International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 11 (2017) pp. 2703-2706

A Novel NHPSRGM Tool for Enhancing the Quality of the Testing Phase

Chandra Mouli Venkata Srinivas Akana
Research Scholar, Dept. of CSE, JNTUK, Kakinada, Andhra Pradesh, India,

ORCID: 0000-0001-8636-0522

Dr. C. Divakar
Principal, Avanthi's Research and Technological Academy, Vijayanagaram, Andhra Pradesh, India.

Dr. Ch. Satyanarayana
Professor, Dept of CSE, & Director, Academic and Planning, JNTUK, Kakinada, Andhra Pradesh, India.

Abstract

The testing phase of the SDLC is responsible for the validation of faults. An efficient Software Reliability Growth Model (SRGM) can enhance or improve the reliability. This paper proposes a novel approach for the SRGM by adapting the non-homogeneous poisson process enhancement (NHPPE) concept. It serves as a generalized framework for SRG modelling based on Testing coverage (TC) or Testing effort (TE) and also includes the change-point concept. The proposed method is evaluated on KC_2 NASA Dataset and the result indicates the efficiency of the proposed method.

Keywords: Fault, Validation, Number of Arrivals, Evaluation, Change-Point, NHPPE, NHPSRGM, Testing Coverage, Testing effort.

INTRODUCTION

The field testing of software draws a lot of attention from researchers because of the importance of the reliability of software. Releasing quality software products to the end-users after a thorough software testing or software assessment is very important. To assess the software reliability, many developers have proposed many SRGMs. Reliability is validated by measuring the probability of the software to conduct a failure-free operation over a given time interval and within a specified environment [1]. Nowadays industries need reliable software, particularly in applications that are mission-critical and where safety is paramount like satellite and shuttle control, defence systems, transportation, hospitals, banking sectors etc. and hence the need for reliability. To evaluate a software system's performance, a reliability prediction for that software is needed. The reliability of a software product is quantified based on the residual faults that remain in the software system. The primary goal of the testing phase is on eliminating or reducing the residual faults; thereby enhancing the quality of the software.

Most SRGMs developed are based on the testing time (CPU time or calendar time) and the number of faults identified [2]. All these SRGMs make an assumption that all the residual faults identified within the software system are of the same type. But in the real world, it is not so [3].

The software may contain different types of errors or faults. Different errors need to be tackled in different ways and also need a varying amount of testing effort (or testing coverage) for debugging. A test is said to be successful when it identifies the existence of the latent faults. The procedure involved in identifying the faults and resolving them is called debugging. Testing coverage helps the software developers a lot in assessing the quality of the software and it also aids in calculating the amount of effort required to improve the software's reliability [4].

Here, we present a framework which is very helpful in SRG modelling with respect to testing coverage (TC) or testing effort expenditure.

Different types of faults need different amount of testing effort. Testing has three phases viz., Errors Observation, Errors Isolation and Errors Removal.

The SRGM tool is developed by considering the concept of change-point [11]. Whenever software developers try to eliminate the identified errors they face two types of cases where:

Case 1: Identified errors are removed successfully and the elimination of the identified errors does not introduce any new errors.

Case 2: Identified errors are removed successfully and new errors are introduced because of the elimination of the identified errors. Here, the number of faults keep changing and they are never fixed. This can be considered as a perfect example of the change-point case.

Through this concept we can provide reliable and quality software to the end-users with quantitative confidence.

Initially, Zhao [2] introduced the change-point concept and Huang et.al [3] developed SRGMs incorporating Testing effort functions and change-points. Shyur Kapur et.al [5, 6] introduced imperfect debugging with a change-point concept. Yamada presented the process of Yamada time dependent behaviour of testing coverage (TC). There are a variety of SRGMs based on the Non-Homogeneous Poisson process (NHPP) [7, 8]. Goel-Okumoto introduced a concept of fault detection and removal rate which increases with respect to time [9]. Flexible models like S-shaped and exponential growth curves are called Delayed S-shaped model of Yamada [12] and exponential models of goal respectively. Ever since the advent of SRGMs, several SRGMs have been proposed by researchers and it is really an arduous task to make a selection from so many available SRGMs. To overcome this, unified modelling approaches were introduced by many researchers. One of the unified methodologies is that of Fault detection and Fault correction process [10]. So, the present paper proposes a novel approach for designing SRGM tool that relies on non-homogeneous poisson process enhancement (NHPPE) concept. It serves as a generalized framework for SRG modelling with respect to Testing coverage (TC) or Testing effort (TE) and also includes the change-point concept.

METHODOLOGY

A non-homogeneous poisson process enhancement (NHPPE) concept is found to be efficient for designing the SRGM tool. So, the present paper proposes a non-homogeneous Poisson based SRGM tool (NHPSRGM) for enhancing the software reliability. The proposed NHPSRGM consists of the following features:

- The bug-identification or bug-removal procedures are designed based on the Non-Homogeneous Poisson Process enhancement.
- For every fault occurrence, a debugging effort will be kick-started for handling it.
- During the fault-removal phase, it is ascertained whether fault removal is successful or not. Along with this, checks are made about generation of new faults.

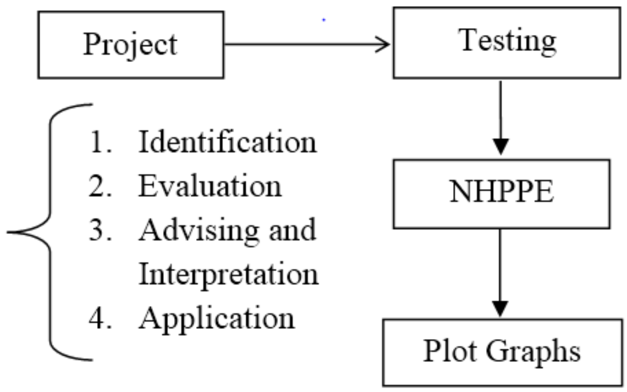

Figure 1. Structure of the proposed NHPSRGM tool.

The various steps in the proposed model:

Identification:

It computes the following features.

- ***TNOC:*** Counts the total number of attributes/objectives in a class.
- ***WMPC:*** Whole methods per class are measured by this software [11].
- ***NOCP:*** Total class count in a package.
- ***SIZE:*** Total number of lines in a code.

Evaluation:

During this phase, various measures are calculated based on the source code.

Interpretation:

The evaluated measures are used as input for the interpretation part of the proposed NHPSRGM tool.

In the present paper, the interpreted values and threshold values of the corresponding metric values are compared with the benchmarks. While doing this, any one of the following two cases may occur:

Case 1:

Metric values extracted lie below the corresponding metric threshold value. In this case, we don’t have any occurrence of the issue which is observed.

Case 2:

Metric values extracted lie above the corresponding metric threshold value. In this case we have an occurrence of the issue which is observed.

Non-homogeneous Poisson Process Enhancement Technique:

It is used as a framework for the finite failure software. In this framework, the expected number of 'detected failures' are identified in a given time interval 't'. The Mean Value Function is expressed as:

$$MVF = a \times b(t) \qquad \text{... (1)}$$

where,

a represents Number of short comings,

b(t) = Capacity of scope.

The expected Number of Failures (NF) is estimated using the equation:

$$NF = TNL \times FD \qquad \text{... (2)}$$

where,

TNL represents Total number of lines of code,

FD represents the Fault density.

The production function's mathematical form is given by Equation (3).

$$X = B \times K^a \times L^b \qquad \text{... (3)}$$

where,

X represents total production,

B represents Total factor productivity,

K represents the Input of labour,

L represents Input of capital,

a represents the capital elasticity constant,

b represents the labour elasticity constant.

The rate function R(t) varies from time to time. Number of events between t_1 and t_2 is given by Equation (4).

$$R_{t_1,t_2} = \int_{t_1}^{t_2} R(t)dt \qquad \text{... (4)}$$

During the time interval t_1 to t_2, estimation of the number of arrivals (NoA) is given by Equation (5).

$$NoA = N(t_2) - N(t_1) \qquad \text{... (5)}$$

RESULTS AND DISCUSSIONS

The present paper proposes a novel approach (NHPSRGM) for the design of an efficient SRGM based on NHPEE. It considers a repairable system whose failure-data is presented below in Table 1.

Table 1: Failure-Data for the proposed NHPSRGM.

S. No.	Failure time	Cumulative number of failures
1	6.70	8
2	8.50	9
3	11.20	10
4	16.20	11
5	17.10	12
6	19.00	13
7	22.50	14

The proposed NHPSRGM tool takes the input data given in Table 1 above and the plot for 'cumulative number of failures vs. time' is illustrated below (Figure 2).

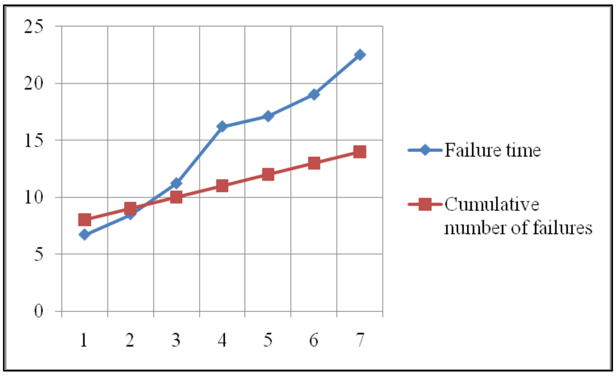

Figure 2. Plot of Cumulative number of Failures vs. Time.

The proposed model provides the ability to implement an individual NHPP model on each segment. In the present tool, the change-point connects the two models, and therefore they cannot be considered as entirely independent.

From KC_2 NASA dataset, the present tool considers the data given in Table-2. From this, the fault prone zones and non-fault zones are found and presented in Table 3. The median is calculated, and using the proposed model the initial categorization of classes is accomplished [8].

Table 2: KC_2 NASA Dataset.

S.No.	Defects	Cumulative No. of Defects
1.	75	75
2.	81	156
3.	86	242
4.	90	332
5.	93	425
6.	96	521

7.	98	619
8.	99	718
9.	100	818
10.	100	918
11.	100	1018

Table 3: Classification of Fault and Non-Fault.

S. No.	Fault (> 96)	Non Fault (<= 96)
1.	98	75
2.	99	81
3.	100	86
4.	100	90
5.	100	93
6.	-	96

The mvf of the proposed model and the results obtained are compared against benchmark as shown in Figure-3. The proposed NHPSRGM model shows m(t) value. This indirectly indicates that the proposed model performs far better than the benchmark and hence can be utilized for real-time development of software system.

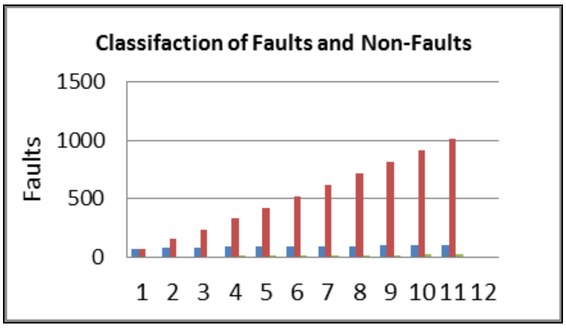

Figure 3. Classification of Faults and Non-Faults Plot.

CONCLUSION

The non-homogeneous Poisson process enhancement based method is found to be efficient for designing SRGMs. The proposed NHPSRGM tool efficiently improves the quality of the testing phase. It can ensure higher reliability and quality of the software product. The proposed NHPSRGM tool successfully eliminated the failures and hence this tool is found to be authentic.

REFERENCES

[1] Pan Jiantao, Software Reliability, 18 -849b, Dependable Embedded Systems, Spring 1999, Carnegie Mellon Univ.

[2] Kapur P.K., Tandon Abhishek, Anu G. Aggarwal, “Two Dimensional Flexible Software Reliability Growth Model and Related Release Policy”, Department of Operational Research, University of Delhi.

[3] Motley R W, Brooks,W D, “Analysis of Discrete Software Reliability Models”, Technical Report (Radctr pp 80-84, 1980, NY.

[4] Massey Vishwas, Prof. K J Satao, “Comparing Various SDLC Models And The New Proposed Model On The Basis Of Available Methodology”, IJARCSSE, ISSN: 2277 128X

[5] S Inoue & S Yamada, “Testing-Coverage Dependent Software Reliability Growth Modeling”, IJRQSE, , 2004, Volume 11, Issue 4, pp. 303 -312

[6] Thabasum Saira S, “Need For Design Patterns And Frameworks For Quality Software Development”, International Journal Of Computer Engineering & Technology(IJCET), 2012, Vol. 3, Issue 01, pp 54 - 58.

[7] Manivannan S., Dr. Balasubramanian S., “Software Process and Product Quality Assurance In IT Organizations”, International Journal Of Computer Engineering & Technology(IJCET), Vol. 01, Issue 01, May 2010, pp 1 - 11.

[8] Zhao Lu, “Software Reliability Growth Model Based On Fuzzy Wavelet Neural Network”, Second International Conf. On Future Computer & Communication, 2010, pp. V1 - 664 to V1 - 668.

[9] Gokhale S S et.al, “Unification of Finite Failure Non-Homogeneous Poisson Process Models through Test Coverage”, 7th International Symposium on Software Reliability Engineering, 1996, pp. 299–307.

[10] Kapur P K, Younes S. & Agarwala S, “Generalized Erlang model with n types of faults”, ASOR, 1995, No. 01, Volume 14, pp 5–11.

[11] Kapur P K and. Garg R B, “Software reliability growth model for an error removal phenomenon,” Software Engineering Journal, 1992, No. 04, Volume 7, pp. 291–294.

[12] Yamada Shigeru, “S-Shaped SRGMs and Their Applications”, IEEE Transactions on Reliability, 1984, Vol. R33 Issue. 04, pp 289 – 292.