Introducing Burr Type XII Testing-Effort Function in Software Reliability Growth Model

Dr. Syed Faizul Islam

Email Id: syed123468@gmail.com

Abstract

To asses, analyze and predict a software reliability we need to develop a software reliability growth model (SRGM). We can apply different types of testing-effort function during the development of SRGM. As we know that software reliability is the probability that software functions correctly under a given environment and during a specified period of time. The software reliability is highly related to the amount of testing-effort. Also, the reliability of software needs to be carefully assessed and analyzed during the software development process. Furthermore, the prediction of software reliability is also very important for failure free operation by software. It means, testing-effort functions are very important for the successful operation of SRM. In this paper, we have introduced a SRGM with a very descent testing effort function which is Burr Type XII testing effort function.

1. Introduction

Software is considered to have performed a successful operation, when it functions completely as expected, without any failure. However, computer software is created by human being and a high degree of certainty in reliability cannot be guaranteed (Musa et al., 1987). Software reliability is defined as the probability of failure free operation of a computer program for a specified time in a specified environment (Musa et al., 1987; Lyu, 1996) and is a key factor in software development process. Therefore, accurately modeling of software reliability and predicting its possible trends are essential for determining the software's reliability overall.

In software testing, the key factors are the testing-effort function. During the last three decades several authors have been proposed SRGMs with different testing effort functions. Here we are going to discuss the Burr Type XII testing effort function, which is very realistic.

Many testing-efforts are consumed, such as the CPU time, the human power and the executed test cases in software testing process (Ahmad et al., 2009). The consumed testing-effort indicates how the errors are detected effectively in the software and can be modeled by different distributions (Putnam, 1978; Pillai et al., 1997; Musa et al., 1987, 1999; Yamada et al., 1986, 1993; Yamada et al., 1990; Kapur et al., 1999; Huang et al., 2002; Khan et al., 2008; Bokhari et al., 2006, 2007). Actually, the software reliability is highly related to the amount of testing-effort expenditures spent on detecting and correcting software errors.

On the other hand, in practice, if we want to detect more potential faults for a short period of time, we may introduce new techniques or tools that are not yet used, or bring in consultants to make a radical software risk analysis. In addition, there are newly proposed automated test tools for increasing test coverage and can be used to replace traditional manual software testing regularly. The benefits to software developers/testers include increased software quality, reduced testing costs, improve release time to market, repeatable test steps, and improved testing productivity. These technologies can make software testing and correction easier, detect more bugs, save more time, and reduce much expense. Altogether, we wish that the consultants, new automated test tools or techniques could greatly help us in detecting additional faults that are difficult to find during regular testing and usage, in identifying and in assisting clients to improve their software development processes. Thus, the fault detection rate may not smooth and can be changed at some time moment called change point (Zhao, 1993; Chang, 2001; Chen et al., 2001; Enachescu, 2002; Shyur, 2003; Zou, 2003). In other words, the proportionality is not just a constant or in some case may be changed at some point of time which is called change point.

In the remaining of this paper, we have three more segments. In Segment 2, we have the description of Burr type XII testing-effort function. In section 3 we discussed SRGM with Burr Type XII Testing-Effort, also, we presented the comparison criteria for SRGM in Segment 4. In the segment 5 we have conclusions and references.

2. Burr Type XII Testing-Effort Function

Burr (1942) introduced twelve different forms of cumulative distribution functions for modeling actual data. The curve of Burr Type XII testing-effort function is very flexible and having a wide variety of possible expenditure patterns in real software projects (Bokhari et al., 2007). Also, Burr Type XII testing-effort function along with change point reveals significant prediction of software reliability. There are several advantages of Burr Type XII models over other SRGMs, these are:

- It covers the curve shape characteristics of normal, log-normal, gamma, logistic and Pearson type X distributions.
- It has simple algebraic forms for reliability and hazard rate functions.
- It provides a wide variety of density shapes along with functional simplicity.
- The special cases for this model include exponential, Weibull and Log-logistic.

We observed from the earlier studies (Huang et al., 2002) that actual testing-effort consumption pattern, sometimes the testing-effort consumption are difficult to describe only by Exponential, Rayleigh, Weibull or Logistic curve. But it is easy to describe with the help of Burr Type XII testing-effort function. Therefore, we tried to include a Burr Type XII test-effort function (Huang et al., 1997).

The current testing effort consumption curve at testing time t is given as

$$w(t) = \frac{\alpha\, \beta\, m\delta(\beta.t)^{\delta-1}}{[1 + (\beta.t)^{\delta}]^{m+1}} \qquad (1)$$

$\alpha > 0, \beta > 0, m > 0, \delta > 0, t > 0$

where α, β, m and δ are constant parameters, α is the total amount of testing-effort expenditure required by software testing, β is the scale parameters, and m, δ are shape parameters.

The integral form of equation (1) is called the cumulative testing-effort consumption of Burr type XII in time [0, t] and is given by

$$W(t) = \int_0^1 w(x)dx = \alpha[(1 - (1 + (\beta.t)^{\delta})^{-m}] \qquad (2)$$

$\alpha, \beta, m, \delta > 0$, $t \geq 0$

3. SRGM with Burr Type XII Testing- Effort

The basic assumptions for a SRGM with Burr Type XII testing-effort are:

- The fault removal process is modeled by an NHPP.
- The software application is subject to failures at random times caused by the remaining faults in the system.
- The mean number of faults detected in the time interval $(t, t + \Delta t)$ by the current testing-effort is proportional to the mean number of remaining faults in the system at time t, and the proportionality in a constant over time.
- Testing effort expenditures are described by Burr Type XII testing-effort function.

4. Comparison Criteria

There are three comparison criteria we can use to judge the performance of SRGM.

- *The Accuracy of Estimation*

The Accuracy of Estimation (AE) is defined as:

$$\left|\frac{m_a - a}{m_a}\right| \tag{3}$$

where m_a is the actual cumulative number (Goel et al.,1979) of detected faults after the test, and a is the estimated number of initial faults. For practical purposes, m_a is obtained from software fault tracking after software testing.

- *The Mean Square Errors*

We use the mean of (Kapur et al.,1996) squared errors (MSE) for long term predictions since it provides better understood measure of the differences between actual and predictive values (Lyu et al.,1992) .

The MSE can be calculated as follows:

$$\frac{\sum_{i=1}^{k}[m(t_i) - m_i]^2}{k} \tag{4}$$

where $m(t_i)$ is the expected number of faults by time t_i estimated by a model, and m_i is the observed number of faults by t_i. A smaller MSE indicates a smaller fitting error and better performance.

- *Relative Error*

The capability of the model to predict failure behavior from present and past failure behavior is called predictive validity, which can be represented by computing the relative error (RE) for the data (Musa et al., 1987; Huang, 2004).

$$Relative\ Error(RE) = \frac{m(t_q) - q}{q} \quad (5)$$

Assuming we observed q failures by the end of test time t_q, we use the failure data up to time t_e ($t_e \leq t_q$) to estimate the parameters of $m(t)$. Substituting the estimates of these parameters in the mean value function yields the estimate of the number of failures $m(t_q)$ by t_q. The estimate is compared with the actual number of q. The procedure is repeated for various values of t_e. We can check the predictive validity by plotting the relative error for different values of t_q.

5. Conclusions

As we can see that Burr Type XII testing-effort function gives a good predictive capability and better performance. And the model proposed using Burr Type XII testing-effort function is more realistic, practical and more suitable for describing the software reliability the. There are also, some more advantages of Burr Type XII testing-effort function having more advantages on other testing-effort functions. In the subsequent paper, we are going to propose a SRGM with Burr Type XII testing-effort function and Change Point.

REFERENCES

[1] Ahmad, N., Khan, M.G.M and Rafi, L. S.(2011) "Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging", IJCSNS International Journal of Computer Science and Network Security, Vol.11 No.1 pp. 161-171.
[2] Ahmad, N., Bokhari, M.U., Quadri, S.M.K. and Khan, M.G.M. (2007), "The Exponentiated Weibull Software Reliability Growth Model with various Testing-Efforts and Optimal Release Policy: a Performance Analysis", International Journal of Quality and Reliability Management, Vol. 25 (2), pp. 211-235]
[3] Ahmad, N., Khan, M. G. M., Quadri, S. M. K. and Kumar, M. (2009), "Modeling and analysis of software reliability with Burr type X testing-effort and release-time determination", Journal of Modeling in Management, Vol. 4(1), pp. 28-54.
[4] Ascher, H. and Feingold, H. (1984), "Repairable Systems Reliability: Modeling, Inference, Misconceptions and their Causes", Marcel Dekkes, New York.
[5] Huang, C.Y. and Kuo, S.Y. (2002),"Analysis of Incorporating Logistic Testing-Effort Function into Software Reliability Modeling", IEEE Transactions on Reliability, Vol. 51 No. 3, pp. 261-70.
[6] Khan, M. G. M., Ahmad, N., and Rafi, L. S. (2008), "Optimal Testing Resource Allocation for Modular Software Based on a Software Reliability Growth Model: a Dynamic Programming Approach", in: Proceedings of the International Conference on Computer Science and Software Engineering (CSSE-2008), Wuhan, China, IEEE Computer Society, pp. 759-762.
[7] Kapur, P.K. Garg, R.B. and Kumar, S. (1999), "Contributions to Hardware and Software Reliability", World Scientific Publications, Singapore.
[8] Lyu, M.R. (1996), "Handbook of Software Reliability Engineering", Mc Graw Hill.
[9] Musa, J.D., Iannino, A. and Okumoto, K.(1987), "Software Reliability; Measurement, Prediction and Application", McGraw – Hill, New York, NY.
[10] Putnam, L. (1978), "A Great Empirical Solution to the Macro Software Sizing and Estimating Problem", IEEE Transactions on Software Engineering", Vol.4, pp. 485-497.
[11] Pillai, K. and Nair, V.S.S. (1997), "A Model for Software Development Effort and Cost Estimation", IEEE Transactions on Software Engineering", Vol.4, pp. 343-461.

[12] Musa, J.D. (1999), "Software Reliability Engineering: More Reliable Software, Faster Development and Testing", McGraw – Hill, New York, NY.

[13] Yamada, S., Ohtera, H. and Narihisa, H. (1986), "Software Reliability Growth Model with Testing-Effort", IEEE Transactions on Reliability, Vol. R-35 No. 1, pp. 19-23.

[14] Yamada, S., Hishitani, J. and Osaki, S.(1993), " Software Reliability Growth with a Weibull Test-Effort: a Model and Application", IEEE Journals & Magazines, Vol. 42, pp. 100-106.

[15] Bokhari, M.U and Ahmad, N. (2006), "Analysis of a Software Reliability Growth Models: The Case of Log-Logistic Test-Effort Function", In: Proceeding of the 17th IASTED International Conference on Modeling and Simulation (MS'2006), Montreal, Canada, pp. 540-545.

[16] Bokhari, M.U and Ahmad, N.(2007), "Software Reliability Growth Modeling for Exponentiated Weibull Functions with Actual Software Failures Data", Advances in Computer, Science and Engineering: Reports and Monographs, World Scientific Publication Company, Singapore, Vol. 2, pp. 390-396.

[17] Zhao, M., (1993), "Change point Problems in Software and Hardware Reliability", Communication in Statistics-Theory and Methods", Vol. 22(3), pp. 757-768.

[18] Chang Y.P., (2001), "Estimation of Parameters for Non Homogeneous Poisson Process Software Reliability with Change Point Model", Communications in Statistics; Simulation and Computation, Vol. 30, pp. 623-635.

[19] Chen, M.C., Wu, H.P., Shyur, H.J., (2001), "Analysis Software Reliability Growth Model with Imperfect-Debugging and Change Point by Genetic Algorithms", In: Proceeding of the 29th International Conference on Computers and Industrial Engineering (ICC & IE), Montreal, Canada, pp. 520-526.

[20] Enachescu, D., (2002). "Model Comparison in Change Point Reliability Modeling", In: Proceeding of the Conference CompStat 2002, Short Communications and Poster.

[21] Shyur, H.J., (2003), "A Stochastic Software Reliability Model with Imperfect-Debugging and Change point". Journal of Systems and Software, Vol. 66, pp. 135-141.

[22] Ahmad, N. and Islam, A. (1996), "Optimal Accelerated Life Test Designs for Burr Type XII Distributions under Periodic Inspection and Type Censoring", Naval

[23] Research Logistics, Vol. 43, pp. 1049-1077.Bokhari, M.U and Ahmad, N. (2007), "Software Reliability Growth Modeling for Exponentiated Weibull Functions with Actual Software Failures Data", Advances in Computer, Science and Engineering: Reports and Monographs, World Scientific Publication Company, Singapore, Vol. 2,

[24] pp. 390-396.Huang, C.Y. and Kuo, S.Y. (2002), "A Pragmatic Study of International Symposium Software Reliability Engineering", ISSRE'98, pp. 111-123.

[25] Huang, C. Y., Kuo, S.Y. and Chen, I.Y. (1997), "Analysis of Software Reliability Growth Model with Logistic Testing-Effort Function," Proceeding of the 8th International Symposium on Software Reliability Engineering (ISSR'97). Albuquerque, New Maxico, pp. 378-388.

[26] Bokhari, M.U., Ahmad, M.I., and Ahmad, N. (2007a)," Software Reliability Growth Modeling for Burr Type XII Function: Performance Analysis", Presented in the International Conference on Modeling and Optimization of Structures, Processes and System (ICMOSPS' 07), Jan, 22-24, Durban, South Africa.

[27] Goel, A.L. and Okumoto, K. (1979), "Time Dependent Error-Detection Rate Model for Software Reliability and other Performance Measures," IEEE Transaction on Reliability, Vol. R-28, No.3 pp. 206-211.

[28] Kapur, P.K. and Garg, R.B. (1996), "Modeling an Imperfect Debugging Phenomenon in Software Reliability", Microelectronics and Reliability, Vol. 36 pp 645-650.

[29] Lyu, M.R. and Nikora, A(1992), "Applying Software Reliability Model More Effectively, "IEEE Software, pp. 43-52.
[30] Musa, J.D., Iannino, A. and Okumoto, K.(1987), "Software Reliability; Measurement, Prediction and Application", McGraw – Hill, New York, NY.
[31] Ohba, M. (1984), "Software Reliability Analysis Models", IBM Journal of Research and Development, vol. 28, pp. 428-443.
[32] Tohma Y., Jacoby, R., Murata, Y. and Yamamoto, M. (1989), "Hyper-Geometric Distribution Model to Estimate the Number of Residual Software Fault", in Proceeding of COMPSAC-89, Orlando, pp. 610-617.