



A Study of Different Reliability Models Under Fault and Failure Analysis

Sangeeta*

Student, N.C.College of Engineering Israna
Panipat, Haryana, India

Sukhvir Singh

Associate Professor, N.C.College of Engineering Israna
Panipat, Haryana, India

Abstract— A software system is called the quality product under different attribute concerns. One of such main attribute to analyze the software quality is the reliability of the software. The reliability of the software system can be represented in different ways under different vectors. One of such vector is the software faults. In this paper, a discussion is been defined on the software reliability approaches as well as the exploration to the software fault analysis is defined in this work.

Keywords— Reliability, Software Fault Analysis, Software Quality.

I. INTRODUCTION

The two main concern of software engineers are: software quality and reliability. Software reliability means trustworthiness or dependability i.e. the ability of a program to perform a required function under given condition for a given period of time. In general, Software Engineers follow a systematic and organized approach for their work, as this is often the most effective way to produce high-quality software. However, engineering is all about selecting the most correct method for a given set of circumstances and a more innovative, informal approach to development may be effective in some circumstances Software reliability of a system can be measured in terms of chances of failure occurrence of the system. It is one of the major aspect of software quality that basically defines its contribution to the software quality estimation. Software reliability analysis is required at each phase of software development during the software plan. Based on the reliability vector, the associated test cases as well as associated testing time are evaluated. Software reliability can be evaluated under different software metrics[1,3,7].

IEEE 982.1-1988 defines software reliability as “The ability of a system or component to perform its required functions under stated conditions for a specified period of time.” The definition provided by IEEE is intentionally worded vaguely to allow organizations to define a level of reliability that is just right for each specific software project. In fact, the definition for software reliability is the same as the definition of hardware reliability so the reliability of a system can be determined from both software and hardware components even though software and hardware failures generally occur for different reasons[2,4].

Ensuring the reliability of a software project is important to all parties involved including Managers, Marketing, Programmers, and Customers. Unreliable systems can impact software developers and consumers by simply being an annoyance, by costing time and money, or worse case scenario, by costing single or multiple lives. Everybody involved in the software process has reasons for desiring reliable software. The major components used to define the software reliability are given as under[5]

A) Software Failure

A Software failure is defined as the situation in which a software program crashes and fails to provide the particular service. The failures are also identified in terms of their severity levels. A failure can give some data loss. Higher the chances of data loss, lesser the software reliability will be. The failures in a system can be expected or unexpected. These failure affect the software life and the software cost. The severity levels of the failure also vary from application to application[3].

B) Software Faults

A fault or the defect is basically identified when some internal error is detected from the system. It is also called a run time bug. For a software system, it is required to identify these software faults and remove them. A software can return incorrect results for the software system or the particular software module. If a software system not work as expected and behave abnormally, it means there is some fault in the software system. Sometimes the faults are also associated with the system and the environment. These kind of faults are because of some compatibility issues[4,11].

C) Errors

The errors are basically defined as some situation or the condition when the system is not working for a particular case in a particular module. The reason of error can be improper use of software system such as wrong input provided to the software system. These errors are basically occur if the user does not follow the associated constraints with the software systems[11].

D) Failure Functions

When a time basis is determined, failures can be expressed in several ways: the cumulative failure function, the failure intensity function, the failure rate function, and the mean-time-to-failure function. The *cumulative failure function* (also called the *mean-value function*) denotes the expected cumulative failures associated with each point of time. The *failure intensity function* represents the rate of change of the cumulative failure function. The *failure rate function* (or called the *rate of occurrence of failures*) is defined as the probability that a failure per unit time occurs in the interval $[t, t + Dt]$, given that a failure has not occurred before t [6,7].

II. LITERATURE REVIEW

Ravishankar. S has defined a work on the estimation of software cost. Author has defined the fuzzy based to system to improve the software quality and to estimate the software cost. The presented system provides more accurate estimation of software system. Author has tested the work on dataset provided by NASA as well performed the comparative analysis with COCOMO 1 and COCOMO 2. Eight membership functions available in fuzzy logic are used and a comparison is made to find out which membership function yields better result in terms of Mean Magnitude of Relative Error (MMRE) and PRED[1]. Another product specific fuzzy based analysis work is been presented by B.K. Mohanty[2]. Author has defined the decision support system for the internet products and the business. The author has defined the multiple attributes for analysis under the product specification with consumer preference to the system.

S. Malathi performed a work on the performance evaluation of the software system under the effort as well as the cost estimation. Author has provided a fuzzy based study to identify the complexity of the software system based on which the prediction of the software cost is also done by the author. Author identified the different reasons because of which software cost is affected. The improper understanding of software requirements has often resulted in inaccurate cost estimation. In analogy concept, there is deficiency in handling the datasets containing categorical variables though there are innumerable methods to estimate the cost[3]. H. Iranmanesh defined a work in direction of software risk evaluation under fuzzy logic. Author explore the risk evaluation methodology to prioritize the risks under different development stages for the IT projects. These stages includes the product design, integrated information system etc.[4].

Mikhail Glukhikh[5] has defined the work for the estimation of software reliability based on static error detection and based on code analysis. The static analysis is here defined as the development algorithm to calculate the software cost and the program reliability characteristics. The characteristics are the probability of successful program termination, the probability of the program is operable after execution of n statements, and mean number of executed statements before failure. Harpreet Singh has defined an application specific work for the evaluation of software complexity using fuzzy logic. Author has defined the analysis for the sensor network. The objective of the present paper is to propose a definition which could be used for internet and sensor networks applications. In this paper a new definition of software complexity is utilized for the proposed applications. It is suggested to use Binary Decision Diagrams for this purpose as these diagrams result in disjoint expression which is needed in the calculation of the software complexity. Simple cases of software complexity for series, parallel, series-parallel and non series-parallel networks are investigated[6].

Another fuzzy based work is defined by author to analyze the software risk. Author develops a fuzzy multi objective optimization model approach for selecting the optimal COTS software product among alternatives for each module in the development of modular software system. The problem is formulated for consensus recovery block fault tolerant scheme. In today's ever changing environment, it is arduous to estimate the precise cost and reliability of software[7]. A K Verma has defined a fuzzy logic based work to perform the defect rating for different software modules. The authors propose a revised model for defect rating that can be used for calculating group maturity within the organization. Fuzzy logic approach is used for the proposed model considering the linguistic or imprecise nature of the software measurements[8]. Another estimation of software cost under fuzzy logic is provided by F. JAVIER. CRESPO[8]. In this paper, fuzzy regression techniques based on fuzzification of input values are explored as an alternative to conventional regression for obtaining estimating equations. The available COCOMO-81 project database and the Furea fuzzy regression tool are used as a case study, emphasizing more realistic approaches to the expression of widely used cost driver values.

B. Praba consider uncertainty in the possibility measure (fuzzy possibility) and a method for finding fuzzy system reliability using fuzzy posfust reliability theory is demonstrated, where the system is modeled as a non homogeneous unified fuzzy possibilistic Markov model which consists of non homogeneous fuzzy possibilistic Markov models assigned in order together with transition fuzzy possibilities between them[9].

S. Mahmoud Taheri defined the analysis on statistical trends of software system. In this article Author review essential works on fuzzy estimation, fuzzy hypotheses testing, fuzzy regression, fuzzy Bayesian statistics, and some relevant fields[10].

Terry L. Hardy presents methods employing fuzzy logic concepts to assist in the decision-making process. In addition, this paper describes software developed at NASA Lewis Research Center for assisting in the decision-making process. Two diverse examples are used to illustrate the use of fuzzy logic in choosing an alternative among many options and objectives[11]. S. Sardar Donighi defined the fuzzy based reliability model for the software system. Fuzzy set based methods have been proved to be effective in handling many types of uncertainties in different fields, including reliability engineering. This paper presents a new approach on fuzzy reliability, based on the use of beta type distribution as membership function[12]. Byoung-Jun Park introduce the concept of the Rule-based fuzzy polynomial neural networks (RFPNN) as a hybrid modeling architecture and discuss its comprehensive design methodology. The

development of the RFPNN dwells on the technologies of Computational Intelligence (CI), namely fuzzy sets, neural networks, and genetic algorithms[13].

III. RELIABILITY MODELS

Software reliability models are basically divided in two major categorized based on reliability prediction. The First model can inhibit some specific features of software testing process and it is based on well-known S shaped Ohba model. This advanced model is applicable only for non-rare bug testing. For the rare bug rate prediction other model is proposed which is based on introduction of the additional control parameter last suspended time. He describes the advanced parametric models for assessment and prediction of software reliability that is to find the bugs in the software which is based on statistics of bugs at the initial stage of testing. The parametric model approach commonly associated with reliability issues which deal with the evaluation of the amount of bugs in the code.

Different models are developed for doing the reliability estimation are Duane Reliability Growth Model, Goel Model, Weibull Model, Classical S-shaped Model, Ohba S-shaped Model, etc. By taking some details of these models and by doing practical aspects of the software testing process a few Advanced Models were developed and implemented. The model which is proposed is sensitive to the situations typical for the early stages of Software development. This one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. In his research paper he describes the advanced parametric models for assessment and prediction of software reliability which is based on statistics of bugs at the initial stage of testing. The parametric model approach, commonly associated with reliability issues, deals with the evaluation of the amount of bugs in the code. The proposed models are sensitive to the situations typical for the early stages of Software development. As a result, one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. To support the optimization of such complex models, the Cross-Entropy Global Optimization Method is proposed. Some authentic numerical examples are considered to demonstrate the efficiency of the proposed models. The model which is proposed is sensitive to the situations typical for the early stages of Software development. This one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. In his research paper he describes the advanced parametric models for assessment and prediction of software reliability which is based on statistics of bugs at the initial stage of testing. The parametric model approach, commonly associated with reliability issues, deals with the evaluation of the amount of bugs in the code[4,7,8]. Some of these models are listed here

A) *Jelinski Moranda Model*

This model is one of basic model defined in 1972. The basic concept of this model is to estimate the software failure frequency analysis. Author has defined the software faults under the testing stage along with failure specifications during the testing phase. Author has defined the time analysis with each failure and the fault. Analysis is defined on two major factors called fault frequency and the failure interval.

B) *Littlewoods Model*

In the earlier model, no classification for the software fault is done but in this work the faults are categorized based on the defect size. According to this model, the larger faults are fixed earlier. The estimation of the system reliability is done based on average fault occurrence in the system.

C) *Goel Okumoto Imperfect Model*

The model is based on the failure estimation with the assumption of correct debugging of the system. The author has defined the defect analysis for the system. The presented model is imperfect to the limitation of hazard estimation for the specific model. The probabilistic evaluation of the software system is been defined under the fault analysis, debugging type and the failure rate.

IV. CONCLUSIONS

In this paper, the study of the software reliability is been done under different models. These models basically explore the software reliability under the fault and failure analysis. The presented paper has given the better view of different reliability models analyzed by different authors.

REFERENCES

- [1] Ravishankar. S, " Software Cost Estimation using Fuzzy Logic", International Conference on Recent Trends in Computational Methods, Communication and Controls (ICON3C 2012) Proceedings published in International Journal of Computer Applications
- [2] B.K. Mohanty, " Product classification in the Internet business—a fuzzy approach".
- [3] S. Malathi, " Performance Evaluation of Software Effort Estimation using Fuzzy Analogy based on Complexity", International Journal of Computer Applications (0975 – 8887)
- [4] H. Iranmanesh, " Risk Evaluation of Information Technology Projects Based on Fuzzy Analytic Hierarchal Process", World Academy of Science, Engineering and Technology
- [5] Mikhail Glukhikh, " Software Reliability Estimation Based on Static Error Detection".

- [6] Harpreet Singh," Software Complexity for Computer Communication and Sensor Networks Using Binary Decision Diagrams".
- [7] P. C. Jha," Fuzzy Approach for Selecting Optimal COTS Based Software Products Under Consensus Recovery Block Scheme", BVICAM's International Journal of Information Technology (BIJIT)
- [8] A K Verma," Fuzzy Logic Based Revised Defect Rating for Software Lifecycle Performance Prediction Using GMR", BVICAM'S International Journal Of Information Technology
- [9] F. JAVIER. CRESPO," On the Use of Fuzzy Regression in Parametric Software Estimation Models: Integrating Imprecision in COCOMO Cost Drivers".
- [10] B. Praba," Fuzzy Reliability of Non Homogeneous Unified Fuzzy Possibilistic Markov Model", European Journal of Scientific Research ISSN 1450-216X
- [11] S. Mahmoud Taheri," Trends in Fuzzy Statistics", AUSTRIAN JOURNAL OF STATISTICS
- [12] Terry L. Hardy," MULTI-OBJECTIVE DECISION-MAKING UNDER UNCERTAINTY: FUZZY LOGIC METHODS".
- [13] S. Sardar Donighi," A fuzzy reliability model for series-parallel systems", ISSN: 1735-5702