International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.3 Issue No.12, pp : 1483-1487 01 Dec. 2014

Reliability Measurement of an Object Oriented Design: A Systematic Review

Nidhi Gupta, Dr. Rahul Kumar
Department of Computer Science & Engineering, R.K.G.I.T Ghaziabad, India
nidhi.gupta0707@gmail.com

Abstract: Reliability is one of the highly significant quality indicators of object oriented software. Its proper measurement or evaluation, constantly facilitate and improve the software development process. 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 feature in order to promise the quality software. A perfect measure of software quality fully depends on reliability measurement, and as an outcome estimating reliability is a complex problem attracting significant research consideration.

This review paper presents the results of a systematic literature review conducted to gather facts on software reliability estimation of object oriented design. In this review paper, our contribution is to discover the available recognized comprehensive and an absolute model or frameworks for measuring the reliability of object oriented design at an early phase of development life cycle.

***Keywords*— Software reliability, reliability measurement, Software metrics, Object Oriented Design, Software Quality.**

I. Introduction

In software development industry the steps towards corrective actions for victorious software development procedure comes too late resulting in uselessness, delayed delivery, in excess of budget and poor quality with reduced capabilities software. An early assessment towards software post-release quality can be a useful therapy to maximize the business result by shortening the time and increasing the probability of project success. The development group is also a beneficiary of the software quality estimation technique as they obtain an early recommendation regarding the quality of their product. Software quality estimation has been proved to be one of the most upcoming as well as interesting research topics of the decade which aims to recognize and minimize the error prone tasks to reduce the overall software development cost and time.

For the motivation that measurement is the key to achieving high reliability software, it is significant for software engineers to be familiar in this area. The software engineer would apply the body of information to improve the reliability of software throughout the development life cycle. In addition, the body of knowledge may be used as course of action for practitioners, licensing of software professionals, and used for training in software reliability measurement. Whenever we refer to "measurement", we will be referring to the measurement of the reliability of the software. Our motivation is that without measurement, software engineers would not be able to achieve reason that the complexity of software tends to be high. Whereas any system with a high degree of complexity, as well as software, will be hard to reach a certain level of reliability, system developers tend to move forward complexity into the high reliability software. Thus, design phase measurement is important to developing reliable software.

The assistance for software engineers of this process is to identify the facts and skills that are required to advance the measurement component of software engineering from an expertise to a profession. Rather than focus on the coding phase of the development process, as has been the case historically, it is important to identify how measurement can be applied to the initial stage and to key the necessary information to the process phases. This move toward is important for three reasons. First, early detection and resolution of reliability problems save considerable time and money in software development. Second, product and process measurements must be integrated so that the communication between the two can be assessed throughout the life cycle. Third, software engineers must have complete knowledge of the role of measurement in contributing to the development of high reliability products and the processes that produce them.

II. Approaches to Identifying Knowledge Requirements

There are two approaches to identifying the knowledge that is compulsory to plan and implement a software reliability program. One approach is issue oriented, as shown in Table 1. The other is life cycle phase oriented. The two approaches are compatible although different views of achieving the same objective and have been provided to show the software engineer why (issue oriented) and when (phase oriented) the need for measurement arises.

Issue Oriented

Issues happen because there are important considerations in achieving software reliability goals at acceptable cost. Using this approach, the associations among issues, functions, and knowledge requirements are shown in Table 1. This table shows some of the important functions that would be performed by the software engineer in executing a life cycle reliability management plan, oriented to the issues in the first column; this table is not exhaustive. Software reliability is a key feature to software quality. 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". Software reliability defines as the failure free operation of computer program in a specified environment for a specified time Software. Reliability is an essential attribute of software quality, in cooperation with functionality, usability, performance, serviceability, capability, install ability, maintainability, and documentation. Software Reliability is hard to attain, for the

software layer, with the speedy growth of system size and ease of doing so by advancement the software. Despite the fact that the analyzability of software is inversely correlated to software reliability, it is straightforwardly related to other important

factors in software quality, particularly software functionality and capability, etc. call attention to these features will be liable to include additional reliability to software.

III Reliability

According to IEEE, software reliability as "The capability of a system or component to perform its required functions under declared conditions for a specified period of time. The user based reliability of a program is described as the possibility that the program will give the correct output with a typical set of input data from the user background [23]. Software reliability is a branch of software quality. It relates to several areas where software quality is concerned. Therefore quantifying software reliability remains a complex problem as we don't have a high understanding of the character of software. Reliability is considered as the possibility that a system will not fail to perform its proposed functions over a specified time period. Customers are seriously aware of the reliability of software, they are likely to be mostly unworried with the point of the reusability of the components making up the source code.

Unreliability has a number of regrettable consequences and as a result for many products and services is a severe warning. For example low reliability can have inference for:

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

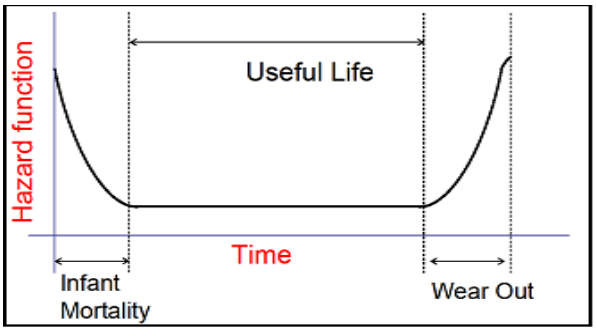

Fig.1 The bath-tub curve

Software Reliability R (t): The probability of failure-free operation of a computer program for a specified time under a specified environment.

Failure: The departure of program operation from user requirements.

Fault: A defect in a program that causes failure.

Failure Intensity (rate) f (t): The expected number of failures experienced in a given time interval.

Mean-Time-To-Failure (MTTF): Expected value of a failure interval.

Expected total failures m (t): The number of failures expected in a time period t.

Reliability Theory

Let "T" be a random variable representing the failure time or lifetime of a physical system.

For this system, the probability that it will fail by time "t" is:

The probability of the system surviving until time "t" is:

Failure rate - the probability that a failure will occur in the interval $[t_1, t_2]$ given that a failure has not occurred before time t_1. This is written as:

Hazard rate - limit of the failure rate as the length of the interval approaches zero.

A reliability goal expressed in terms of one reliability quantify can be easily transformed into another measure as follows (assuming an "average" failure rate, l, is measured).

IV OOD Metrics

As object oriented analysis and design techniques turn out to be broadly used, the demand on evaluate the quality of object-oriented designs significantly increases. In recent times, there has been a lot research effort to develop and empirically validate metrics for object oriented design quality characteristics especially Complexity, coupling and cohesion have received a significant interest in the field. regardless of the rich body of research and put into practice in developing design quality metrics, there has been less importance on dynamic metrics for object-oriented designs. The complex dynamic behaviour of a lot of real-time applications motivates a shift in interest from conventional static metrics to dynamic metrics.

Object-oriented technology is gaining considerable interest as a valuable paradigm for developing software applications. To estimate the quality of OO software, we need to judge and analyze its design and implementation using suitable metrics and assessment techniques. As a result, numerous quality metrics have been introduced to evaluate the application quality at various development phases. Assessing the application quality at an early development phase is helpful in guiding the development effort at succeeding phases. In this thesis, we are concerned with metrics that are appropriate at the design phase.

Table 1[23]

Name	Author	Metrics
MOOSE/CK	Chidamber et.al.	WMC, DIT, NOC, CBO, RFC, LCOM
MOOD	Abrreu et.al.	MIF, AIF, MHF, AHF, POF, COF
LK	Lorenz et.al.	CS, NOO, NOA, SI, OS, OC, NP
QMOOD	Bansiya	DSC,NOH,NSI,NMI, NNC,NAC,NLC,ADI,AWI,ANA,MFM,
LiW	Li et.al.	NAC, NLM,CMC,NDC,CTA,CTM
SATC	Rosenberg et.al.	CC, LOC,WMC,RFC,LCOM,DIT,NOC
STREW-J	Nagappan et.al.	NTC/SLC, NTC/NR, TLC/SLC,NA/SLC,NTC/NSC,NC,NLC/NC

V Research Methodology

A systematic literature review is a technique of recognizing, estimating and understanding the existing research result significant to a particular research area or subject [22]. 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 [22]. 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 [22].

1. Data source selection
2. Search strategy development
3. Search string formation
4. Study selection criteria identification
5. Study quality assessment identification
6. Study extraction strategy identification

VI Systematic Literature Review

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. in the midst of software reliability models, one can differentiate two main categories software reliability prediction model which address the reliability of the software early in the lifecycle and software reliability estimation model which evaluates current and future software reliability from failure data gathered beginning with the integration test of the software [32].

The following are concise examples of the functions and knowledge requirements programmed in Table 1. Our purpose is to give details to the software engineer the functions and the knowledge requirements that address the issues in Table 1. In addition, study describes techniques that can be employed to implement each of the functions. For example, with respect to Issue 1, we could interview key personnel and examine documentation in carrying out the function of "analyzing reliability goals and specifying reliability requirements".

Proposed study first; briefly describe the OOD method, mistake, error, and breakdown, the theory of Reliability, Software Reliability Model, and Software Metrics, followed by recognized Reliability Metrics. Leslie Cheung et al. planned a structure for predicting reliability of software components at architectural design. Author recognized reliability parameters and concludes the particular effects of reliability components. They talk about the mechanism to defeat the lack of failure by using defect investigation and categorization techniques, lack of operational profile information. [15].Claes Wholin introduces three ways to calculate the parameters of the model. Author described that p He described that by unifying three approaches given by Claes Wholin with two parameters given by G.O model, total number of failure is greatly dependent on current project [16]. Myron Hecht described that for terrestrial and space elements software becomes a more significant cause of operational failures. So there is an increasing need for group of valid software failure data that can be correctly used to evaluate and improve dependability [17].

Early estimation of reliability, exclusively at design phase assists the designers to improve their designs before the coding starts. A decision to change the design in order to improve reliability after coding has started may be very expensive and error-prone. While estimating reliability early in the development process may greatly reduce the overall cost [29].

VII Conclusions

A software reliability realization is necessary and inflexible to achieve it. It can be improved by suitable understanding of software reliability, characteristics of software and sound software design. Ensuring software reliability is no simple job. As rigid as the problematic is, promising progresses are still being made toward more reliable software.

After the above discussion our conclusion is that reliability is a quality factor that attempts to predict that how much effort will be required for software testing. After an exhaustive review process we found that reducing effort in measuring reliability of object oriented design is must in order to deliver quality software within time and budget.

VIII Critical Observations

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.

- Items are designed for a specific operating environment and if they are then used outside this environment then failure can occur.

ACKNOWLEDGEMENT

I am very much thankful to **Dr. Rahul Kumar Sir and Dr. M.H. Khan Sir** for their valuable suggestions and support.

REFERENCES

i. 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.

ii. Jintao zeng, Jinzhong Li, Xiaohui Zeng, Wenlang Luo "A Prototype System of Software Reliability Prediction and Estimation". IITSI 2010.

iii. J. Peter Rooney, "Customer Satisfactions", In Proceedings of annual Reliability and Maintainability Symposium, 24- 27 Jan. 1994 pp.376 – 381.

iv. J. D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", Bell Lab, IEEE, 1984, pp. 230-238.

v. M. L. Shooman, "Yes, Software Reliability Can be Measured and Predicted", Division of Computer Science, Polytechnic University, 1987, pp.121-122.

vi. G. Junhong, Y. Xiaozong, L. Hsongwei, "Software Reliability Nonlinear Modelling 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.

vii. N. F. Schneidewind, "Software Reliability Measurement", the R & M Engineering Journal, Vol. 23 No. 2, June 2003, pp. 1-10.

viii. S.R. Chidamber & C.H. Kemerer, "A Metrics Suite for Object Oriented Design", IEEE Transactions on Software Engineering, Vol.20, no. 6, June 1994.

ix. Briand, L.C., Wust, J., Daly, J.W. and porter, D.V., " Exploring the Relationships between Design Measures and Software Quality in Object Oriented Systems", The Journal of Systems and Software, Vol 51,pp. 245-273,2000.

x. Briand, L.C. Melo, W.L. and Wust, J., "Assessing the Applicability of Fault-Proneness Models Across Object-Oriented Software Projects", ISERN Report No. ISERN- 00-06, Version 2.

xi. EI-Emam, K.EI and Melo, W., "The Prediction of Faulty Classes Using Object-Oriented Design Metrics", National Research Council Canada, Nov. 1999.

xii. Musa, J.D., Iannino, A. & Okumoto, K., "Software Reliability: Measurement, Prediction, Application", Mc-Graw-Hill, New York, 1987.

xiii. Xie, M., "Software Reliability Modeling", World Scientific, Singapore 1991.

xiv. [13]Lyu, M.R. "Hand Book of Software Reliability Engineering", IEEE Computer Society press 1996.

xv. Hu, Q.P., Dai Y.S. Xie, M. and Ng S.H., "Early Software Reliability Prediction with Extended ANN Model", Proceeding of the 30thAnnualInternational Computer Conference COMPSAC, 2006.

xvi. W. Farr, "Software reliability modelling survey", Naval Surface Warfare Centre, Technical Foundation, 1996, pp. 73-109.

xvii. L. Cheung, R. Roshandel, N. Medvidovic, L. Golubchik, "Early Prediction of Software Component Reliability", ACM, 2008, pp. 111-121.

xviii. M. R. Lyu, "Applying Reliability Model More Effectively", University of Lowa, Jet Prolusion Laboratory, Caltech, July 1992, pp. 43-52.

xix. Shinji Inoue and Shigeru Yamada "Two–Dimensional Software Reliability Measurement Technologies'' IEEE 2009.

xx. Shinji Inoue "Generalized Discrete Software Reliability Modelling with Effect of Program Size" IEEE Trans on Systems, man, and cybernetics-Part A: Systems and humans, Vol, 37, No. 2, March 2007.

xxi. R. A. Khan, et. al., "An Empirical Validation of Object Oriented Design Quality Metrics", J. King Saud Univ., Vol. 19, Comp. & Info. Sci., pp. 1-16, Riyadh (1427H. /2007).

xxii. M. Xie, Software Reliability Modelling, World Scientific Publishing Co. Ltd., 1991.

xxiii. D. R. Jeske, and X. Zhang,"Some successful approaches to software reliability modelling in industry,''J. Syst.Softw., vol. 74, no. 1, 2005, pp.85-99.

xxiv. JOHNY ANTONY P & HARSH DEV , ESTIMATING RELIABILITY OF SOFTWARE SYSTEM USING OBJECT-ORIENTED METRICS, International Journal of Computer Science Engineering and Information Technology Research (IJCSEITR) ISSN 2249-6831 Vol. 3, Issue 2, Jun 2013, 283-294.

xxv. Purnaiah, Rama Krishna V and Bala Venkata Kishore "Fault removal efficiency in software reliability growth models" ISSN: 0975-

xxvi. 3275 & E-ISSN: 0975-9085, volume 4, issue 1, 2012, PP-74-77.

xxvii. Pardeep Ramachandran , Sarita Adve, Pradip Bose, Jude Rivers and Jayanth Srinivasan " Metrics for lifetime reliability " August 2006

xxviii. Vinay Tiwari1, Dr. R.K. Pandey2 International Journal of Advanced Research in Computer and Communication Engineering Vol. 1,Issue 10, December 2012 "open source software and reliability metrics"

xxix. Eduardo Valido-Cabrera, "software reliability methods" , August 2006

xxx. Ritika Wasen , P. Ahemed, M. Qasim Rafiq "New paradigm for software reliability estimation " Volume 44, No. 14, April 2012

Table 2: Software Reliability Evaluation

Issue	Function	Knowledge
1. **Goals:** What reliability goals are specified for the system?	**Analyze** reliability goals and **specify** reliability requirements.	• Reliability Engineering • Requirements Engineering
2. **Cost and risk**: What is the cost of achieving reliability goals and the risk of not doing so?	**Evaluate** economics and risk of reliability goals.	• Economic Analysis • Risk Analysis
3. **Context**: What application and organizational structure is the system and software to support?	**Analyze** the application environment.	• Systems Analysis • Software Design
4. **Operational profile**: What are the criticality and frequency of use of the software components?	**Analyze** the software atmosphere.	• Probability • Statistical Analysis
5. **Models**: What is the feasibility of creating or using an existing reliability model for assessment and prediction, and how can the model be validated?	**Model** reliability and **validate** the model.	• Probability Statistical Models
6. **Data requirements**: What data are needed to support reliability goals?	**Define** data type, phase, time, and frequency of collection.	• Data Analysis
7. **Types and granularity of measurements**: What measurement scales should be used, what level of detail is appropriate to meet a given goal, and what can be measured quantitatively, qualitatively, or judgmentally?	**Define** the statistical properties of the data.	• Measurement Theory
8. **Product and process test and evaluation**: How can product reliability measurements be fed back to improve process quality?	**Analyze** the relationship between product reliability and process stability.	• Inspection Test Methods
9. **Product Reliability and Process Quality Prediction**: What types of predictions should be made?	**Assess** and **predict** product reliability and process feature.	• Estimation Tools