INTEGRATING TESTING COVERAGE, EFFORT AND CHANGE POINT IN A SOFTWARE RELIABILITY GROWTH MODEL: A COMPREHENSIVE ANALYSIS

Sudeep Kumar

•

Department of Mathematics,
AIAS, Amity University, Uttar Pradesh – 201303 (India).
Corresponding author: sudeepdeepak777@gmail.com

Anu G Aggarwal

•

Department of Operational Research,
University of Delhi, Delhi (India)
E-mail: anuagg17@gmail.com

Abstract

Software reliability growth models (SRGMs) are essential for forecasting and controlling the reliability of software systems. In present article, we propose an enhanced SRGM that incorporates three important factors: testing coverage, testing effort, and change point detection. We introduce a novel testing coverage function that captures the delayed S-shaped behaviour commonly observed in software reliability growth. Weibull distribution is utilized to model the testing effort. Finally, we address the impact of change points in software reliability. To assess how well our suggested model works, we conducted experiments using real-world software failure data provide by Tandem computers. The results demonstrate that our model outperforms existing SRGMs by providing more accurate predictions and a better understanding of the interplay between testing coverage, testing effort, and change points.

Keywords: software reliability growth model, testing coverage, change point detection, reliability modelling, testing effort.

I. Introduction

Software reliability is a critical aspect of software systems, as it directly impacts their quality, user satisfaction, and overall success. To ensure reliable software, it is essential to accurately manage and predict its reliability throughout the development lifecycle. SRGMs have been widely employed for this purpose, aiming to estimate the number of remaining defects and predict the software's reliability over time. Yamada et al. [1] offer two software reliability evaluation models with imperfect debugging. Poonam and Ravneet [2] suggest a method for estimating software's remaining faults using both imperfect and perfect software reliability growth models. Aggarwal et al. [3] propose a non-homogeneous Poisson process (NHPP) based SRGM by combining imperfect debugging.

Traditional SRGMs often assume simplistic reliability growth patterns, such as the popular J-shaped or S-shaped curves. However, these models fail to capture the complex dynamics of software reliability growth, particularly the delayed S-shaped behaviour commonly observed in practice. In reality, the effectiveness of testing activities varies over time, with initial testing efforts being more efficient in detecting defects, followed by a period of diminishing returns as the software matures. Therefore, there is a need to incorporate a delayed S-shaped testing coverage function into SRGMs to better reflect the actual reliability growth process. With the variability of operational conditions, Chang et al. [4] create a new testing-coverage based SRGM. In the work, Chatterjee and Shukla [5] introduced and integrated a temporal variant fault detection probability into the s-shaped coverage SRGM. Using two distinct testing-time functions, Minamino et al. [6] offered to expand the current univariate SRGMs. Based on testing efforts that consider different testing coverage functions, Bibyan et al. [7] and Kumar et al. [8] developed three SRGMs.

Furthermore, testing effort plays a crucial role in improving reliability of the software. The allocation and distribution of testing effort throughout the development lifecycle significantly impact the detection and removal of defects. The Weibull distribution has proven to be a versatile tool for modeling testing effort, as it can accommodate various shapes of effort curves, such as increasing, decreasing, or constant effort over time. By integrating the Weibull distribution into an SRGM, we can capture the diverse characteristics of testing effort and enhance the accuracy of reliability predictions. Yamada [9] and Kapur et al. [10] suggested a NHPP based SRGM that is adaptable enough to characterise different software failure/reliability curves. For software reliability modeling, both time-dependent fault detection rate (FDR) and testing efforts are taken into account. Kapur et al. [11] and Khatri et al. [12] and Jin and Jin [13] proposed an SRGM that includes a testing-effort function. Pradhan et al. [14] offer SRGMs that incorporate the key notion of testing-effort function.

Additionally, change points pose a significant challenge in software reliability modeling. Change points occur when there are substantial modifications in the software or testing environment, leading to shifts in the software reliability growth pattern. Failure to identify and account for these change points can result in inaccurate reliability estimates and poor decision-making regarding software reliability management. Therefore, it is crucial to incorporate change point detection techniques into SRGMs to effectively handle these shifts and enhance the reliability estimation process. Zhao et al. [15], Inoue et al. [16] Arora et al. [17], Dhaka and Nijhawan [18] and Huang et al. [19] explored an SRGM that was based on the Non-Homogeneous Poisson Process while taking into account the phenomenon of change point.

This research article is to develop an enhanced SRGM by incorporating a delayed S-shaped testing coverage function, utilizing the Weibull distribution for modeling testing effort, and integrating change point detection techniques, the aim to provide a comprehensive approach to software reliability modeling. The proposed model will contribute to more accurate reliability predictions, better understanding of the interplay between testing coverage, testing effort, and change points, and ultimately improve software reliability management practices.

This article is organized into several sections. Section 2 presents the formulation of the proposed model. In this section, the model is described in detail, outlining its key components and mathematical formulation. In Section 3, the results of the experiments are presented, and their implications are discussed. This section highlights the findings and outcomes of the study, examining how well the proposed model performed in predicting software reliability. The implications of these results for software development and management are also explored. Lastly, Section 4 concludes the article and provides directions for future research.

II. Model Description and Development

Maecenas Software reliability estimation plays a critical role in ensuring the dependability of software systems. It is essential to accurately predict the growth of software reliability and effectively manage testing efforts for successful software development and deployment. However, many existing software reliability growth models have limitations that hinder their accuracy and effectiveness. Furthermore, change points, which represent significant shifts in the reliability growth pattern, are often disregarded in existing models. Change points can occur due to various factors, such as changes in development methodologies, software updates, or modifications in the system environment. Failing to account for change points can lead to inaccurate reliability predictions and ineffective management of testing efforts.

To address these limitations, research is focused on developing more advanced software reliability growth models. These models aim to incorporate the delayed S-shaped nature of testing coverage, variations in testing effort over time, and the impact of change points. By considering these factors, these models can provide more accurate and reliable predictions of software reliability growth.

I. Notations

a:	initial number of faults
b:	fault detection rate
ϕ:	constant
$m, m(t), m(W)$:	mean value function (MVF)
W/ W(t):	testing effort function
γ, ζ :	Weibull distribution scale, shape parameter
c/ c(W):	testing coverage function
τ	change point
β	scale parameter of logistic distribution function (constant)
α	fault generation rate
$\overline{W}$:	total testing effort available (constant)
b_1/b_2:	fault detection rate before/after change point

II. Model Assumptions

- During the testing phase, the rate at which faults are detected and removed may vary at any given point.
- The number of faults detected is directly proportional to the number of faults that remain undiscovered in the system.
- The level of testing coverage is influenced by the amount of effort invested in testing.
- Delayed S-shaped testing coverage function is incorporating.
- Fault detection rate for removal is logistically distributed $\left(\frac{b}{1+\beta e^{-bW(t)}}\right)$.

- The testing coverage can be quantified in terms of the rate of fault detection, represented as $\frac{c'(W(t))}{1-c(W(t))}$, where $c(W(t)) = \left[\frac{1-(1+bW)e^{-bW(t)}}{1+\beta e^{-bW(t)}}\right]$ represents the proportion of code covered during testing.
- The extent of code coverage during testing directly impacts the number of faults that can be detected.

III. Model Development

In accordance with the previously mentioned assumptions, we have formulated an NHPP based SRGM. The Mean Value Function (MVF) of our proposed model is expressed as follows:

$$\frac{dm(t)}{dt} = \frac{dm}{dc}\frac{dc}{dW}\frac{dW}{dt} \tag{1}$$

The first component represents the relationship between the level of testing coverage and number of detected faults. This component can be represented using a model that incorporates the delayed S-shaped testing coverage function, capturing the changing effectiveness of testing activities over time. The second component represents the rate at which the amount of testing coverage changes over time. This component can be modeled using the Weibull distribution for testing effort, which allows for varying patterns of testing effort, such as increasing, decreasing, or constant effort over time.

$$\frac{dm}{dc} = \frac{c'(W)}{1-c(W)}(a - m) \tag{2}$$

$$\frac{dc}{dW} = \phi(\text{constant}) \tag{3}$$

with the initial condition

$$m(0) = 0 \tag{4}$$

By combining these two components, the MVF of our proposed SRGM can be accurately represented and used to predict and manage software reliability growth.

$$m(W) = a\left[1 - \left(1 - c(W)\right)^{\phi}\right] \tag{5}$$

The testing effort is represented by Weibull function in this paper can be expressed as:

$$W(t) = \bar{W}\left(1 - e^{-\gamma t^{\zeta}}\right) \tag{6}$$

SRGM 1: Proposed model with perfect debugging

Let us consider the concept of perfect debugging without change point in proposed SRGM, we get the following mean value function:

$$m(t) = a\left[1 - \left(\frac{(1+bw(t)+\beta)e^{-bw(t)}}{1+\beta e^{-bw(t)}}\right)^{\phi}\right] \tag{7}$$

SRGM 2: Proposed model with imperfect debugging

Let us consider the concept of imperfect debugging without change point in proposed SRGM. The fault content of the software undergoes a constant rate of change over time. The rate of removal of faults is directly proportional to this changing fault content. It can be expressed mathematically as:

$$\mathrm{a(t) = a + \alpha m(t)}$$

and the corresponding mean value function is:

$$m(t) = \frac{a}{(1-\alpha)}\left[1 - \left(\frac{(1+bw(t)+\beta)e^{-bw(t)}}{1+\beta e^{-bw(t)}}\right)^{(1-\alpha)\phi}\right] \tag{8}$$

SRGM 3: Proposed model with perfect debugging and change point

Let us consider the concept of perfect debugging without change point in suggested SRGM. With the change-point at time (τ) as the detection point, the detection rate of faults is as follows:

$$\mathrm{b(t)} = \begin{cases} \mathrm{b_1; 0 \le t \le \tau} \\ \mathrm{b_2; t > \ \tau} \end{cases} \tag{9}$$

while satisfying the condition $\mathrm{W(t = \tau) = W(\tau)}$.
where

$$\mathrm{W(t - \tau) = W(t) - W(\tau)}$$

The resulting mean value function is:

$$\mathrm{m}(t) = \begin{cases} a\left[1 - \left(\frac{(1+b_1w(t)+\beta)e^{-b_1w(t)}}{1+\beta e^{-b_1w(t)}}\right)^{\phi}\right]; \ 0 \le \mathrm{t} \le \tau \\ \mathrm{a}\left[1 - \left(1 - \left(1 - \left(\frac{(1+b_1w(\tau)+\beta)e^{-b_1w(\tau)}}{1+\beta e^{-b_1w(\tau)}}\right)^{\phi}\right)\right)\left(\left(\left(\frac{1+\beta e^{-b_2w(\tau)}}{1+\beta e^{-b_2w(\mathrm{t})}}\right)\left(\frac{(1+b_2w(\mathrm{t})+\beta)}{(1+b_2w(\tau)+\beta)}\right)e^{-b_2w(\mathrm{t}-\tau)}\right)^{\phi}\right)\right]; \ t > \tau \end{cases} \tag{10}$$

III. Model Validation and Comparison Criteria

To exemplify the estimation process of both the existing and new SRGMs, the analysis used a dataset of software failures provided by tandem computer [20] that was tested over a 20-week period and contained a total of 100 faults that were found. 10,000 units of effort were used in the first release. SRGM parameters were computed using SPSS, a statistical programme. The proposed model's estimations are evaluated using four goodness-of-fit metrics, which are shown in Table 3. A change-point analyser is used to identify the change-point in the dataset. According to the analysis, the transition point happened in the eighth week of testing, and the corresponding efforts are 4.606 For k observations in the data, the actual number of detected faults, and their predicted values are represented by $\mathrm{x_j}$ and $\mathrm{m(t_j)}$ respectively. The explanation of the criteria and mathematical formulae in Table 1.

On the basis of the software failure data, the estimated parameter values for the testing effort function that follows the Weibull distribution are shown in Table 2. To determine the optimal values for the Weibull function shape and scale parameters, a non-linear regression technique was used in the estimate procedure. The estimated parameter values and fitting of the proposed SRGM's 1, 2 and 3 are shown in Table 3 and Figure 1.

Figure 1 provides an illustration of how the proposed software reliability growth models (SRGMs) are fitted to Tandem Computers' software testing data. The figure visually represents the alignment between the predicted values generated by the SRGMs and the actual observed data points.

Table 1: Goodness of fit criteria for model

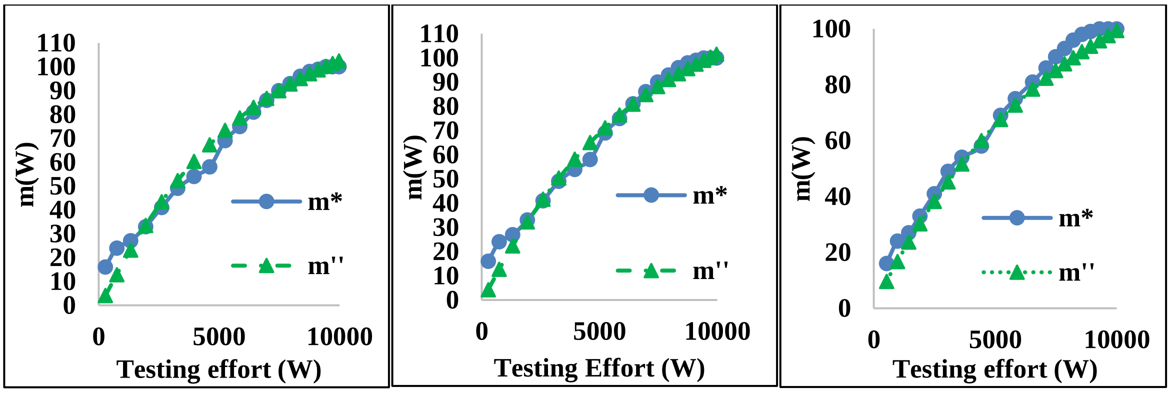
Performance Criteria	R^2	MSE	PRR	PP
Interpretation	larger the value of R^2, indicates model fitting is better to the data	smaller the value of MSE indicates a better fit of the model to the data, indicates model fitting is better to the data	smaller the value of PRR, indicates model fitting is better to the data	Smaller the value of PP, model fitting is better to the data
Expression	$1 - \frac{\text{residual SS}}{\text{actual SS}}$ SS: - sum of squares	$\frac{1}{k}\sum_{j=1}^{k}\left(m(t_j) - x_j\right)^2$	$\sum_{j=1}^{k}\left(\frac{m(t_j) - x_j}{m(t_j)}\right)^2$	$\sum_{j=1}^{k}\left(\frac{m(t_j) - x_j}{x_j}\right)^2$

Table 2: Weibull testing effort function parameter estimation

Effort function	Estimated Parameters	Estimated values
Weibull	W'	11740.754
	γ	0.024
	ζ	1.460

Table 3: Estimation results of parameter of proposed SRGM's

Proposed SRGM	a	b/b₁	b₂	ϕ	α	β
SRGM 1	125	.022	-	.008	-	.179
SRGM 2	101	.040	-	.005	.220	.099
SRGM 3	136	.133	.131	.001	-	.900



(i) (ii) (iii)

Figure 1: Goodness of fit curves for SRGM 1, 2 and 3 respectively

Table 4, on the other hand, presents a comprehensive comparison of the proposed SRGMs in terms of goodness-of-fit curves and metrics, considering various coverage functions. The table provides a quantitative evaluation of the models' performance by comparing different metrics such as mean square error (MSE), predictive power (PP), prediction ratio risk (PRR), and other relevant measures.

By examining the goodness-of-fit curves and metrics in Table 4, we can assess the effectiveness of each SRGM and identify the most suitable model based on their specific requirements and evaluation criteria. This table serves as a concise summary of the performance evaluation results and facilitates a comparative analysis of the proposed SRGMs across different coverage functions.

Both Figure 1 and Table 4 contribute to the comprehensive assessment of the proposed SRGMs, offering visual and quantitative insights into their fitting accuracy, predictive capabilities, and overall goodness of fit to Tandem Computers' software testing data. These visual and tabular representations aid in understanding and interpreting the results obtained from the comparison of the SRGMs, assisting researchers and practitioners in making informed decisions regarding model selection and further analysis.

Table 4: Goodness of fit criteria for proposed SRGM’s

Proposed SRGM	MSE	PP	PRR	R^2
SRGM 1	23.58422	0.848339775	9.489643	0.987
SRGM 2	19.40284	0.813731	8.381536	0.988
SRGM 3	17.79812	0.3138299	0.71271163	0.991

The inclusion of a change point in SRGM 3 contributes to its improved performance. The lower PP, MSE, and PRR values indicate that the predictions made by SRGM 3 align more closely with the actual observed data. This suggests that the model's estimates of software reliability are more accurate and precise when compared to SRGM 1 and SRGM 2. Additionally, the greater coefficient of determination (R^2) in SRGM 3 indicates that a larger proportion of the variability in the software reliability data can be explained by the model. This signifies a stronger relationship between the predicted values and the actual values, further supporting the superior accuracy of SRGM 3. Based on the results, it can be concluded that the proposed model, which incorporates a change point, offers a higher level of precision and accuracy compared to the model without a change point. These findings have implications for software development and management, as they highlight the potential benefits of considering change points in software reliability growth models for more reliable predictions and decision-making.

IV. Conclusion

In this research article, we have suggested an enhanced SRGM that integrates a delayed S-shaped testing coverage function, the Weibull distribution for testing effort, and change point detection techniques. To test the models, the researchers have utilized software failure data from a literature source, and the goodness of fit curves demonstrate the level of similarity between the actual and estimated values of the proposed model. The results of our research demonstrate the effectiveness and accuracy of the suggested SRGM. By integrating the delayed S-shaped testing coverage function, the Weibull distribution for testing effort, and change point detection techniques, our model provides valuable insights for the estimation and management of reliability of the software system. This research article presents a significant advancement in software reliability modeling by incorporating a delayed S-shaped testing coverage function, the Weibull distribution for testing effort, and change point detection techniques. The proposed model offers a comprehensive and accurate framework for software reliability estimation and management, contributing to the development of more reliable software systems and improving overall software quality.

References

[1] Aggarwal, A. G., Gandhi, N., Verma, V. and Tandon, A. (2019). Multi-release software reliability growth assessment: an approach incorporating fault reduction factor and imperfect debugging. *International Journal of Mathematics in Operational Research*, *15*(4), 446. https://doi.org/10.1504/IJMOR.2019.103006

[2] Arora, R., Tandon, A., Aggarwal, A. G. and Verma, V. (2019). Testing Resource based Optimal Release Policy for Software System Incorporating Fault Reduction Factor and Change point. *International Journal of Recent Technology and Engineering*, *8*(4), 1909–1920. https://doi.org/10.35940/ijrte.C5277.118419

[3] Bibyan, R., Anand, S., Aggarwal, A. G. and Kaur, G. (2023). Multi-release software model based on testing coverage incorporating random effect (SDE). *MethodsX*, *10*, 102076. https://doi.org/10.1016/j.mex.2023.102076

[4] Chang, I. H., Pham, H., Lee, S. W. and Song, K. Y. (2014). A testing-coverage software reliability model with the uncertainty of operating environments. *International Journal of Systems Science: Operations & Logistics*, *1*(4), 220–227. https://doi.org/10.1080/23302674.2014.970244

[5] Chatterjee, S. and Shukla, A. (2016). Effect of Test Coverage and Change Point on Software Reliability Growth Based on Time Variable Fault Detection Probability. *Journal of Software*, *11*(1), 110–117. https://doi.org/10.17706/jsw.11.1.110-117

[6] Dhaka, V. and Nijhawan, N. (2022). Effect of change in environment on reliability growth modeling integrating fault reduction factor and change point: a general approach. *Annals of Operations Research*, 1–35. https://doi.org/10.1007/S10479-022-05084-6/FIGURES/23

[7] Huang, Y.-S., Chiu, K.-C and Chen, W.-M. (2022). A software reliability growth model for imperfect debugging. *Journal of Systems and Software, 188*, 111267. https://doi.org/10.1016/j.jss.2022.111267

[8] Inoue, S., Fukuma, K. and Yamada, S. (2010). Two-Dimensional Change-Point Modeling for Software Reliability Assessment. *International Journal of Reliability, Quality and Safety Engineering*, *17*(06), 531–542. https://doi.org/10.1142/S0218539310003925

[9] Jin, C. and Jin, S.-W. (2016). Parameter optimization of software reliability growth model with S-shaped testing-effort function using improved swarm intelligent optimization. *Applied Soft Computing*, *40*, 283–291. https://doi.org/10.1016/j.asoc.2015.11.041

[10] Kapur, P. K., Goswami, D. N., Bardhan, A. and Singh, O. (2008). Flexible software reliability growth model with testing effort dependent learning process. *Applied Mathematical Modelling*, *32*(7), 1298–1307. https://doi.org/10.1016/j.apm.2007.04.002

[11] Kapur, P. K., Pham, H., Anand, S. and Yadav, K. (2011). A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation. *IEEE Transactions on Reliability*, *60*(1), 331–340. https://doi.org/10.1109/TR.2010.2103590

[12] Khatri, S. K., Kumar, D., Dwivedi, A. and Mrinal, N. (2012). Software Reliability Growth Model with testing effort using learning function. *2012 CSI Sixth International Conference on Software Engineering (CONSEG)*, 1–5. https://doi.org/10.1109/CONSEG.2012.6349470

[13] Kumar, S., Aggarwal, A. G. . and Gupta, R. (2023). Modeling the Role of Testing Coverage in the Software Reliability Assessment. *International Journal of Mathematical, Engineering and Management Sciences*, *8*(3), 504–513. https://doi.org/10.33889/IJMEMS.2023.8.3.028

[14] Minamino, Y., Inoue, S. and Yamada, S. (2022). Extension of software reliability growth models by several testing-time functions. In *System Assurances* (pp. 155–174). Elsevier. https://doi.org/10.1016/B978-0-323-90240-3.00010-2

[15] Panwar, P. and Kaur, R. (2016). Effect of Imperfect Debugging on Prediction of Remaining Faults in Software. In *Advances in Intelligent Systems and Computing* (pp. 175–185). https://doi.org/10.1007/978-981-10-0448-3_14

[16] Pradhan, V., Dhar, J. and Kumar, A. (2022). Testing-Effort based NHPP Software Reliability Growth Model with Change-point Approach. *Journal of Information Science and Engineering*, *38*(2), 343–355. https://doi.org/10.6688/JISE.202203_38(2).0004

[17] Yamada, S. and Ohtera, H. (1990). Theory and Methodology Software reliability growth models for testing-effort control. *European Journal Of Operational Research*, *46*, 343–349.

[18] Yamada, S., Tokuno, K. and Osaki, S. (1992). Imperfect debugging models with fault introduction rate for software reliability assessment. *International Journal of Systems Science*, *23*(12), 2241–2252. https://doi.org/10.1080/00207729208949452

[19] Zhao, J., Liu, H.-W., Cui, G. and Yang, X.-Z. (2006). Software reliability growth model with change-point and environmental function. *Journal of Systems and Software*, *79*(11), 1578–1587. https://doi.org/10.1016/j.jss.2006.02.030

[20] Wood, A. (1996). Predicting software reliability. *Computer*, *29*(11), 69–77.

https://doi.org/10.1109/2.544240