Software Reliability – A Review

Alisha Garg[#1], Anil Arora[#2]

1Department of Computer Science
Gateway Institute of Engineering & Technology (GIET), Fazilpur, Sonepat
1alishagarg1291@gmail.com
2Department of Computer Science
Gateway Institute of Engineering & Technology (GIET),Fazilpur, Sonepat
2anilarora382@gmail.com

Abstract— Software reliability is about to define the stability or the life of software system with different properties. These properties include the trustfulness of software system, software cost, execution time, software stability etc. The aspects related to these software system includes the probability of software faults, frequency of fault occurrence, criticality of fault, associated module with respective fault etc. In a software development process, the pre estimation of software reliability is required to deliver the software product. According to the required level of software quality estimation of software cost, development time is also estimated. There are number of quality measure that approves the software reliability. Each stage of software life cycle itself takes some time quantum to deal with software reliability. Higher the software quality, lesser the software maintainability. This paper presents a review on the software reliability models.

Keywords— Software Reliability, Reliability Metrics, MTBF, MTTR.

I. Introduction

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].

Towards moving 21’s century, software becomes a driver for everything from elementary education to genetic engineering. Dependency & requirements on computer increases the difficulties & failures. Due to increase in dependency the size & complexity of system has grown. To avoid the failures & faults, reliability of software needs to be study during development of software so as to come up with reliable software.

There are several projects executed by NASA, & DOD that deal with highly complex software. Due to change in the three lines of code in a single program in 1991 the telephone system was collapsed in California and eastern parts [1]. Because of software failure, aircraft industry also faced lots of airliner crashes and abnormal flight conditions due to incompatible response to the pilots [2]. This paper presents a review on the software reliability models. The study throws the light on various dimensions of reliability models.

II. Software Reliability Model

Figure 1 shows dimension of reliability models. Software reliability models include estimation and prediction of software [1].

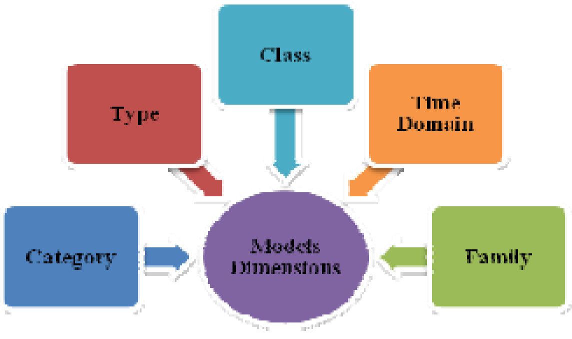

Figure 1: Dimensions of model

These dimensions include the following kind of models characteristics as shown in table 1:

Table 1: Characteristics of Models Dimension

Dimension	Characteristics
Category	Finite & Infinite Failures
Type	Poisson & Binomial
Class	Finite Failures
Time Domain	Execution Time & Clock
Family	Infinite Failure

In general, there are two major types of software reliability models: the deterministic and the probabilistic. The deterministic one is employed to study the number of distinct operators and operands in the program. The probabilistic one represents the failure occurrences and the fault removals as probabilistic events.

The probabilistic models can be further classified into different classes, such as error seeding, failure rate, and non-homogeneous Poisson process (NHPP). Among these classes, the NHPP models are the most popular ones. The reason is the NHPP model has ability to describe the software failure phenomenon. The first NHPP model, which strongly influences the development of many other models presented a NHPP model with S-shaped mean value function. These NHPP models are widely used, they impose certain restrictions or a priori assumptions about the nature of software faults and the stochastic behavior of software failure process.

III. SOFTWARE RELIABILITY METRICS

Software Reliability is defined as the ability of software to function under given environmental conditions for particular amount of time by taking into account all precisions of the software [6]. Software Reliability Testing is one of the testing field, which deals with checking the ability of software to function under given environmental conditions for particular amount of time by taking into account all precisions of the software. In Software Reliability Testing, the problems are discovered regarding the software design and functionality and the assurance is given that the system meets all requirements. Software Reliability is the probability that software will work properly in specified environment and for given time. Probability = Number of cases when we find failure / Total number of cases under consideration Using this formula, failure probability is calculated by testing a sample of all available input states. The set of all possible input states is called as input space. To find reliability of software, we need to find output space from given input space and software.

Some popular Reliability metrics are explained below:

Rate of Occurrence Of Failure (ROCOF): ROCOF measures the frequency of occurrence of failures .ROCOF measure of a software product can be obtained by observing the behaviour of a software product in operation over a specific time interval and then calculating the ROCOF value as the ratio of the total number of failures observed and the duration of observation.

Mean Time To Failure (MTTF): MTTF is the average time between two successive failures. To measure MTTF we can record the failure data for n failures. Let the failures occur at the time instants t_i, t_{i+1},t_2,.....,tn. Then MTTF can be calculated as

$$\sum_{i=1}^{n} \frac{t_{i+1} - t_i}{n-1}$$

Mean Time To Repair (MTTR): Once failure occurs, some time is required to fix the error. MTTR measures the average time it takes to track the error causing the failure and to fix them.

Mean Time Between Failure (MTBF): The MTTF and MTTR metrics can be combined to get MTBF metric:
MTBF = MTTF + MTTR.

Probability Of Failure On Demand: POFOD is the likelihood of the system failing when a service request is made.

Availability: Availability of a system is a measure of how likely would the system be available for use over a given period of time

IV. Literature Survey

Different papers provide different types of research work for software reliability which are explained below.

In 2009, Yi WAN *et al.* [7] proposed Software reliability analysis Markov model by Markov theory and mathematical statistics principle based on repairable debugging characteristic of software system. If all units that were presented in software repaired to new normal state after fault called reparable debugging characteristics of software in that case system behaviour was able to be described by homogeneous Markov Chain. Software reliability was analyzed by Markov theory. Effective measures based on Markov model were put forward, It was important to reduce software fault and improve operational quality of software. The mathematical statistics principle was combined with Laplace transformation as a result software reliability synthetically analysis model was built and evaluation characteristic parameters were obtained. Availability, reliability and mean time-to-failure were derived in that paper, many results were obtained by actual example analysis. It provided a theoretical basis and new method for reliability design and scientific management of software.

In 2010, Jianwen Chen *et al.* [8] proposed Bounded Monte Carlo Method. It was an improvement upon the basic Monte Carlo Method which can increase the computing accuracy with the help of lower and upper bounds. It was applied to the cases where the lower and upper bounds can be estimated easily before the basic Monte Carlo Method was invoked. This paper illustrated it in the two examples - computing the number π and computing event probability. These examples were shown that a more accurate result can be obtained with the help of lower and upper bounds. In the Bounded Monte Carlo Method, the basic Monte Carlo Method was repeated many times that increased the cost and time.

In 2010, Yashodhan Kanoria *et al.* [9] presented a new technique for statistical static timing analysis (SSTA) based on Markov chain Monte Carlo (MCMC), that allowed fast and accurate estimation of the right-hand tail of the delay distribution. Several modifications and enhancements were presented in this paper that enabled application of MCMC to SSTA. The new technique was also used for:

1. Yield calculation
2. Critical path identification
3. Shipped-product Quality Loss (SPQL) estimation

In 2011, A. Avritzer *et al.* [10] proposed an automatic test case generation approach by applied performability theory. Performability theory said that the requirements were specified in terms of a chosen performance metric. The expected response time and the number of tasks executed per time unit (throughput) were the two common metrics. Both the reliability modelling and test case generation approaches that were presented in this paper used the function p(n, t), the transient value of the probability of program P's correct execution, for input “n” and time “ t” . Two case studies were presented to illustrate their proposed approach. Proposed model-based test case generation approach was used to assess the reliability of large industrial mission-critical systems

In 2012, Thanh-Trung Pham *et al.* [11] presented an approach to predict the reliability of component-based systems. Reliability prediction methods for component-based systems used Markov models were often limited to a model of sequential executions. This approach relaxes these constraints by incorporating error propagation analysis and multiple execution models together consistently. It helps to improve the quality of the system in a cost-effective manner.

In 2013, Bo Zhou *et al.* [12] proposed a software random testing scheme based on Markov chain Monte Carlo (MCMC) method for addressing the problem associated with random testing. Random testing (RT) was one of the most classical software testing strategies. Although the RT was quite simple for implementation, it was often argued that RT was inefficient. A probability model was proposed to represent the activities for finding failures in software testing. In experiments, author compared effectiveness of MCMC random testing with both ordinary random testing and adaptive random testing in real program sources. These results provided that MCMC random testing was drastically improved the effectiveness of software testing.

In 2014, Madhavi Mane et. al. [13] performed a work, “Software Reliability and Quality Analyser with Quality Metric Analysis along with Software Reliability Growth Model”. This paper describes a novel software reliability growth model based on non homogeneous Poisson process with allowing for imperfect debugging. Maintaining and improving quality of the software is a very difficult task due to many factors like ambiguous requirement specification, lack of required resources etc. Many reliability growth models have been proposed until now according to different context and hence there is no globally accepted model. Software quality metric highlights the quality aspects of product, process, and project. As there is proportional relationship between quality and reliability, analyzing quality metrics is also a way to estimate reliability. So, they analyzed quality metrics along with maintaining the defect database.

In 2015, Arashdeep Kaur [14] performed a work, “Comparative Analysis of Reliability Models – Based on Uncertainty Factors”. This paper described that quality; schedule and cost are considered as important characteristics of software product. Reliability and availability are the measures considered for the software quality. Number of failures and Time between failures is considered as important uncertainty factors in the software reliability. But still many limitations are there in the recent approaches that do not consider these factors as an influential factor in terms of software reliability.

In 2015, Chahat Sharma et. al. [15] performed a work, “Perspective Approach of Software Reliability Models and Techniques”. Their paper is mainly divided into five sections: elucidate the evolution of various models and approaches of software reliability, illustrates the object oriented metrics used for estimation of software reliability, the review approach, the literature review and the review results along with certain merits and issues which form basis to bridge the gap between the current and the past research done on software reliability. It also discusses about the future work to stretch the breadth of the relevant literature in order to conduct more research on the extensively used reliability techniques in software industry.

V. CONCLUSION

Software is an essential component of many safety critical systems. These systems depend on the reliable operation of software components. Software reliability is about to define the stability or the life of software system with different properties. These properties include the trustfulness of software system, software cost, execution time, software stability etc. Software reliability is the probability of fault free operation of software components in a specified period of time in a specified environment. In this paper we studied number of studies have been conducted for measuring reliability of given software as a result a number of analytical models has been introduced.

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. R. Lyu, “Software Reliability modeling Survey”, *IEEE Computer Society Press*, AT&T Lab, 1996, pp 71-117.

[3] J. Peter Rooney, “Customer Satisfactions”, *In Proceedings of annual Reliability and Maintainability 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] N. F. Schneidewind, “Software Reliability Measurement”,*the R & M Engineering Journal*, Vol. 23 No. 2, June 2003, pp. 1-10.

[7] Yi Wan, Chengwen WU “*Software reliability model based on stochastic theory*” in 2009 IEEE.

[8] Jianwen Chen and Ling Feng “*Using Lower and Upper Bounds to Increase the Computing Accuracy*

of Monte Carlo Method" International Conference on Computational and Information Sciences 2010.

[9] Yashodhan Kanoria, Subhasish Mitra and Andrea Montanari "*Statistical Static Timing Analysis uses Markov Chain Monte Carlo*" in 2010 EDAA.

[10] A. Avritzer, E. de Souza e Silva, R.M.M. Leao and E.J. Weyuker "*Automated generation of test cases using a performability model*" journal Published in IET Software Received on 29th March 2011.

[11] Thanh-Trung Pham, Xavier Defago *"Reliability Prediction for Component-based Systems: Incorporating Error Propagation Analysis and Different Execution Models*" 12th International Conference on Quality Software 2012.

[12] Bo Zhou,Hiroyuki Okamura and Tadashi Dohi "*Enhancing Performance of Random Testing through Markov Chain Monte Carlo Methods*" IEEE Transactions on Computers, VOL. 62, NO. 1, January 2013.

[13] Madhavi Mane, Manjusha Joshi, Prof. Amol Kadam, Prof. Dr. S.D. Joshi, "Software Reliability and Quality Analyser with Quality Metric Analysis along with Software Reliability Growth Model", International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 3803-3806

[14] Arashdeep Kaur, "Comparative Analysis of Reliability Models – Based on Uncertainty Factors", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 2, February 2015

[15] Chahat Sharma, Sanjay Kumar Dubey, "Perspective Approach of Software Reliability Models and Techniques", ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 16, SEPTEMBER 2015