International Journal of Engineering and Techniques - Volume 4 Issue 3, May - June 2018

RESEARCH ARTICLE OPEN ACCESS

A New Reliability Growth Model to Estimate the Quality of Software

Gayathry G[1], ThirumalaiSelvi R[2]

[1](Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore, India
& Assistant Professor, Department of Computer Science, Mar Gregorios College, Chennai, India)

[2](Assistant Professor, Department of Computer Science, Govt. Arts College for Men, Nandanam, Chennai, India)

Abstract:

Software reliability is one of the significant factor in finalizing the quality of software. In recent years, Software Reliability Growth Models(SRGM) are used by number of software development organizations to assess and analyse the software product's quality. The estimation of reliability of software can save loss of time and cost. These models are applicable at the final stage of software development. In this paper, we propose a new model for assessing the software product's reliability. It first focuses on some of available reliability models and discuss about the problems associated with existing models and arrive at a new reliability model.

Keywords — **software reliability growth models(SRGM), hazard function, failure rate.**

I. INTRODUCTION

Due to rapid growth in technology computer applications became more diverse and spread in our day-to-day life. They are used in diverse areas for various applications including air traffic control, aircraft, industrial process control, and automotive mechanical and safety control. As the functionality of computer operations becomes more essential and yet more complicated and critical application increase in size and complexity, there is a great need for looking at ways to quantify and predict the reliability of such systems in various complex operating environments [1]. To produce a reliable software, it is necessary to measure and control its reliability. Software reliability is stochastic and dynamic. The exact value of product is never precisely known at any point of time.

From the commercial point of view, reliability is one of the most important characteristics of software quality. Reliability is defined as the probability of failure free operation of software for specified period of time in a specified environment [1]. Software reliability growth models(SRGM) are frequently used in the literature for reliability characterization of commercial software. SRGM is a prominent class of software reliability models(SRM).

SRM is a mathematical expression that specifies the general form of the software failure process as a function of factors such as fault introduction, fault removal and the operational environment [2]. Due to defect identification and removal the failure rate of a software system generally decreases over time. Software reliability modelling is done to estimate the failure rate in the form of curve by statistically estimating the parameters associated with the selected model. The purpose of this measure is twofold: 1) to estimate the extra test time required to meet a specified reliability objective and 2) to identify the expected reliability of software after release [2].

II. SOFTWARE RELIABILITY GROWTH MODELS

Software reliability models(SRM) can both assess and predict reliability. In reliability assessment SRM are fitted to the collected failure data using statistical techniques for example Linear, Non-Linear regression based on the nature of collected data. In reliability prediction, the total number of expected future failures is forecasted on the basis of fitted SRM. Both assessment and prediction need good data, which implies accuracy i.e. data is accurately recorded at the time the failures occurred and pertinence i.e. data relates to an environment that resembles to the environment for which the forecast is performed [3].

The 5 main steps in reliability modelling is listed below

- Keep a log of past failures
- Plot the failures
- Determine the model that best fits the observation
- Measure the accuracy of the model.
- Predict the future reliability in terms of predicting the total number of expected defects in the software system.

SRGM is one of the most widely used SRM models. They assume that reliability grows after a defect has been detected and fixed. SRGM can be applied to guide the test board in their decision of whether to stop or continue testing. SRGMs measure and model the failure process itself. Each model includes a time component, that is based on recording times t_i of successive failures i. The failure history is affected by a number of factors, including the environment within which the software is executed and how it is executed. A general assumption of these models is that software must be executed according to its operational profile i.e. test inputs are selected according to the probability of their occurrence during actual operation of the software in a given environment [4].

III. Existing Reliability Models

Some of the available models to estimate the reliability of software are discussed below:

A. Jelinski-Moranda Model

This is one of the most commonly used models for estimating software reliability. During testing N independent software faults that may cause failures are taken for consideration. During the debugging process no, new faults are introduced and the detected fault is removed in a negligible time. The hazard function also known as software failure rate is calculated during ti time between (i-1) th and ith failure is given by

$$Z(ti)= \phi\ [N-(i-1)]$$

Where ϕ is proportionality constant.

B. Goel-Okumoto Imperfect Debugging Model

Goel and Okumoto proposed an imperfect debugging model, which is the extension of Jelinski-Moranda model. In this model, the number of faults in the system at time t, X(t) is treated as a Markov process whose transaction probability is governed by the probability of imperfect debugging. Time between the transitions of X(t) is taken to be exponentially distributed with rates dependent on the current fault content of the system. The hazard function during the interval (i-1) th and ith failure is given by

$$Z(ti)= \phi\ [N-p(i-1)]\ \lambda$$

where ϕ is proportionality constant, N is the initial fault content of the system, P is the probability of imperfect debugging and λ is the failure rate per fault.

C. Musa Execution Time Model

In this model the reliability of software is analysed based on execution time. The hazard function for this model is given as

$$Z\ (t)= \phi f(N-nc)$$

t is the execution time, f is the linear execution frequency, ϕ is the proportionality constant and nc is the corrected number of faults.

IV. Problems faced by existing reliability models:

SRGMs are widely used to predict the reliability of software. Still there are some problems associated with these models.

- The arguments or parameters in these models are not suitable for all types of software.
- Introduction of new faults during fault removal process can lead to failure of software.
- It is difficult to generate proper test case.

- Time is considered as the main factor to find the failure rate and to estimate the reliability. Apart from time factor, some factors such as programmer efficiency, code complexity should be taken into consideration.

IV. PROPOSED NEW MODEL

In this paper, we propose a new model to estimate the reliability of software. A simple random failure-based model is developed for predicting the reliability of software.

Following assumptions are taken into account:

- Consider the failures caused by both hardware and software
- Software failures can be removed by Goel-Okumoto model and hardware faults can be removed by replacing faulty hardware.

Step 1:

Estimation of Failure Rate

Failure rate is estimated by adding the failures due to software and failures due to hardware.

The Equation for Failure Rate is given in equation (1)

$$\lambda (t) = \lambda_{hw} (t) + \lambda_{sw} (t) \qquad (1)$$

Substituting the Goel-Okumoto failure rate of software in the above equation (1) we get equation (2)

$$\lambda (t) = \lambda_{hw} (t) + abe^{-bt} \qquad (2)$$

Step 2:

Function for mean value calculation

The mean value function can be written as

$$m(t) = e^{-\lambda_{hw}(t)} + a(1-e^{-bt}) \qquad (3)$$

Step 3:

Estimation of Reliability

The software reliability R(x|t) is defined as the probability of failure free operation of a software for a specified time interval i.e. (t, t+x) in a specified environment. The resultant equation to calculate the reliability is given in equation (4).

$$R(x|t) = e^{-(\lambda_{hw} * x)} + e^{-a\,(e\text{-}bt - e\text{-}b(t+x))} \quad (4)$$

The reliability of software can be estimated using above said equation.

IV. CONCLUSION AND FUTURE WORK

Software reliability is a vital research area and it is a very significant part of software quality. In this paper, we generated a new software reliability growth model that incorporates both hardware and software failures. When compared to other traditional methods, this model provides us more accurate estimation for predicting the reliability of software.

The future work will involve validation of newly proposed model based on recent data sets.

REFERENCES

1. *J.D.Musa, A.Iannino, and K.Okumoto, Software Reliability, Measurement, Prediction and Application. McGraw-Hill,1987.*
2. *Lyu,M.(ed): Handbook of Software Reliability Engineering. New York: McGraw-Hill,1996.*
3. *Ohba, M.,1984. "Software reliability Analysismodel". IBMJ.Res. Develop. 28,428-443.*
4. *Chaurasia PK. "Classification of software Reliability models". IJARCSSE. 2014.Aug,4(8):1084-91.*
5. *Jyoti et al., "Use of Software Reliability Growth Models to Estimate the Reliability of Web Applications" in IJARCSSE, Vol.3, Issue 6, June 2013*
6. *Kwang Yoon Song et al.,"A Software Reliability Model with a Weibull Fault Detection Rate Function Subject to Operating Environments", Applied Science 2017,7,983.*
7. *Chatterjee, S.; Shulka, "A Software Reliability Modelling with different type of faults incorporating both imperfect debugging and change point". In proceedings of the 4th International Conference on Reliability, Infocom, Technologies and Optimization, Noida, Sep-2015.pp 1-5.*
8. *Pham, H." A Generalized fault-detection software reliability model subject to random operating environments". Vietnam, J.Comp.Sci.2016,3,145-150.*
9. *Song, K.Y; Chang, I.H.; Pham, H. "A Three-parameter fault-detection software reliability model with the Uncertainty of operating environments". J.Syst.Sci.Syst.Eng.2017,26,121-32.*
10. *Gayathry G, ThirumalaiSelvi R.," Classification of software reliability models to improve the reliability of software", IJST.Vol8(29), Nov.2015.*