International Journal of Reliability, Risk and Safety: Theory and Application

Online ISSN: 2676-3346

OPEN ACCESS

Vol. 8/ Issue 1/ Year 2025/ pp. 74-81
DOI: 10.22034/IJRRS.2025.8.1.6
Received: 29 April 2025, Revised: 09 July 2025, Accepted: 12 July 2025

Available online at: https://www.ijrrs.com

Original Research Article

Considering Testing Environment Uncertainty in an NHPP Model with Exponentiated Weibull TEF

Javaid Iqbal[1*], Nyla Manzoor[1], Refath Farooq[1], Tariq Rasool[2], and Rabia Nazir[1]

1. Department of Computer Science, School of Applied Sciences and Technology, University of Kashmir, Srinagar, India

2. University Institute of Computing, Chandigarh University, Punjab, India

***iamjavaid@gmail.com**

Abstract

This study presents an approach to software reliability estimation by incorporating an Exponentiated Weibull Testing Effort Function into a Logistic Software Reliability Growth Model (SRGM) under the influence of uncertain factors. These uncertainty factors account for the variability and imprecision present in the software-testing environment, which often arise from assumptions and parameter estimations. The proposed model's practical applicability is illustrated using a real-world software failure dataset. To evaluate its performance, comparisons are made with several well-established SRGMs using three standard evaluation criteria. The results indicate that the proposed model offers improved reliability estimation accuracy and outperforms selected existing models by obtaining the highest R^2 value and the lowest MSE and SSE values.

Keywords: Logistic model; Project management; Software reliability; Software reliability Growth Model (SRGM); Software testing; Testing effort; Exponentiated Weibull.

Nomenclature

CTE	Current Testing Effort Expenditure
CTEE	Cumulative Testing-Effort Expenditure
EWTEF	Exponentiated Weibull Testing Effort Function
EW	Exponentiated Weibull
FRE	Fault Removal Efficiency
ID	Imperfect Debugging
LSRGM	Logistic Software Reliability Growth Model
LLTEF	Log-Logistic Testing Effort Function
MSE	Mean Square Error
NHPP	Non-Homogeneous Poisson Process
PD	Perfect Debugging
MVF	Mean Value Function
R^2	Coefficient of multiple determination
SRGM	Software Reliability Growth Model
SSE	Sum of Squared Errors
TE	Testing effort
TEF	Testing effort Function

List of symbols

$W(t)$	Cumulative testing-effort consumption at time t
$W^*(t)$	$W(t) - W(0)$
$W_{EW}(t)$	Cumulative testing-effort consumption at time t for EWTEF
$w(t)$	Current testing effort expenditure
$w_{EW}(t)$	Current testing effort expenditure EWTEF
v	Uncertainty of the testing environment
μ, μ_1	Certainty factor $\mu = 1 - v$
γ	Total testing effort expenditure
σ_1, σ_2	Scale parameter
ψ, τ	Shape parameters
$b(t)$	Time-dependent FDR function

How to cite this article:
J. Iqbal, N. Manzoor, R. Farooq, T. Rasool, and R. Nazir, "Considering testing environment uncertainty in an NHPP model with exponentiated weibull TEF," *International Journal of Reliability, Risk and Safety: Theory and Application*, vol. 8, no. 1, pp. 74-81, 2025, doi: 10.22034/IJRRS.2025.8.1.6.

1. Introduction

Software Reliability Growth Models (SRGMs) are primarily designed to identify a suitable theoretical or statistical distribution for observed failure data. Their goal is to assess the attained level of software reliability and support decision-making regarding the optimal balance between the desired reliability level and the earliest feasible time for releasing the software into a competitive market [1]. Numerous researchers and practitioners have proposed many SRGMs that involve use of perfect and imperfect debugging concepts. Reliability, as a quality attribute, is inherently probabilistic, making it difficult to measure it precisely. Although new research directions are being explored to improve its assessment, reliability remains a challenging metric [2]. NHPP-based SRGM has been proposed by Iqbal et al [3], with imperfect debugging and Fault Detection Rate (FDR) influenced by a learning factor. An imperfect debugging-based SRGM has been proposed by Iqbal et al [4] where a Dynamic Total Fault Content function has been used as a function of system testing. The Fault Reduction Factor (FRF) plays a crucial role in reliability. A constant FRF with imperfect debugging is used by Khurshid et al [5] to propose a multi-release model incorporating a testing effort function and a change point. An NHPP-based unified framework designed for multi-release, two-stage fault detection and removal processes has been used by Saraf et al [6] to propose a new model considering change point and other environmental factors. Based on the time lag between fault detection and fault correction, Saraf et al [7] have developed an NHPP-based two-stage framework considering change point, imperfect debugging and error generation. A constant FRF has been used by Khurshid et al [8] in an imperfect debugging multi-release model using the testing effort function and the changepoint. Saraf et al [9] have proposed another imperfect debugging-based two-stage multi-release framework with error generation and change point. An NHPP-based SRGM has been proposed by Saraf et al [10] for multi-release OSS. Another multi-release model by Khurshid et al [11] considers FRF. An NHPP-based Testing Coverage (TC) framework by Khurshid et al [12] includes Fault withdrawal efficiency. A model by Iqbal et al [13] captures the cumulative number of faults using an S-shaped curve, formulated through a novel mathematical expression based on a Sigmoid function to reflect the S-shaped characteristics.

SRGMs are built under the assumption that all faults are properly fixed and that no new problems are added throughout the debugging process in the perfect debugging environment [14]. In practical software projects, however, it is unrealistic to assume that no new errors are introduced during the process of error detection and removal [15]. There are three kinds of manifestations of the imperfect debugging process. These scenarios include fault removal efficiency and error generation [16]. It is well-known that some small fraction of faults remains resident in the software after testing [17]. In error generation, the total number of faults may alter as new faults may be introduced throughout the debugging process. FRE and error generation entail that some faults can be added during the repair with some probability of removal of a fault [18].

In this paper, we propose an LSRGM with an EW testing effort function and an uncertainty factor. The performance of the proposed LSRGM is put into perspective with other select existing SRGMs using select comparison criteria.

The rest of the paper is organized as follows: Section 2 explains related work; Section 3 explains the EW testing effort function. Section 4 explains Model development via its assumptions. Section 5 illustrates the numerical application of the LSRGM with the EW testing effort function. Section 6 concludes the paper. Then the references section follows.

2. Related work

Realistic capturing of the dynamics of the software testing environment has been the guiding mantra of the research in this field. Testing can be represented in terms of the effort put into the software testing and debugging, as well as meeting the release deadlines of the project. For the sake of measuring software reliability, SRGMs integrate testing effort in terms of mathematical functions called testing effort functions. Many testing efforts on function-based SRGMs have been proposed in the literature. A logistic testing effort function was developed by Huang et al. [19] as a test effort curve representation of the software testing environment. Pham [20] is credited with the proposal of a general LSRGM that uses time-dependent fault detection rate (FDR). The EW testing effort function was proposed by Bokhari et al. [21] and Ahmad et al. [22], wherein they have illustrated how the EW testing effort function is a more reliable and flexible testing effort curve. Ahmad et al. [23] suggested a flexible NHPP SRGM that uses the EW testing effort function in an S-shaped inflection model. Ahmad et al. [24] put forth an NHPP SRGM that used a Burr type X testing effort function to represent the software development TE consumption curve. Ahmad et al. [25] are credited with developing a flexible NHPP SRGM that incorporates a log-logistic testing effort function (LLTEF) into an inflection S-shaped model. Ahmad et al. [26] proposed an SRGM that combines the LLTEF with exponential and S-shaped SRGM. An effort-based SRGM proposed by Khurshid et al. [27] includes error generation, FRF and change point for development. Saraf et al. [28] also proposed an effort-based SRGM. Jain et al. [29] presented NHPP-based SRGM using the Gompertz testing effort function. Rafi et al. [30] put forth an SRGM with a Gompertz testing effort function. Iqbal et al [31] consider a study on various testing effort functions. Pradhan et al. [32] consider a study on an NHPP model with testing effort function and a Change point. Iqbal et al [33] consider NHPP models with Testing Coverage, ID, FRE and error generation. Iqbal et al [34] consider Burr testing effort functions in the NHPP model with an uncertainty factor. Haque & Ahmad [35] consider a reliability modeling under an uncertain testing environment. Li et al [36] evaluate the usefulness of infinite-failure NHPP models. Behera & Agarwal [37] discuss the effect of Weibull TEF under field environment in their reliability model. Behera & Agarwal [38] discuss effect of Power law TEF under operational uncertainty in their reliability model. Pham [39] defines a generalized fault-detection reliability model considering random operating environments. In [40], non-parametric approaches to reliability modeling are considered. [41] and [42] use neural networks. Mahapatra & Mahapatra [43] discuss a framework for reliability analysis under multiple change points and imperfect debugging. Roy et al [44] discuss the NHPP model with imperfect debugging and error generation. Roy & Pham [45] discuss the development of a conventional time series-based web error forecasting approach. Nazir et al [46] proposed a non-homogeneous Poisson process model with a sigmoid testing effort function. A Testing Coverage model proposed by Iqbal et al. [33] considers fault removal efficiency and error generation. Another recent consideration of researchers has been the uncertainty of testing and operational environments. The uncertainty in the testing environment has been considered in [47], according to which the uncertainty of the testing environment manifests in model assumptions and parameters. This motivates this work to consider both TE and the uncertainty of the testing environment in the LSRGM and evaluate the performance of the proposed model using the EW testing effort function.

3. Model Development

The Stochastic process $\{N(t), t \geq 0\}$ actually represents a count of the number of fault events on a timeline. A Poisson process is a suitable representation of this stochastic counting process when the failure intensity is non-homogeneous, represented below accordingly:

$$P\{N(t) = \mathrm{i}\} = \frac{[m(t)]^{\mathrm{i}}}{\mathrm{i}!} \mathrm{e}^{-m(t)}, \; i = 0,1,2,\ldots$$

The MVF $m(t)$ is given as follows:

$$m(t) = \int_0^t \lambda(h) \; dh$$

Where $m(t)$ is the expected number of errors detected in the time (0, t], $\lambda(h)$ is the failure intensity function.

NHPP makes the following standard assumptions [34]:

1. Any software experiences failures at random times caused by faults remaining in it.
2. Upon the failure event, the fault that led to that failure is immediately removed, and no new faults are injected into the software.
3. The NHPP counting process models the fault removal process in software testing.
4. The number of faults detected in the time interval $(t, t + \Delta t)$ by the current testing effort expenditures is proportional to the remaining faults in the system.

Under these standard assumptions of the NHPP framework, the additional assumptions for the development of LSRGM are:

- EW TEF describes the testing effort expenditure.
- There is uncertainty in the testing environment, which becomes manifest in assumptions and parameter values of the model equation. The impact of uncertainties is considered as $\underline{v}$ [47].
- Uncertainty of testing environments (v) is the combined impact of all uncertainty factors, with values in terms of probabilities/ percentages. In this study, a value of 0.05 is assigned to v [47].

Exponentiated Weibull TEF [21-22, 48]: The EW curve doesn't experience a peak phenomenon. The CTEE is [21]:

$$W_{EW}(t)=\gamma \left(1 - e^{-\sigma_2 t^{\psi}}\right)^{\tau} \quad \gamma > 0, \sigma_2 > 0, \psi > 0, \tau > 0 \tag{1}$$

The CTE is:

$$w_{EW}(t) = W_{EW}(t)' = \gamma \cdot \sigma_2 \cdot \psi \cdot \tau \cdot t^{\psi-1} \cdot e^{-\sigma_2 t^{\psi}} \left(1 - e^{-\sigma_2 t^{\psi}}\right)^{\tau-1} \tag{2}$$

Incorporating TE and with the uncertainty factor v in the modeling process [47], we have the following differential equation

$$\frac{\partial m(t)}{\partial t} * \frac{1}{w(t)} = b(t)[a - (1 - v)m(t)]$$

Consider [47]:

$$b(t) = b(1 - v)\frac{m(t)}{a}$$

Where b is the constant of proportionality, $w(t)$ is the CTE at time t, and a is the expected number of faults in the system (PD).

Assume $(1\text{-}v)$ =μ and solve the above differential equation under the boundary conditions $m(0)=0$. MVF m(t) is an increasing function of t.

$$m(t) = \frac{a}{\mu_1 + \sigma_1 e^{-b\mu_1 W(t))}}$$

$$m(t) = \frac{a}{\mu_1+\sigma_1 e^{-b\mu_1 W(t)-W(0)}} = \frac{a}{\mu_1+\sigma_1 e^{-b\mu_1 W^*(t)}} \tag{3}$$

Where $\sigma_1 = e^{-b}$ and b is an integral constant. Value of Uncertainty factor μ=1−v is fixated using a method as discussed in [47]. Substituting $W_{EW}(t)$ from equation (1) and $W_{EW}(0)$=0, we get the MVF of the proposed model using equation (3) as:

$$m_{EW}(t) = \frac{a}{\mu_1+\sigma_1 e^{-b\mu_1\left(\gamma\left(1-e^{-\sigma_2 t^{\psi}}\right)^{\tau}\right)}} \tag{4}$$

The MVFs of all models are presented in Table 1 below.

Table 1. Summary of the software reliability models and their mean value functions

Model name	Mean value function m(t)
G-O model[14]	$m(t) = a(1 - e^{-bt})$
HD/G-O model[49]	$m(t) = \left[n\left((e^{a} - c) / e^{ae^{-bt}} - c\right)\right]$
Yamada Exponential model [50]	$m(t) = a\left(1 - e^{\gamma\alpha(1-e^{-\beta t})}\right)$
Yamada Imperfect model [51]	$m(t) = \frac{ab}{\alpha+b}(e^{\alpha t} - e^{-bt})$
Yamada Imperfect model 2 [51]	$m(t) = a(1 - e^{bt})\left(1 - \frac{\alpha}{b}\right) + \alpha a t$
P-Z model[52]	$m(t) = \frac{1}{(1+\beta e^{-bt})}\left((c + a)(1 - e^{-bt}) - \frac{ab}{b-\alpha}(e^{-\alpha t} - e^{bt})\right)$
Proposed Model 1	$m_{EW}(t) = \frac{a}{\mu_1+\sigma_1 e^{-b\mu_1\left(\gamma\left(1-e^{-\sigma_2 t^{\psi}}\right)\right)^{\tau}}}$

4. Numerical Illustration

After deriving the Mean Value Function (MVF) for our Software Reliability Growth Model (SRGM), the next important step is to carefully evaluate its performance using recognized goodness-of-fit measures [53-55]. The criteria used for this analysis include: R^2 [13, 28, 56], SSE [28, 56], MSE [13, 28, 56]. Through the application of a comprehensive set of goodness-of-fit criteria, we seek to thoroughly evaluate the effectiveness of our SRGM in representing the underlying data trends. This assessment provides a solid foundation for understanding the model's performance, supporting well-informed decisions regarding its suitability and reliability in practical software development contexts.

a. MSE (Mean Square Error) measures the deviation between estimated and actual values and is defined as

$$MSE = \frac{1}{n-N}\sum_{i=1}^{n}(y_i - \widehat{m}(t_i))^2$$

Lower values of MSE indicate better goodness of fit performance.

b. SSE (Sum of Squared Error) measures the total deviation of predicted values from the actual observed data.

$$\text{SSE} = \sum_{i=1}^{n}\left(y_i - m(t)\right)^2$$

c. R-Square is a statistical measure that indicates how well a model explains the variability of the response variable around its mean.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - m(t_i)\right)^2}{\sum_{i=1}^{n}\left(y_i - \underline{y}\right)^2}$$

The Alcatel/Lucent Technologies Dataset [57] has been used to compare the performance of the proposed model vis-à-vis other select models.

5. Results and Discussion

A comparison between the proposed model and existing SRGMs, based on the LSE method, is presented in Table 2. Each of the proposed SRGMs involves four parameters $a, b,$ μ and σ. Since α is obtained by a method as described in [47], we did not apply the estimation method to it and fixating μ_1 = 0.95 [47]. In order to estimate the parameters $\gamma, \sigma_2, \psi, \tau$ of the proposed model with EW testing effort function, we fit the actual TE data into equation (1). The estimated parameters are obtained as: γ= 7048.395, σ_2 = 0.0001, ψ = 2.507, τ = 0.816.

Using the estimated values of $\gamma, \sigma_2, \psi, \tau,$ and μ_1 = 0.95, the SRGM parameters a, b, σ_1 in Equation (4) are derived as:
μ = 115.232, b= 0.001, σ_1 = 4.513.

The R^2value for proposed model 1 is 0.984, which is larger when compared with the existing models. MSE and SSE values are 24.752 and 1311.906, respectively.

Summary of results: Among all the models, the proposed model has the highest R^2 value. Similarly, the proposed model has the lowest MSE and SSE values**.** Thus, the proposed model performs best among the listed models.

Table 2. Comparison of SRGMs

Model name	Parameter estimation	R²	MSE	SSE
GO	*a=159.023* *b= 0.032*	0.975	39.454	2130.544
HD/GO	*a= 0.032* *b= 0.032* *c= 0.01*	0.975	40.198	2130.544
Yamada Exponential	*a=97720.4 33* *r= 0.057* *α= 0.028* *β= 0.032*	0.975	40.988	2131.386
Yamada Imperfect 1	*a=158.166* *b= 0.032* *α= 0.0001*	0.975	39.587	2137.732
Yamada Imperfect 2	*a=158.167* *b= 0.032* *α= 0.0001*	0.975	40.333	2137.692
P-Z	*a=200.000* *β=343.162* *b=9999.99* *c= 0.001* *α= 0.022*	0.962	.	3246.707
Proposed Model (with μ_1 = 0.95)	*a=115.232* σ_1 *= 4.513* *b= 0.0012*	0.984	24.752	1311.906

6. Conclusion and future Scope

Various SRGMs assist in making precise estimates of the cost, time, and resources required for a software project. In this study, we introduce an NHPP-based SRGM with a unique parameter designed to account for the impact of diverse uncertainties within an LSRGM framework. The proposed EW testing effort function model offers greater flexibility and better captures the real expenditure trends observed during the software development process. A comparison between the proposed SRGM and several existing NHPP-based models has been conducted and presented. In this study, the Alcatel/Lucent Technologies dataset [57] has been utilized to validate and assess the performance of the proposed model.

Additionally, three evaluation criteria were employed to compare model performance. The results demonstrate that the proposed models consistently achieve better fit and predictive accuracy than the existing models. In future work, change points, fault reduction factors, and multi-release SRGMs incorporating uncertainty and testing effort (TE) can also be explored.

The proposed model is particularly well-suited for software reliability data that exhibit S-shaped fault detection behavior, which is commonly observed in real-world projects with delayed fault discovery due to ramp-up in testing efforts. The use of the Exponentiated Weibull Testing Effort Function allows for flexible modeling of varying testing intensities — including early under-testing and later over-testing phases — making it ideal for projects with non-uniform, skewed testing resource allocation. Moreover, by incorporating uncertainty, the model is better aligned with practical environments where perfect test conditions are rarely guaranteed.

However, the model may have limited applicability in cases where:

- Fault detection follows a linear or exponential pattern without an S-shaped curve.
- Testing effort is constant or negligible, rendering the added complexity of EW-TEF unnecessary.
- Very small datasets are used, as the increased number of parameters may lead to overfitting or unreliable estimates.

Conflict of Interests

No conflict of interest has been expressed by the authors.

7. References

[1] J. Iqbal, N. Ahmad, and S. M. K. Quadri, "A software reliability growth model with two types of learning," in *International Conference on Machine Intelligence and Research Advancement*, Katra, India, 2013, pp. 498–503, https://doi.org/10.1109/ICMIRA.2013.105

[2] J. Iqbal, N. Ahmad, and S. M. K. Quadri, "A software reliability growth model with two types of learning and a negligence factor," in *IEEE Second International Conference on Image Information Processing*, Shimla, India, 2013, pp. 678–683, https://doi.org/10.1109/ICIIP.2013.6707680

[3] J. Iqbal, S. M. K. Quadri, and N. Ahmad, "An imperfect-debugging model with learning-factor based fault-detection rate," in International Conference on Computing for Sustainable Global Development, New Delhi, India, 2014, pp. 383–387, https://doi.org/10.1109/IndiaCom.2014.6828164

[4] J. Iqbal, "Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations," Cogent Engineering, vol. 4, no. 1, p. 1286739, 2017, https://doi.org/10.1080/23311916.2017.1286739

[5] S. Khurshid, A. K. Shrivastava, and J. Iqbal, "Fault prediction modelling in open source software under imperfect debugging and change-point," *International Journal of Open Source Software and Processes*, vol. 9, no. 2, pp. 1–17, 2018. https://doi.org/10.4018/IJOSSP.2018040101

[6] I. Saraf and J. Iqbal, "Generalized multi-release modelling of software reliability growth models from the perspective of two types of imperfect debugging and change point," *Quality and Reliability Engineering International*, vol. 35, no. 7, pp. 2358–2370, 2019, https://doi.org/10.1002/qre.2516

[7] I. Saraf and J. Iqbal, "Generalized software fault detection and correction modeling framework through imperfect debugging, error generation and change point," *International Journal of Information Technology*, vol. 11, no. 4, pp. 751–757, 2019. https://doi.org/10.1007/s41870-019-00321-x

[8] S. Khurshid, A. K. Shrivastava, and J. Iqbal, "Generalized multi-release framework for fault prediction in open source software," *International Journal of Software Innovation*, vol. 7, no. 4, pp. 86–107, 2019, https://doi.org/10.4018/IJSI.2019100105

[9] I. Saraf, A. K. Shrivastava, and J. Iqbal, "Generalised fault detection and correction modelling framework for multi-release of software," *International Journal of Industrial and Systems Engineering*, vol. 34, no. 4, pp. 464–493, 2020, https://doi.org/10.1504/IJISE.2020.106085

[10] I. Saraf, J. Iqbal, A. K. Shrivastava, and S. Khurshid, "Modelling reliability growth for multi-version open source software considering varied testing and debugging factors," *Quality and Reliability Engineering International*, vol. 38, no. 4, pp. 1814–1825, 2022, https://doi.org/10.1002/qre.3048

[11] S. Khurshid, A. K. Shrivastava, and J. Iqbal, "Generalised multi release framework for fault determination with fault reduction factor," International Journal of Information and Computer

Security, vol. 17, no. 1–2, pp. 164–178, 2022, https://doi.org/10.1504/IJICS.2022.121296

[12] S. Khurshid, J. Iqbal, I. A. Malik, and B. Yousuf, "Modelling of NHPP based software reliability growth model from the perspective of testing coverage, error propagation and fault withdrawal efficiency," *International Journal of Reliability, Quality and Safety Engineering*, vol. 29, no. 4, Art. no. 2250013, 2022, https://doi.org/10.1142/S0218539322500139

[13] J. Iqbal, T. Firdous, A. K. Shrivastava, and I. Saraf, "Modelling and predicting software vulnerabilities using a sigmoid function," International Journal of Information Technology, vol. 14, no. 2, pp. 649–655, 2022, https://doi.org/10.1007/s41870-021-00844-2

[14] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, vol. R-28, no. 3, pp. 206–211, Aug. 1979, https://doi.org/10.1109/TR.1979.5220566

[15] M. Ohba and X. M. Chou, "Does imperfect debugging affect software reliability growth?," in *11th international conference on Software engineering*, Pittsburgh, USA, 1989, pp. 237–244. https://doi.org/10.1145/74587.74619

[16] M. Ohba, "Software reliability analysis models," IBM Journal of Research and Development, vol. 28, no. 4, pp. 428–443, 1984. https://doi.org/10.1147/rd.284.0428

[17] V. Pradhan, A. Kumar, and J. Dhar, "Emerging trends and future directions in software reliability growth modeling," in *Engineering Reliability and Risk Assessment*, H. Garg and M. Ram, Eds. Elsevier, 2023, pp. 131–144, https://doi.org/10.1016/B978-0-323-91943-2.00011-3

[18] P. K. Kapur, H. Pham, S. Anand, and K. Yadav, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 331–340, 2011, https://doi.org/10.1109/TR.2010.2103590

[19] C. Y. Huang and S. Y. Kuo, "Analysis of incorporating logistic testing-effort function into software reliability modeling," *IEEE Transactions on Reliability*, vol. 51, no. 3, pp. 261–270, 2002, https://doi.org/10.1109/TR.2002.801847

[20] H. Pham, "A generalized logistic software reliability growth model," Opsearch, vol. 42, no. 4, pp. 322–331, 2005, https://doi.org/10.1007/BF03398744

[21] M. U. Bokhari and N. Ahmad, "Software reliability growth modeling for exponentiated weibull function with actual software failures data," *Innovative Applications of Information Technology for the Developing World*, pp. 390-395, 2007, https://doi.org/10.1142/9781860948534_0062

[22] N. Ahmad, M. U. Bokhari, S. M. K. Quadri, and M. G. M. Khan, "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, no. 2, pp. 211–235, 2008, https://doi.org/10.1108/02656710810846952

[23] N. Ahmad, M. G. M. Khan, and L. S. Rafi, "A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging," International Journal of Quality and Reliability Management, vol. 27, no. 1, pp. 89–110, 2010, https://doi.org/10.1108/02656711011009335

[24] N. Ahmad, M. G. M. Khan, S. M. K. Quadri, and M. Kumar, "Modelling and analysis of software reliability with Burr type X testing-effort and release-time determination," *Journal of Modelling in Managemen*, vol. 4, no. 1, pp. 28–54, 2009, https://doi.org/10.1108/17465660910943748

[25] N. Ahmad, M. G. M. Khan, and L. S. Rafi, "Software reliability modeling incorporating log-logistic testing-effort with imperfect debugging," AIP Conference Proceedings, vol. 1298, no. 1, pp. 651-657, 2010, https://doi.org/10.1063/1.3516395

[26] N. Ahmad and M. Z. Imam, "Software reliability growth models with Log-logistic testing-effort function: A comparative study," *International Journal of Computer Applications*, vol. 75, no. 12, pp. 6–11, Aug. 2013, https://doi.org/10.5120/13161-0818

[27] S. Khurshid, A. K. Shrivastava, and J. Iqbal, "Effort-based software reliability model with fault reduction factor, change point and imperfect debugging," *International Journal of Information Technology*, vol. 13, no. 1, pp. 331–340, 2021, https://doi.org/10.1007/s41870-019-00286-x

[28] I. Saraf, A. K. Shrivastava, and J. Iqbal, "Effort-based fault detection and correction modelling for multi release of software," Int. J. Inf. Comput. Secur., vol. 14, no. 3–4, pp. 354–379, 2021. https://doi.org/10.1504/IJICS.2021.114711

[29] M. Jain, P. Agarwal, and R. Solanki, "NHPP-Based SRGM Using Time-Dependent Fault Reduction Factors (FRF) and Gompertz TEF," in *Decision Analytics Applications in Industry*, P. K. Kapur, G. Singh, Y. S. Klochkov, and U. Kumar Eds. Singapore: Springer Nature Singapore, 2020, pp. 81-89, https://doi.org/10.1007/978-981-15-3643-4_6

[30] S. Rafi and S. Akthar, "Software reliability growth model with Gompertz TEF and optimal release time determination by improving the test efficiency," International Journal of Computer Applications, vol. 7, no. 11, pp. 34–43, 2010, https://doi.org/10.5120/1337-1741

[31] J. Iqbal, N. Manzoor, and R. Farooq, "Study and analysis of testing effort functions for software reliability modeling," in System Reliability and Security, pp. 143–163, Auerbach Publications, 2023, https://doi.org/10.1201/9781032624983-8

[32] V. Pradhan, J. Dhar, and A. Kumar, "Testing-effort based NHPP software reliability growth model with change-point approach," *Journal of Information*

Science and Engineering, vol. 38, no. 2, 2022, pp. 343-355, https://doi.org/10.6688/JISE.202203_38(2).0004

[33] J. Iqbal, R. Nazir, and T. Rasool, "NHPP based testing coverage model with fault removal efficiency and error generation," *International Journal of Reliability, Quality and Safety Engineering*, vol. 32, no. 1, 2024, 2025, Art. no. 2450046, https://doi.org/10.1142/S0218539324500463

[34] J. Iqbal, N. Manzoor, A. K. Shrivastava, and I. A. Malik, "Integrating Burr type testing effort functions in logistic reliability growth model with uncertainty factor," *International Journal of System Assurance Engineering and Management,* vol. 14, no. 6, pp. 2365–2375, 2023, https://doi.org/10.1007/s13198-023-02084-y

[35] M. A. Haque and N. Ahmad, "Software reliability modeling under an uncertain testing environment," *International Journal of Modelling and Simulation*, vol. 45, no. 1, pp. 321-327, 2023, https://doi.org/10.1080/02286203.2023.2201905

[36] S. Li, T. Dohi, and H. Okamura, "Are infinite-failure NHPP-based software reliability models useful?," Software, vol. 2, no. 1, pp. 1–18, 2022. https://doi.org/10.3390/software2010001

[37] A. K. Behera and P. Agarwal, "Effect of weibull TEF on SRGM subject to field environment," *International Journal of Reliability, Quality and Safety Engineering*, vol. 32, no. 03, 2024, Art. no. 2450054, https://doi.org/10.1142/S0218539324500542

[38] A. K. Behera and P. Agarwal, "Modeling software reliability with power law testing effort function under operational uncertain environment," *Journal of Software: Evolution and Process*, vol. 37, no. 7, 2025, Art. no. e70037, https://doi.org/10.1002/smr.70037

[39] H. Pham, "A generalized fault-detection software reliability model subject to random operating environments," Vietnam Journal of Computer Science, vol. 3, no. 3, pp. 145–150, 2016, https://doi.org/10.1007/s40595-016-0065-1

[40] A. M. U. Din, J. Iqbal, and S. Qureshi, "Software reliability prediction using neural networks: A non-parametric approach," in System Reliability and Security, pp. 14–27, Auerbach Publications. https://doi.org/10.1201/9781032624983-1

[41] A. M. U. Din, S. Qureshi, and J. Iqbal, "GNN approach for software reliability," in *System Reliability and Security*, Javaid Iqbal et al. Eds. New York: Auerbach Publications, 2023, http://dx.doi.org/10.1201/9781032624983-1

[42] J. Iqbal, F. S. Masoodi, I. A. Malik, S. Khurshid, I. Saraf, and A. M. Bamhdi, Eds., *System Reliability and Security: Techniques and Methodologies.* New York: CRC Press, 2023. https://doi.org/10.1201/9781032624983

[43] N. N., A. Mahapatra, and G. S. Mahapatra, "Predictive framework of software reliability analysis under multiple change points and imperfect debugging," Software Quality Journal, vol. 33, 2025, Art. no. 21, https://doi.org/10.1007/s11219-025-09718-3

[44] P. Roy, G. S. Mahapatra, and K. N. Dey, "An NHPP software reliability growth model with imperfect debugging and error generation," *International Journal of Reliability, Quality and Safety Engineering*, vol. 21, no. 2, 2014, Art. no. 1450008, https://doi.org/10.1142/S0218539314500089

[45] A. Roy and H. Pham, "Toward the development of a conventional time series based web error forecasting framework," *Empirical Software Engineering*, vol. 23, no. 2, pp. 570–644, 2018, https://doi.org/10.1007/s10664-017-9530-4

[46] R. Nazir, J. Iqbal, F. S. Masoodi, and A. K. Shrivastava, "Developing an innovative imperfect debugging software reliability growth model with enhanced testing coverage strategies," *International Journal of Reliability, Quality and Safety Engineering*, vol. 31, no. 5, 2024, Art. no. 2450017, https://doi.org/10.1142/S0218539324500177

[47] M. A. Haque and N. Ahmad, "A logistic growth model for software reliability estimation considering uncertain factors," *International Journal of Reliability, Quality and Safety Engineering*, vol. 28, no. 5, Art. no. 2150032, 2021, https://doi.org/10.1142/S0218539321500327

[48] M. Z. Ahmad and N. Ahmad, "Parametric software reliability growth model with testing effort: A review," in *International Conference on Computational Performance Evaluation*, Shillong, India, 2021, pp. 899–904, https://doi.org/10.1109/ComPE53109.2021.9752232

[49] S. Hossain and R. Dahiya, "Estimating the parameters of a non-homogeneous Poisson-process model for software reliability," *IEEE Transactions on Reliability*, vol. 42, no. 4, 1994. https://doi.org/10.1109/24.273589

[50] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on Reliability*, vol. 32, no. 5, 1983, https://doi.org/10.1109/TR.1984.5221826

[51] S. Yamada, K. Tokuno, and S. Osaki, "Imperfect debugging models with fault introduction rate for software reliability assessment," *International Journal of Systems Science*, vol. 23, no. 12, pp. 2241–2252, 1992, https://doi.org/10.1080/00207729208949452

[52] H. Pham and X. Zhang, "An NHPP software reliability model and its comparison," *International Journal of Reliability, Quality and Safety Engineering*, vol. 4, no. 3, pp. 269–282, Sep. 1997, https://doi.org/10.1142/S0218539397000199

[53] W. D. Penny, J. Mattout, and N. Trujillo-Barreto, "Bayesian model selection and averaging," in *Statistical Parametric Mapping: The Analysis of Functional Brain Images*, K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny, Eds. Academic Press, 2007, pp. 454–470, https://doi.org/10.1016/B978-012372560-8/50035-8

[54] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Processing Magazine*, vol. 21, no. 4, pp. 36–47, 2004, https://doi.org/10.1109/MSP.2004.1311138

[55] K. J. Friston, W. D. Penny, J. Ashburner, S. J. Kiebel, and T. E. Nichols, Eds. *Statistical Parametric Mapping: The Analysis of Functional Brain Images,* London: Academic Press, 2006, https://doi.org/10.1016/B978-0-12-372560-8.X5000-1

[56] J. Iqbal, S. M. K. Quadri, and N. Ahmad, "Software Reliability Growth Models from the Perspective of Learning Effects and Change-Point," Ph.D. dissertation, Department of Computer Sciences, University of Kashmi, 2014.

[57] S. Z. Ke and C. Y. Huang, "Software reliability prediction and management: A multiple change-point model approach," *Quality and Reliability Engineering International*, vol. 36, no. 5, pp. 1678–1707, 2020, https://doi.org/10.1002/qre.2653