EFFECTIVE TESTING FUNCTION FOR DEBUGGING IN SOFTWARE RELIABILITY GROWTH SYSTEMS USING SOFT COMPUTING NON-HOMOGENEOUS POISSON PROCESS MODEL

Anusha Merugu[1] Chaithanya Kadari[2]

[1,2]Assistant Professor, Dept of CSE, Sri Venkateswara Engineering College, Suryapet.

Abstract –

A great deal of significance has been appended to the testing period of the Software Development Life Cycle (SDLC). It is amid this stage it is checked whether the software item meets client prerequisites or not. Any errors that are distinguished are expelled. Be that as it may, testing should be observed to build its viability. Software Reliability Growth Models (SRGMs) that determine numerical connections between the disappointment phenomenon and time have demonstrated helpful. Our proposed strategy; build up a reliability model will give a technique to motivation behind testing-exertion functions for software reliability growth model dependent on non-homogeneous Poisson process (NHPP). In this proposed technique, we will consider the situation where the time-subordinate practices of testing-exertion consumptions are depicted by Weibull-Type Testing-Effort Function. In the proposed model contained four phases, to be specific: Initial Stage: In this stage, we will gather fundamental suspicion property for the model. From that point onward, we will get for the First stage, here we will introduce Modeling Software Reliability and Testing Effort functions and a similar time software discharge approach. Another blemished debugging discrete software reliability growth model with testing exertion. In the third stage: we will get an execution investigation model to reach out for the tradeoff examination as for goal level for the reliability of the SRGM. The prescient on cost, reliability, and power of the model has been dealt with two genuine disappointment datasets.

Keywords: *Software Reliability, Non-Homogeneous Poisson Process (NHPP), Reliability Factor.*

I. INTRODUCTION

With the fast progression in PC innovation and the snappy improvement of PC applications, the size of the software framework has upgraded and the function has turned out to be increasingly convoluted issues. Henceforth, the prerequisite of software quality is in extraordinary interest. By and large terms, software reliability can be characterized as the likelihood of disappointment free software task for a specific timeframe in a specific situation is a standout amongst the most critical quality measurements. Software reliability, just as the expense of software improvement, have risen as a significant standard of software quality... Progressively precise software reliability model are expected to figure or quantify the ideal discharge time and the expense of testing exertion with expanding request to give great software. Thus an extensive number of software reliability growth models dependent on NHPP is proposed to look at the reliability of software application at the season of testing stage [3]. The most proficient method for taking care of software advancement is to create software in characterized stages and to offer total functionality in different discharges. The advancement of software in numerous discharges offers various points of interest to the creating firms. The points of interest incorporate early income age, quick conveyances and upgraded showcase life of the item. [4].

Over recent decades, a few Software Reliability Growth Models (SRGMs) is proposed to evaluate software reliability dependent on genuine gathered disappointment information and it gives various quantitative data, for example, how to upgrade the process of software improvement and testing. Amid SDLC, the undertaking administrator needs to allot improvement and! or then again testing exertion in an orderly way to intensify the software reliability and to lessen potential disappointment punishments. Before, the scientists examined the connection between testing-exertion (TE) utilization just as reliability examination. It is likewise seen that TE can be typically signified as the quantity of CPU hours, the measure of expended labor, the quantity of executed experiments and so on [5].

Software reliability models are encouraging the enterprises regarding monetarily just as quantitatively. Software reliability growth models (SRGM) are characterized by a numerical model to speak to the continuous testing condition. For the most part software, reliability growth models are ordered as both logical and information-driven model. Logical models are delegated non-homogeneous toxin process models (NHPP), Bayesian and Markovian model. A non- homogeneous toxic substance process model goes up against the stochastic process to characterize the software disappointment phenomenon. The reliability growth models are ordered as an ideal and flawed debugging model. These models are proposed to foresee future disappointments.

II. Related Works:

Mohammad Ubaidullah Bokhari et at [15] have created software reliability growth models (SRGM) in light of the non-homogeneous Poisson process (NHPP) which consolidates the Burr Type XII testing-exertion functions (TEF). Various testing-exertion functions for modeling software reliability growth dependent on NHPP have been proposed in the previous decade. This paper demonstrates that the Burr Type XII testing exertion function can be communicated as the real testing-exertion utilization amid the software advancement process. Its blame expectation ability is assessed through the numerical investigations. SRGM parameters are evaluated by least square estimation (LSE) and greatest probability estimation (MLE) strategies and computational investigations performed on real software disappointment informational collection from different software ventures.

R. Peng et al in the paper [9] have examined the blame identification process (FDP) and blame redress process (FCP) with the reconciliation of testing exertion function and blemished debugging. So as to guarantee high reliability, it is indispensable for software to experience a testing stage, amid which flaws can be recognized and redressed by debuggers. The testing asset assignment at this stage, which is by and large outlined by the testing exertion function, altogether impacts the blame identification rate as well as an opportunity to address a recognized blame. In this paper, they have appeared at coordinate testing exertion function and blame presentation into FDP and afterward they have created FCP as deferred FDPwith a remedy exertion. Different explicit matched FDP and FCP models are accomplished dependent on various speculations of blame presentation and revision effort.B.Anniprincy et al in paper [23] have proposed two dimensional software reliability growth models by utilizing Cobb-Douglas generation function by consolidating the impact of testing time and testing inclusion on the quantity of shortcomings expelled in the software framework to catch the joined impact of testing time and testing inclusion. In this paper, they have built up a S-molded model with defective debugging and blame age to unravel the above issues happened amid the testing of software. The proposed model is approved on genuine informational indexes.

Liu Chang et al in the paper [25] have talked about how to consolidate both TEF and TCF into the customary NHPP software reliability modeling process to catch the coordinated impact of testing exertion just as testing inclusion on reliability estimation. To begin with, they have displayed another TEF, for example the arched S-shaped TEF(IS-TEF) for characterizing the S-formed differing pattern of the testing-exertion upgrading rate all the more precisely. In addition, they have proposed another NHPP SRGM (named IS-SRGM) which considers the IS-TEF. Then, a far reaching modeling structure for coordinating the TEF and TCF is proposed. Like this system, they have proposed another NHPP SRGM (named IS-LO- SRGM) with both the IS-TEF and calculated TCF (LO-TCF).

Bijoyeta Roy et al in the paper [26] have displayed a general structure of two non-homogeneous Poisson processes based SRGM models, for example, Goel Okumoto and Delayed S-Shaped model. The primary goal of these two models is to process the shortcomings or disappointments staying in the framework. They have portrayed two classes of diagnostic models alongside their hidden suspicions.

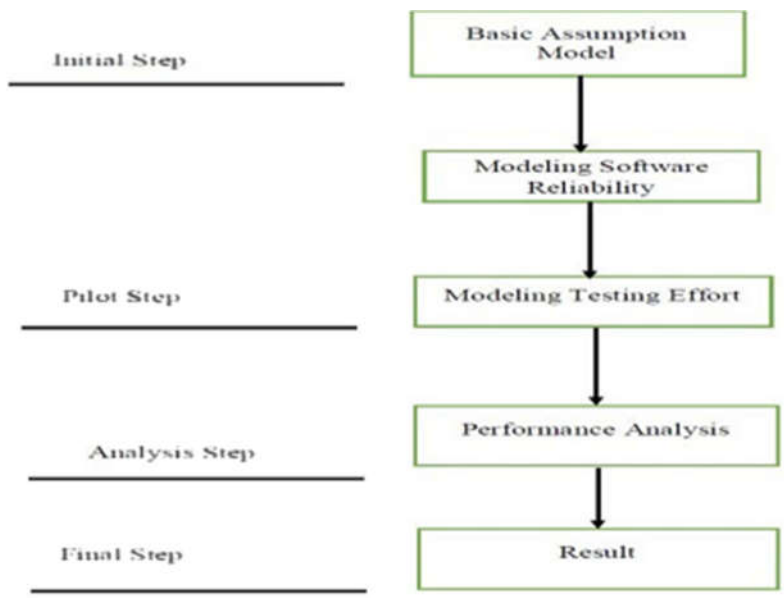
III. Proposal work

In this paper, we build up a SRGM with the extra suppositions that:

I. Software flaws are of two kinds, specifically, Type-I and Type-II which are of various seriousness. The testing exertion required relies upon the seriousness of shortcomings.

2. The blame evacuation rate is time subordinate concerning testing exertion. This can represent realizing which increments with testing time.

Amid the testing stage, the blame introduction rate unequivocally relies upon the aptitude of test groups, program size, and software testability. Accordingly in this paper, we treat the blame evacuation rate as a function of time and testing exertion. This supposition makes the model progressively adaptable and it can catch an assortment of software reliability growth bends. As talked about, the proposed SRGM fuses both time slack and the idea of the testing exertion. Amid testing, assets, for example, labor and time (PC time) are devoured. The disappointment (blame ID) and expulsion are reliant upon the nature and measure of assets spent.



The time-subordinate conduct of the testing exertion has been examined before. The exponential and Rayleigh bends are utilized to depict the connection between the testing exertion utilization and testing time (the timetable time). An exponential bend is utilized if the testing assets are consistently overwhelmed by regard to the testing time and Rayleigh bend generally. Strategic and Weibull type functions have likewise been utilized to portray the testing exertion. The SRGM with testing exertion proposed in this paper depends on Non-Homogeneous Poisson Process (NHPP) supposition. The model has been approved on three software blame distinguishing proof informational indexes.

3.1 Estimation of Testing Effort Functions

A Software Reliability Growth Model (SRGM) clarifies the time-subordinate conduct of blame evacuation. A few SRGMs have been proposed in software reliability writing under various arrangements of presumptions and testing condition, yet more are being proposed. The proposed SRGM in this paper considers the time-subordinate variety in the testing exertion. The testing exertion (assets) that oversee the pace of testing for practically all the software ventures are Manpower which includes,l. Disappointment distinguishing proof personnel.2.Failure revision personnel.3.Computer time.

The key function of labor occupied with software testing is to run test cases and contrast the test outcomes and wanted particulars. Any takeoff from the particulars is named as a disappointment. On a disappointment, the blame causing it is recognized and after that expelled by disappointment adjustment work force. The PC offices speak to PC time, which is important for disappointment ID and rectification.

To portray the conduct of testing exertion, either Exponential, Rayleigh, Logistic or Weibull function has been utilized. Exponential and Rayleigh models can be gotten from the accompanying differential condition:

$$\frac{dX(t)}{dt} = c(t)[\alpha - X(t)] \quad \text{-------------(1)}$$

where e(/) is the time dependent rate at which testing resources are consumed, with respect
to the remaining available resources.

Solving Eq. illunder the initial condition X(O) = ° we get

$X(t) = \propto \left[1 - exp \int_0^1 C(k)\, dk\right]$-----(2)

When e(/) = P a constant

$X(t) = \propto \left[1 - e^{-\beta t}\right]$-----(3)

If e(t) = P . I illgives a Rayleigh type curve

$X(t) = \propto \left[1 - e^{-\int \beta t}\right]$-----(4)

In this paper, we are building up a SRGM, in light of NHPP with strategic testing exertion function. SRGM with calculated testing exertion function gives better outcome on some disappointment informational collections. The combined testing exertion expended in the interim (0, I] has the accompanying structure:

$X(t) = \frac{\alpha}{1+\beta e^{-et}}$-----(5)

Where a, 6 and e are constants.

Increasingly adaptable and general testing exertion function can be acquired utilizing Weibull function and the aggregate testing exertion devoured in the interim (0, I] has the accompanying structure:

$X(t) = \propto [1 - e^{-}\beta t y]$-----(4)

Where a, 6 and e are constants. In the following area, we talk about the Software Reliability Growth Modeling utilizing one of these testing exertion functions.

IV. Software Reliability Growth Modeling

The impact of testing exertion has been incorporated into some SRGMs. The SRGM with testing exertion created in this paper considers the time slack between the disappointment and blame disengagement/evacuation processes. The model has the accompanying express suppositions: l. Disappointment perception/blame evacuation phenomenon is modeled by NHPP.

2. The software is liable to disappointments amid execution brought about by deficiencies staying in the software.

3. Each time a disappointment is watched, a prompt (deferred) exertion happens to choose the reason for the disappointment and to expel it. The time delay between the disappointment perception and its consequent evacuation is expected to speak to the seriousness of shortcomings.

4.The increasingly extreme the blame, more the time delay. Amid the blame confinement/expulsion, no new blame is brought into the framework.

5. The blame expulsion rate is a strategic learning function as it is normal the learning process will develop with time.

6. The disappointment rate of the software is similarly influenced by flaws staying In the software.7. The blame detachment/evacuation rate as for testing exertion force is corresponding to the quantity of watched disappointments whose reason are yet to be recognized.

4.1 Modeling of S-Shaped Model with Imperfect Debugging:

Assuming D1 and D2 to be Type-I and Type-II faults initially present in the software, respectively. Where

a. Modeling S-Shaped Model

It is assumed that S-Shaped Model are simple faults which can be removed instantly as soon as they are observed. Hence S-Shaped Model are modeled as a one-stage process:

$$\frac{\frac{d}{dt}m_1(t)}{x(t)} = b_1\left(a_1 - m_{lr}(t)\right)$$----------(7)

The one stage process as modeled in Eq. illdescribes the failure observation, fault isolation and fault removal processes.. Solving the differential equation illunder the boundary condition mlr(t = 0) = O. We get

$$m_1r(t) = a_1\left(1 - e_{b_1}^{X(t)}\right)$$---------(8)

Modeling C-Shape Model:

It is expected that the Type-II flaws expend additionally testing exertion when contrasted and Type-I blames. This implies the testing group should invest more energy to break down the reason for the disappointment and thusly requires more noteworthy endeavors to expel them. Henceforth the evacuation process for such blames is modeled as a two-organize process:

$$\frac{\frac{d}{dt}m_2(t)}{x(t)} = b_2\left(a_{12} - m_{2f}(t)\right)$$----------(7)

$$\frac{\frac{d}{dt}m_2(t)}{x(t)} = b_2(t)\left(m_{2r}(t) - m_{2f}(t)\right)$$-----------(8)

Where

$$b_2(t) = \frac{b_2}{1 + \eta e^{-b_{2X(i)}}}$$

The principal phase of the two-arrange process is given by Eq. m. This stage depicts the disappointment perception process. The second phase of the two-arrange process given by Eq. (10) depicts the deferred blame expulsion process. Amid this stage, the blame evacuation rate is thought to be time subordinate. The purpose behind this presumption is to consolidate the impact of learning on the evacuation work force.

b. Modeling total fault removal phenomenon

The proposed model is the superposition of the two NHPP with mean esteem functions given in Eq.[[l and ill}, Thus, the mean esteem function of superposed NHPP is

$m(t) = m_1 r(t) + m_2 r(t)$---------------(11)

From Eq.111l, the fault removal rate per fault for C-shape can be derived as follows:

$b_2(t) = \frac{\frac{d}{dt} m_{2r}(t)}{a_2 + m_{2r}(t)} = \frac{b_2(1+\eta e^{-b_2 X(t)}) - b_2(1+\eta e^{-b_2 X(t)})}{(1+\eta e^{-b_2 X(t)})(1+\eta + b_2 X(t))}$------(12)

Note that b2(t) increments monotonically with time t and keep an eye on steady b2 as t - > 00. In this manner, in the consistent state, C- shape growth bend carries on like the S-shape growth bend and thus there is no loss of all inclusive statement in accepting the enduring state rate b2 to be equivalent to bj.

c. Estimation ofparameters

Technique for least squares and most extreme probability strategy can both be utilized for the estimation of parameters. The parameters of testing exertion function against disappointment time can be evaluated first, utilizing the exertion information. The function giving the best fit should be utilized for evaluating parameters of SRGM. The testing exertion information are given through testing exertion xk(xj < X2 < ... < xn) expended in time (0, oj J = I, 2, ... , n. The testing exertion model parameters an and fJ can be evaluated by the strategy for least squares as pursues

$Minimize\ \sum_{j=1}^{n} [X_j - \check{X}]$-------(13)

Subject to Xn= Xn (for example the assessed benefit of testing exertion is equivalent to the genuine esteem) Once the appraisals of an and fJ are known, the parameters(10) of the SRGM for the modules can be evaluated through most extreme probability estimation strategy utilizing the basic Stochastic Process, which is depicted by a Non-Homogeneous Poisson Process. Amid estimation, evaluated estimations of a. what's more, fJ are kept settled. On the off chance that the blame expulsion information for a module is given as a combined number of flaws expelled yj in time (0, lj]. The probability function for that module is given as the shortcomings and gives the reliability parameters as yield as appeared in the figure bel

$L\left(a, b, \frac{r_1}{y}, W\right) = \prod_{j=1}^{n} \frac{[m_r(\check{X}_j) - m_r(\check{X}_{j-1})]}{(y_j - y_{j-1})}$------(14)

The estimated values of G, band 8 maximizing the above expression.

Conclusion:

In this paper, another SRGM dependent on NHPP has been proposed. In this paper, we have built up a general methodology in determining progressively broad models dependent on basic suspicions, consistent with the fundamental software reliability growth modeling dependent on NHPP. The proposed models embed a more extensive hypothetical structure which represents collaboration between various components of software reliability measurements. The new SRGM is adaptable in nature as it can portray both exponential and S-formed bends as exhibited in this paper. This model is as a rule additionally adjusted to incorporate the impacts of flawed debugging/mistake age. It is likewise being connected to essential administration basic leadership circumstances like discharge time issue. We likewise infer that the proposed SRGM has better execution when contrasted with the other SRGM and gives a sensible prescient capacity for the genuine software disappointment information. Accordingly, this model can be connected to a wide scope of software.

References:

1. Kapur, P. K., Pham, H., Anand, S., & Yadav, K. (2011). A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. Reliability, IEEE Transactions on, 60(1), 331-340.
2. Li, Tao, and Kaigui Wu. "A NHPP Software Reliability Growth Model Considering Learning Process and Number of Residual Faults." Journal of Convergence Information Technology 7.13 (2012): 127-134.
3. Fang LI, Changze WV, Kaigui WV, Wenke ZHANG, Jie XU, "Incorporating Variable Fault Detection Rate and Fault Removal Rate into Software Reliability Growth Model", Journal of Computational Information Systems 10: 3 (2014) 1123-1131.
4. Kapur, P. K., Pham, H., Aggarwal, A. G., & Kaur, G. (2012). Two dimensional multi-release software reliability modeling and optimal release planning. Reliability, IEEE Transactions on, 61(3), 758-768.
5. Ke, S. Z., Huang, C. Y., & Peng, K. L. (2014, November). Software reliability analysis considering the variation of testing-effort and change-point. In Proceedings of the International Workshop on Innovative Software Development Methodologies and Practices (pp. 30-39). ACM.
6. Sharma, Ashish, and Dharmender Singh Kushwaha. "Applying Requirement based complexity for the estimation of Software Development and Testing Effort." ACM SIGSOFT Software Engineering Notes 37.1 (2012): 1-11.
7. Rafi, S. M., & Akthar, S. (2012, September). Incorporating fault dependent correction delay in SRGM with testing effort and release policy analysis. In Software Engineering (CONSEG), 2012 CSI Sixth International Conference on (pp. 1-6). IEEE.
8. Mir, Khurshid Ahmad. "A Software Reliability Growth Model." Journal of Modern Mathematics and Statistics 5.1 (2011): 13-16.
9. Latha Shanmugam, and Dr. Lilly Florence, "An Overview of Software Reliability Models", Volume 2, Issue 10, October 2012 ISSN: 2277 128X.
10. Mane, M., Joshi, M., Kadam, A., & Joshi, S. D. (2014). Software Reliability and Quality Analyser with Quality Metric Analysis Along With Software Reliability Growth Model. International Journal of Computer Science & Information Technologies, 5(3).
11. Bokhari, Mohammad Ubaidullah, and Nesar Ahmad. "Incorporating Burr Type XII Testing- efforts into Software Reliability Growth Modeling and Actual Data Analysis with Applications." Journal of Software 9.6 (2014): 1389-1400.
12. Peng, R., Li, Y. F., Zhang, W. J., & Hu, Q. P. (2014). Testing effort dependent software reliability model for imperfect debugging process considering both detection and correction. Reliability Engineering & System Safety, 126, 37-43.
13. Anniprincy, B., and S. Sridhar. "Two Dimensional Software Reliability Growth Models Using Cobb-Douglas Production Function and Yamada S-Shaped Model." J. Softw. Eng. Simul 2.2 (2014): 1-11.

14. Liu Chang, Liu Yuan2 Ren Zhanyong, and Li Haifeng, "Software Reliability Modelling Considering both Testing Effort and Testing Coverage", International Symposium on Computers & Informatics (ISCI 2015).
15. Roy, B., Misra, S. K., Basak, A., Roy, A., & Hazra, D. (2014). A quantitative analysis of NHPP based software reliability growth models. International Journal of Innovative Research in Computer and Communication Engineering, 2(1), 2338-2432.
16. S. Yamada, M. Ohba, S. Osaki, (1984), S-shaped software reliability growth modelling for software error detection, IEEE Trans. Reliab., 32 (5), 475-484.
17. H. Yamada, H. Ohtera, H. Narihisa, (1886), Software reliability growth models with testing effort IEEE Trans. Reliab., 35 (1), 19-23.