung, R. Roshandel, N. Medvidovic, L. Golubchik, "Early Prediction of Software Component Reliability", ACM, 2008, pp. 111-121.
17. M. R. Lyu, "Applying Reliability Model More Effectively", University of Iowa, Jet Prolusion Laboratory, Caltech, July 1992, pp. 43-52.
18. B. Littlewood. A Reliability model for systems with Markov structure. Applied Statistics, 1975, 24: 172-177.
19. A.L. Goel, K. Okumoto. Time-Dependent Error Detection Rate Model for Software and other Performance Measures. IEEE Transactions on Reliability, 1979, 28(3): 206-211.
20. Arti, Chhikara., R. S. Chhill and Sujata: " Prediction Of The Quality Of Software Product Using Object Oriented Metrics", International Journal Of Computer Engineering And Software Technology 2(1) Jan-June 2011; Pp. 1-6
21. JOHNY ANTONY P & HARSH DEV," ESTIMATING RELIABILITY OF SOFTWARE SYSTEM USING OBJECT-ORIENTED METRICS" International Journal of Computer Science Engineering and Information Technology Research (IJCEITR) Vol. 3, Issue 2, Jun 2013, 283-294
22. Yadava Amitabha and Khan R.A., (2012), "Reliability Estimation Framework – Complexity Perspective" retrieved from <http://airccj.org/CSCP/vol2/csit2509.pdf>.
23. Quyoum, Aasia., Din, Dar and Mehar, Ud Quadri S.M.K., "Improving software reliability using software engineering approach – A Review", International Journal of Computer Application, Vol. 10, No. 5, November2010.
24. Michael, English., Chris, Exton., Irene, Rigon, and Brendan Cleary" "Fault detection and prediction in an open source software project", In PROMISE '09: Proceedings of the 5th International Conference on Predictor Models in Software Engineering, pages 1{11, New York, NY, USA, 2009. ACM.

#### AUTHOR(S) PROFILE



**Abhinav Kumar Chand**, received the B.Tech. degree in Computer Science and Engineering from IIMT College of Engineering Greater Noida in 2010 and currently pursuing M.Tech From Goel Institute of Technology and Management, Lucknow, India.



**Namrata Dhanda**, is working as an Associate Professor and Head in the Department of Computer Science and Engineering/Information Technology at Goel Institute of Technology and Management, Lucknow. She has teaching experience of 13 years. Prior to her current assignment she has taught at Amity University, Lucknow and Babu Banarasi Das National Institute of Technology and Management, Lucknow at different positions. Her areas of interest include Automata theory, Database Management Systems and Algorithm design and Analysis.