Software Reliability Analyzer for improving Software Quality and Reliability

K.Venkata Subba Reddy

Assistant Professor, Department of Computer Science and Engineering
Muffakham Jah College of Engineering and Technology
Banjarahills, Hyderabad, Telangana, India.
kvsreddy2012@gmail.com

Prof.I.Ramesh Babu

Professor and Head, Department of Computer Science and Engineering
Acharya Nagarjuna University
Guntur, Andhra Pradesh, India.
rinampudi@hotmail.com

Abstract— A software product is tested throughout testing stage of the software development life cycle to check whether the software meets the user's necessities or not. For forecasting the reliability of the software, software reliability growth models are used. Various SRGMs are built from past few decades. This paper proposes an analyzer for software reliability development modeling in association with testing exertion or testing coverage, and also implements change-point. Change-Point is an exciting marvels' perceived throughout testing stage of software development. Through this concept we attempt to enhance the extrapolative accurateness of the model.

Keywords- SRGMs (Software Reliability Growth Models), Testing Effort (TE), TestingTime, Change –Point.

I. Introduction

The testing field of software catches a share of attention of numerous researchers because of the importance of the reliability of software. Releasing quality software products to the end-users software testing or software assessment is very important. To asses software reliability many developers have been proposed many SRGMs from few years. Software is said to be reliable that the likelihood of defect free software working for stated duration of time in a stated setting [1].Now a days industries needed reliable software especially those regions where in safety is important like satellite and shuttle control, hospitals, banking sectors e.tc. Hence the reliability catches a lot of attention over this period of time. To assess the software system's performance, dependability, forecast for that software is needed. The reliability of a software product is quantified depending on faults remains in the software system. Throughout the testing time the concept is on decreasing the residual faults there by growing the quality of the software.

Most of the SRGMs are established depending on the testing time, i.e calendar time, or central processing unit time, and number of faults identified [2,3]. All of these SRGMs are assumed that all the faults in the software system are of identical category. But this supposition is not true in the real time. The software comprises diverse kinds of errors or faults. Each fault needed diverse approaches and diverse quantity of testing exertion or testing coverage is needed for debugging. A test is said to be successful that which discloses the occurrence of the hidden faults. Locating the faults and the procedure designing for detecting the faults calls process of debugging [4]. This testing reportage aids a lot to software developers for evaluating the quality of the software and also it helps to find out that how much amount of exertion is required for refining the reliability of the software.

Here we are presenting a framework which is very helpful in software reliability growth modelling in accordance with testing coverage (TC) or testing effort expenditure [5].

1) Errors Observation.
2) Errors Isolation.
3) Errors Removing.

Software reliability growth model tool is established by bearing in mind a concept of change-point [12] here change-point is that every time software developers are trying to eliminate the identified errors they face two cases.

Case 1: Identified errors were removed successfully and there is no errors were introduced because of this elimination of the errors.

Case 2: Identified errors were removed successfully and there is new errors were introduced because of this elimination of the identified errors. Each and every time the number of faults were changed they are not fixed hence the case-2 is the seamless instance for that change-pont case.

Through our concept we can offer reliable and quality software to the end-users, with quantitative confidence. Hence safety is vital component in highly confidential objectives.

II. LITERATURE SURVEY

Firstly Zhao [2] introduce a change-point in software concept and Huang et.al [3] used SRGMs with Testing exertion functions and change-points. ShyurKapuret.al [4, 5] familiarized defective mending with change-point concept. Yamada introduced Yamada time dependent behavior of testing coverage (TC) procedure. We have so many Non-Homogeneous Poisson (NHPP) procedures centered SRGMs [6,7,8].Goel-okumoto[9,10]introduced a notion of fault discovery and elimination rate up surges in accordance with time. [11, 12] Flexible models like S-shaped and exponential growth curves were called Delayed S-shaped model of yamada et.al and exponential model of goal respectively. By observing this literature survey, it's very clear that from previous few eras numerous SRGMs have been proposed by so many researchers and also it's really a tedious task to makes the choice from this much of SRGMs. To overcome this, unified modeling methods have been introduced by numerous researchers. One of that unified methodologies is that Fault discovery and Fault amendment procedure [13].

Herein we are using such a integrated outline for demonstrating a frame work which is much generalized one for software reliability growth modeling in accordance with Testing coverage (TC) or Testing effort (TE) and also a change-point concept.

III. THE MODEL

A. Basic Assumptions

We consider that

- Software faces failures at execution time because of errors or faults presenting in the software.
  - The bug-identification or bug-elimination technique follows NHPPE (Non-Homogeneous Poisson Process Enhancement [14].
  - Usually faults which present in the software are of one of the following three types i.e. simple, hard and complex.
  - When a failure befalls an instantaneous debugging exertion takes charge to identify the reason and delete it.
  - At the phase of fault removal to check whether the error is successfully removed or not and also check is there any new faults generated or not.

B. Phases in NHPPE Model

1) Identification:

This Identification phase contains the software like as follows:

TNOC : Total number of attributes/objectives in a class.
WMPC : Whole approaches per class are measured by this software.
NOCP : this software counts the number of classes in a package.
SIZE : counts the total number of lines in a code.

2) Evaluation:

Here several software measures calculated from source code it is a part of tools construction.

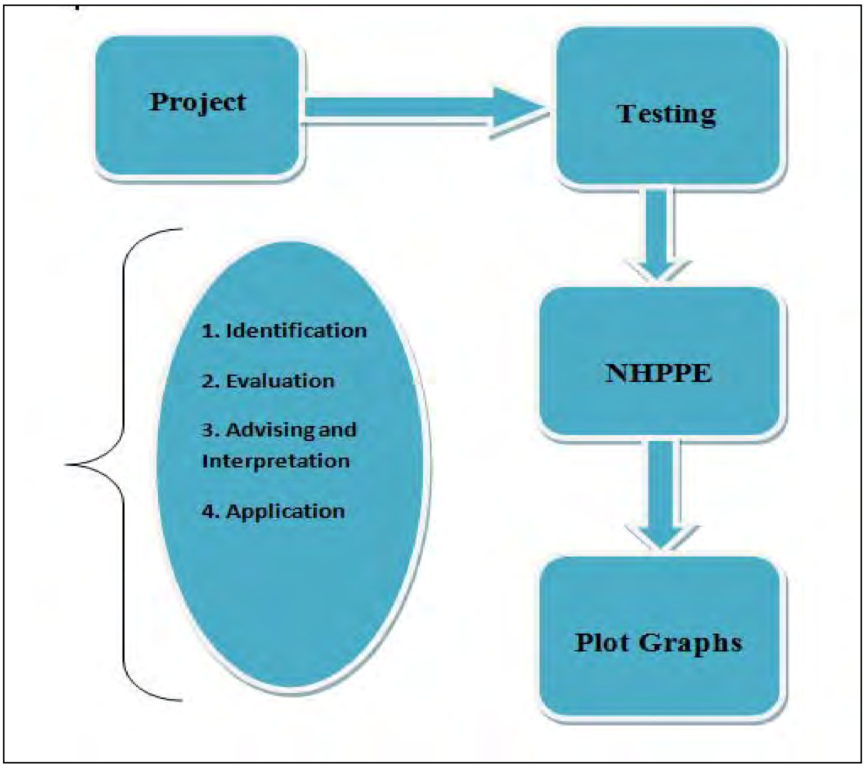

Figure 1. NHPPE Model Structure

3) Interpretation:

The above extracted evaluation values are taken by the elucidation portion of the SRGM tool, now these values and threshold values of the corresponding metric values are compared [15]. Here any of the following two cases are occurring: Case 1: Metric values which are extracted lies below the corresponding metric threshold value, in this case we don't have any occurrence of issue which is observed. Case 2: Metric values which are extracted lies above the corresponding metric brink value, in this case we have an occurrence of issue which is observed.

Application:

Here our SRGM analyser is ready to apply on the given source code.

Non-Homogeneous Poisson process enhancement technique: This model is a frame work for software reliability growth models which are of type applied on finite failure software [16]. By exhausting this notion anticipated number of failures perceived by time t.

The mean value function mathematical expression is as follows.

Mean value function: n(t)=a*b(t).

Where,

- a is Number of short comings,
- b(t) is Volume of possibility.

The expected numbers of failures are estimated as follows:

NF (Number of faults) = TNL*FD

Where

TNL=Total number of lines of code, and FD is Fault density.

The production function mathematical form: $X=BK^aL^b$

Wherein

X=Total production,
B=Total factor productivity,
K=Input of labour
L=Input of capital.
a and b are the output constants of elasticity's of capital and labour respectively.

The rate function R(t) is changes from time to time. Number of events anticipated between time t_1 to time t_2 is R $t_{1,}t_2$= $\int_{t1}^{t2} R(t)dt$ .In the time interval t_1 to t_2 the numbers of arrivals are $n(t_2)-n(t_1)$.

CONCLUSIONS

The proposed SRGM analyser which is helpful at testing phase in software development life cycle can ensures that the software is highly-reliable and quality software. To find out the faults/failures in the current software the non-homogeneous Poisson process enhancement concept is very helpful. The proposed software reliability analyzer is successfully eliminates the failures hence this analyzer is authentic [17].

REFERENCES

[1] A Detailed Study of Nhpp Software Reliability Models. Journal of Software, Vol. 7, No. 6, June 2012.

[2] Two Dimensional Flexible Software Reliability Growth Model With Two Types Of Imperfect Debugging. P.K. Kapur1, Anu G. AggarwalAndAbhishek Tandon3. Department Of Operational Research, University Of Delhi.

[3] Analysis of Discrete Software Reliability Models- Technical Report Radctr- 80-84" 1980; New York:Rome Air Development Center.

[4] Comparing Various Sdlc Models and The New Proposed Model On The Basis Of Available Methodology Vishwas Massey, Prof.K.J.Satao.

[5] Inoue S, Yamada S "Testing-Coverage Dependent Software Reliability Growth Modeling" International Journal Of Reliability, Quality And Safety Engineering,Vol. 11, No. 4, 303-312, 2004.

[6] S.Manivannan And Dr.S.Balasubramanian, "Software Process And Product Quality Assurance In It Organizations", International Journal Of Computer Engineering & Technology (Ijcet), Volume 1, Issue 1, 2010.

[7] [Software Reliability Growth Model Based On Fuzzy Wavelet Neural Network- 2010 2nd International Conference On Future Computer And Communication.

[8] International Journal Of Computer Engineering And Technology (Ijcet), Volume 4, Issue 2, March – April (2012).

[9] Unification of Finite Failure Non Homogeneous Poisson Process Models through Test Coverage.

[10] International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.4, July 2012 .

[11] P. K Kapur., S.Younes, and S. Agarwala, "Generalised Erlang model with n types of faults," ASOR Bulletin, Vol. 14, No.1, pp. 5–11, 1995.

[12] P. K. Kapur and. R. B. Garg, "Software reliability growth model for an error-removal phenomenon," Software Engineering Journal, Vol. 7, No. 4, pp. 291– 294, 1992.

[13] P. K., Kapur. V. B. Singh, and B Yang., "Software reliability growth model for determining fault types," in Proceedings of the 3rd International Conference on Reliability and Safety Engineering (INCRESE '07),pp. 334–349, 2007.

[14] P.K Kapur., P.C. Jha, A.K. Bardhan, , "Dynamic programming approach to testing resource allocation problem for modular software", in Ratio Mathematica, Journal of Applied Mathematics, Vol. 14, pp. 27-40, 2003.

[15] P.K. Kapur, D. N .Goswami, A. Bardhan, O. Singh, "Flexible software reliability growth model with testing effort dependent learning process", Applied Mathematical Modelling, Vol. 32, No. 7, pp. 1298– 1307,2008

[16] P.K. Kapur, J. Kumar and R. Kumar, "A Unified Modeling Framework Incorporating Change Point For Measuring Reliability Growth During Software Testing". To appear in OPSEARCH.

[17] P.K. Kapur, O. Shtnawi, A Aggarwal., R. Kumar, "Unified Framework for Developing Testing Effort Dependent.

AUTHORS PROFILE

K.Venkata Subba Reddy obtained his B.Tech in Information Technology from University of Madras in 2002 and received M.Tech in Software Engineering [CSE] from Bharath Institute of Higher Education and Research, Chennai in 2005; He is pursuing Ph.D., in Computer Science and Engineering, under the guidance of Prof.I.Ramesh Babu, at Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He is currently working in Computer Science & Engineering Department at Muffakham Jah College of Engineering & Technology, Banjarahills, Hyderabad. He has 12 years of experience in teaching. His research interests include Software Engineering, Software Reliability Engineering and Cloud computing. He is a life member of ISTE and a member of CSI.