Int J Syst Assur Eng Manag (October 2023) 14(5):1877–1887
https://doi.org/10.1007/s13198-023-02018-8

ORIGINAL ARTICLE

Multi-release testing coverage-based SRGM considering error generation and change-point incorporating the random effect

Ritu Bibyan[1] · Sameer Anand[1] · Anu G. Aggarwal[1] · Abhishek Tandon[1]

Received: 6 May 2023 / Revised: 19 June 2023 / Accepted: 26 June 2023 / Published online: 24 July 2023

Abstract The most essential step during the development of the software is the testing procedure which makes the software dependable and efficient. During this procedure, the observation and rectification of the faults play a significant role in increasing the reliability of the software. Various Software Reliability growth models (SRGMs) with multiple assumptions were presented by various researchers to study the software's reliability. It is well known that the fault observation/removal rate may get affected by irregular factors causing arbitrary effects. In this study, we aim to capture this irregular variation in fault observation/removal rate by expressing it in terms of testing coverage. The fault observation/removal process has been assumed as a stochastic process and modeled it using an Itô type of stochastic differential equation. Testing coverage enables software designers to check the software's excellence and to see if any extra efforts are required to enhance reliability. In this paper, we have developed an SRGM based on testing coverage by introducing the concept of chang-epoint, error generation, and fault detection rate with irregular fluctuations. The error generation implies that during the testing procedure faults are not disclosed entirely and more faults get introduced. Later on, we focused on the idea of multi-release by considering four releases. We have estimated the parameters of the model by using the fault dataset for consecutive releases of Tandem Computers and validated the performance by evaluating the various goodness-of-fit criteria.

Keywords SRGMs · Error generation · Stochastic differential equation · Testing coverage · Multi-release modeling

1 Introduction

Software product usage has become more and more every day and plays an essential role in safety–critical systems. Due to this, there is a huge market for high-end software products. If by any chance the safety critical systems have a failure, it may result in the unprecedented destruction of life and property. So, reliability is considered to be a major feature of software quality (Lyu 1996). The anticipation that software will be failure-free for a specific time period in operating surroundings is called software reliability. It is very important to understand that to have an effective method for developing reliable software as well as objectively estimating it (Wood 1996).

Many SRGMs are present in the literature to evaluate the reliability of software (Goel & Okumoto 1979; Ohba 1984b; Pham 2000; Yamada et al. 1983; Yamada et al. 1992a, b). There are two types of models namely perfect and imperfect debugging. One can assume that whenever a failure happens, the faults pertaining to that failure are instantly disclosed under perfect debugging. And, meanwhile, no fresh faults are generated. The traditional models under perfect debugging SRGM were the G–O model (Goel & Okumoto 1979), the S-shaped (Yamada et al. 1983, 1984), and the inflected S-shaped model (Ohba

✉ Sameer Anand
sananddu@gmail.com

Ritu Bibyan
ritubibyan.du.or.17@gmail.com

Anu G. Aggarwal
anuagg17@gmail.com

Abhishek Tandon
abhishektandon86@gmail.com

[1] Department of Operational Research, University of Delhi, New Delhi 110007, India

1984a). Under imperfect debugging, one can assume that there is no perfect disclosure of faults or fresh faults are generated during the removal procedure (Goel 1985; Kareer et al. 1990; Pham 1993; Xie et al. 2007; Yamada et al. 1992a, b). Further, imperfect debugging consists of two categories i.e., imperfect fault removal and error generation. During the debugging process, if the faults are not disclosed completely and no new faults are generated then it is called imperfect fault removal but, if the new faults are generated while disclosing initial fault content then it is called error generation. The imperfect debugging scenario was first studied by Goel and Okumoto (1979) and error generation concept was given by Obha (1989). The error generation with the exponential function was considered in the study given by Pham to propose imperfect debugging SRGM (Pham & Zhang 1997). Later, he extended his work by assuming the fault content function to be time-based (Pham et al. 1999). Kapur et al. (2011a, b) proposed a unification modeling based-SRGM by incorporating the idea of imperfect debugging along with error generation. In the past few years, many researchers have discussed the error generation scenario and imperfect debugging (Aggarwal et al. 2015; Ahmad et al. 2010; Anand et al. 2018; Bibyan et al. 2023; Kapur et al. 2011; Kapur et al. 2010; Kapur et al. 2008a, b; Verma et al. 2022; Zhang et al. 2014).

The fault detection/observation rate (FDR) has also been considered a significant aspect in improving the growth of software reliability. Moreover, it is assumed that there is instability in the rate because of some external factors like software size, testing efforts, testability, and so on. As a result, the fault observation rate can show alterations at any point in time. In literature, researchers contributed by capturing the concept of change points in SRGMs (Chang 2001; Dhaka & Nijhawan 2022; Kapur et al., 2007; Kapur et al. 2008a, b; Nijhawan & Aggarwal 2015; Tandon et al. 2016; J. Zhao et al. 2006; M. Zhao 1993). T. Pham and Pham (2019) proposed a generalized SRGM that considers multiple environmental factors and a stochastic fault detection process. The model incorporates a Weibull distribution to capture the randomness of the fault observation process and uses a generalized linear regression to estimate the influence of multiple environmental conditions on software reliability. Zhu and Pham (2022) formulated a generalized SRGM considering the impact of environmental factors such as the proportion of reused modules and change in the frequency of program specification. The randomness is reflected in the detection process. Bevia et al. (2023) analyzed the logistic SRGMs under random settings. The authors demonstrated the use of these models in a real-world setting.

The efficiency of the SRGMs can be increased by capturing the impact of some realistic issues occurring while the testing procedure such as Testing Coverage. The completeness and effectiveness of the test are measured using Testing Coverage (TC). Different testing coverage functions (TCF) have been studied by researchers such as Exponential (Hwang & Pham 2008), S-shaped (Hoang Pham & Zhang 2003), Logarithmic exponential (Huang & Lin 2006), Rayleigh (Xie et al. 2007), and Weibull (Gokhale et al. 1996). A model was proposed by Malaiya et al. (1994) to analyze the relatability between reliability and TC. She also attempted to explain the relationship between TC and defect coverage. Later, TC and testing time was considered to study their relatability with reliability by Malaiya et al. (2002) using log-exponential TCF. An SRGM along with a cost model by taking TC into account was suggested by Pham and Zhang (2003). He compared the performance measures with the pre-existing models and commented that the proposed model shows better results. Later, Pham (2014) discussed two models in which fault observation rate was taken as log–log distribution and TC was assumed to have the presence of uncertainty. The fault detection/observation rate was taken in terms of TC to study the ambiguity of the operating surroundings by Li and Pham (2017).

Before the system is offered in the market, a prolonged testing procedure takes place in which faults are observed and disclosed. Moreover, software users discover some faults in the software due to which firms release a new version. So, the process of detecting the fault content which remained in the software can be taken as a stochastic process (Øksendal 2003). In the past, several SRGMs with continuous state space have been proposed using SDE of an Itô type. A basic SRGM was proposed by Yamada et al. (1994) to explain FDR using Itô type SDE during the testing procedure. Later, SDE-based SRGM was developed using different exponential, inflection, and delayed S-shaped models by Yamada et al. (2003). The irregular fluctuations were considered to present the per-FDR despite a non-homogeneous Poisson process by Leet et al. (2004). He postulated that per-FDR is dependent on the testing time t. A flexible SDE-SRGM model was suggested in 2006 to describe FDR in the distributed development environment (Tamura & Yamada). A generalized SRGM using SDE was modeled by studying simple, hard, and complex faults by Kapur et al. (2007a, b; 2009). A model was proposed based on a similar concept but the fault detection rate was considered as TC with error generation by Bibyan et al. (2023) along with a multi-release model formulation.

With the rapid increase in the usage of software products, the necessity of providing a new release in the market has become quite mandatory for software developing companies. This new release either has some additional features or the faults of the previous release are debugged and then released again. Several SRGMs have been discussed in the past by incorporating SRGMs (Anu G Aggarwal et al. 2018; Kapur et al. 2006; Kapur et al. 2012; Kapur et al. 2014; Nijhawan & Aggarwal 2015; Tandon et al. 2016). Anand et al. (2018) considered a constant fault reduction factor in the presence of imperfect debugging and fault removal rate as a delayed S- shape to develop a 2-D multi-release SRGM. Recently, a time variable FRF-based multi-release SRGM with imperfect debugging was suggested by A G Aggarwal et al. (2019). During the testing procedure, the FRF and TC are the important aspects that should be considered to control software reliability. These two factors were considered to propose multi-release SRGM for an open source software by Tandon & Aggarwal (2020). The ambiguity of the random field operating environment was captured to generate a generalized multi-release SRGM by Mishra et al. (2023). The motivation behind the study is to study the impact of irregular fluctuations on testing coverage which has not been studied by any of the researchers in the past. The following are the contributions made by the study:

(i) A Testing coverage-based SRGM incorporating change points along with error generation has been developed using SDE of an Itô type to study the impact of irregular fluctuations.
(ii) Three different types of testing coverage models have been developed namely Exponential. Weibull, and delayed S-shaped.
(iii) Furthermore, a multi-release model has been generated for the proposed model.

The paper is organized as per the following sections:

- Sect. 2: a model framework has been provided including notations, assumptions, and mathematical model development. The mean value functions for the three different types of testing coverage models have been provided namely Exponential. Weibull, and Delayed S-shaped models.
- Sect. 3: a multi-release model has been explained for four consequent releases.
- Sect. 4: illustrates the numerical performance of the released dataset of Tandem Computer.
- Sect. 5: discusses the conclusion
- Sect. 6: contains the future scope.

2 Model's Framework

The notations and assumptions utilized to develop the model are as follows:

2.1 Notations

$S(t)$:	A random variable representing the fault content observed at testing time t
$m(t)$:	Mean Value Function / expected fault content observed in the time interval 0 to t
$a(t)$:	Overall fault content at time t
a:	Initial fault content
α_1, α_2 :	Error generation rate before and after the change point respectively
$c(t)$:	The portion of potential faults present in the code identified up to time t
$r(t)$:	Fault observation/removal rate
θ :	Shape parameter
a_i:	Initial Fault content for ith version; i = 1,2,3,4
t_i:	Release time for ith version
τ_i:	Change points
$F_k(t)$:	Cumulative Failure function before the change point for the kth model
$G_k(t)$:	Cumulative Failure function after the change point for the kth model
$F_{ik}(t)$:	Cumulative failure function before change point for the ith version for kth model
$G_{ik}(t)$:	Cumulative failure function after change point for the ith version kth model

2.2 Assumptions

(a) The fault observation/removal process is modeled as a stochastic process with continuous state space.
(b) The fault content reduces as the testing increases.
(c) During the execution, the system is subjected to failure because of the residual faults in the code.
(d) The fault observation rate can be expressed in terms of TC as $\frac{c\prime(t)}{1-c(t)}$.
(e) As soon as the failure is observed, instant action is taken to debug the faults by the developers.
(f) When the faults are being removed, new faults generate with the probability α.

2.3 Model development

Many SRGMs in the literature are based on NHPP by taking the fault observation rate to be a discrete counting process. The fault content observed during the testing procedure is proportional to the size of the code of the software. Whereas, the fault content removed or corrected in the process of debugging is inversely proportional to the size of the code of the software (Shigeru & Akio 2003). This nature of fault observation/removal rate is handled using a stochastic model. Moreover, the faults

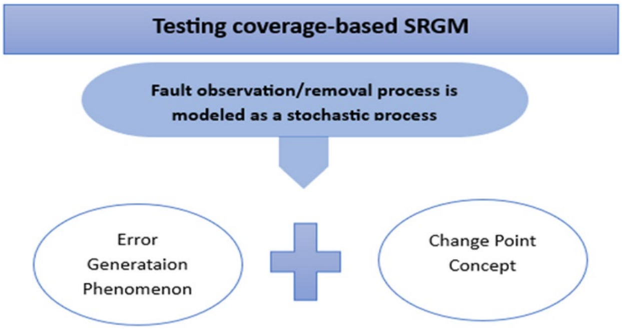
Testing coverage-based SRGM
Fault observation/removal process is modeled as a stochastic process
Error Generataion Phenomenon
Change Point Concept

Fig. 1 Testing coverage-based SRGM

are observed and removed during testing from the software code due to which the residual fault content reduces as the testing continues. We assume that new faults are generated while debugging with an introduction rate α and consider the following differential equation to model SRGM with a change point as shown in Fig. 1:

$$\frac{dS(t)}{dt}=\begin{cases}\frac{c_1\prime(t)}{1-c_1(t)}\left[a+\alpha_1S(t)-S(t)\right];0\le t\le\tau\\ \frac{c_2\prime(t)}{1-c_2(t)}\left[a-\alpha_1S(\tau)-\alpha_2(S(t)-S(\tau))-S(t)\right];t>\tau\end{cases}\tag{1}$$

Equation (1) can be rewritten as:

$$\frac{dS(t)}{dt}=\begin{cases}(1-\alpha_1)\frac{c_1\prime(t)}{1-c_1(t)}\left[a_1-S(t)\right];0\le t\le\tau\\ (1-\alpha_2)\frac{c_2\prime(t)}{1-c_2(t)}\left[a_2-S(t)\right];t>\tau\end{cases}\tag{2}$$

Here, $a_1=\frac{a}{(1-\alpha_1)}$ and $a_2=\frac{a+(\alpha_1-\alpha_2)S(\tau)}{(1-\alpha_2)}$

The irregular variation has been expressed in terms of noise as $\sigma\gamma(t)$. Now, we incorporate this noise in the fault observation rate and extend the Eq. (2) to a stochastic differential as:

$$\frac{dS(t)}{dt}=\begin{cases}\left[(1-\alpha_1)\frac{c_1\prime(t)}{1-c_1(t)}+\sigma\gamma(t)\right]\left[a_1-S(t)\right];0\le t\le\tau\\ \left[(1-\alpha_2)\frac{c_2\prime(t)}{1-c_2(t)}+\sigma\gamma(t)\right]\left[a_2-S(t)\right];t>\tau\end{cases}\tag{3}$$

The Eq. (3) has been extended to an SDE of an Itô type and is given as:

$$S(t)=\begin{cases}(1-\alpha_1)\left(\frac{c_1\prime(t)}{1-c_1(t)}-\frac{1}{2}\sigma^2\right)\left(a_1-S(t)\right)dt+\sigma\left(a_1-S(t)\right)dw(t);0\le t\le\tau\\ (1-\alpha_2)\left(\frac{c_2\prime(t)}{1-c_2(t)}-\frac{1}{2}\sigma^2\right)\left(a_2-S(t)\right)dt+\sigma\left(a_2-S(t)\right)dw(t);t>\tau\end{cases}\tag{4}$$

where W(t) represents the one-dimensional Wiener process and its time derivative is given by $\frac{dW(t)}{dt}=\gamma(t)$. The Weiner process is a Gaussian process with the given properties (Kapur et al. 2009):

$$Probability[w(0)=0]=1,$$

$$Expectationofw(t)=0,$$

$$E[w(t)w(t\prime)]=\min[t,t\prime],$$

Using the initial condition at $t=0,S(t)=0$, and at $t=\tau,S(t)=S(\tau)$, we get the solution of Eq. (4) as:

$$S(t)=\begin{cases}a_1\left(1-e^{\left(-(1-\alpha_1)\int_0^\tau\frac{c_1\prime(t)}{1-c_1(t)}dt-\sigma W(t)\right)}\right);\ 0\le t\le\tau\\ a_2\left(1-e^{\left(-(1-\alpha_1)\int_0^\tau\frac{c_1\prime(t)}{1-c_1(t)}dt-(1-\alpha_2)\int_\tau^t\frac{c_2\prime(t)}{1-c_2(t)}dt-\sigma W(t)\right)}\right)+\frac{(\alpha_1-\alpha_2)}{1-\alpha_2}S(\tau);\ t>\tau\end{cases}\tag{5}$$

Now, we get the MVF by taking the expectation of $S(t)$ as:

$$m(t)=\begin{cases}a_1\left(1-e^{\left(-(1-\alpha_1)\int_0^\tau\frac{c_1\prime(t)}{1-c_1(t)}dt+\frac{\sigma^2t}{2}\right)}\right);\ 0\le t\le\tau\\ a_2\left(1-e^{\left(-(1-\alpha_1)\int_0^\tau\frac{c_1\prime(t)}{1-c_1(t)}dt-(1-\alpha_2)\int_\tau^t\frac{c_2\prime(t)}{1-c_2(t)}dt+\frac{\sigma^2t}{2}\right)}\right)+\frac{(\alpha_1-\alpha_2)}{1-\alpha_2}S(\tau);\ t>\tau\end{cases}\tag{6}$$

SRGM-1: Exponential testing coverage function

$$r(t)=\frac{c\prime(t)}{1-c(t)}=\begin{cases}b_1;\ 0\le t\le\tau\\ b_2;\ t>\tau\end{cases}\tag{7}$$

Now using Eqs. (6) and (7)

$$m_{srgm}(t)=\begin{cases}\frac{a}{(1-\alpha_1)}\left(1-e^{\left(-(1-\alpha_1)b_1t+\frac{\sigma^2t}{2}\right)}\right);\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}\left(1-e^{\left(-(1-\alpha_1)b_1\tau-(1-\alpha_2)b_2(t-\tau)+\frac{\sigma^2t}{2}\right)}\right)+\frac{(\alpha_1-\alpha_2)}{1-\alpha_2}S(\tau);\ t>\tau\end{cases}\tag{8}$$

$$m_{srgm1}(t)=\begin{cases}\frac{a}{(1-\alpha_1)}F_1(t);\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}aG_1(t);\ t>\tau\end{cases}\tag{9}$$

SRGM-2: Delayed S-shaped Testing Coverage

$$r(t) = \frac{c\prime(t)}{1 - c(t)} = \begin{cases} \frac{b_1^2}{1+b_1t}; \ 0 \le t \le \tau \\ \frac{b_2^2}{1+b_2t}; \ t > \tau \end{cases} \tag{10}$$

Now using Eq. (6) and (10)

$$m_{srgm2}(t) = \begin{cases} \frac{a}{(1-\alpha_1)}\left(1 - \left(1 + b_1t\right)^{(1-\alpha_1)} e^{\left(-(1-\alpha_1)b_1t+\frac{\sigma^2 t}{2}\right)}\right); \ 0 \le t \le \tau \\ \frac{a}{(1-\alpha_2)}\left(1 - \left[\frac{(1+b_1t)}{(1+b_2\tau)}\right]^{(1-\alpha_2)} \left[1 + b_1\tau\right]^{(1-\alpha_1)} e^{\left(-b_1(1-\alpha_1)\tau-b_2(1-\alpha_2)(t-\tau)+\frac{\sigma^2 t}{2}\right)}\right) + \frac{(\alpha_1-\alpha_2)}{1-\alpha_2}S(\tau); \ t > \tau \end{cases} \tag{11}$$

$$m_{srgm2}(t) = \begin{cases} \frac{a}{(1-\alpha_1)}F_2(t); \ 0 \le t \le \tau \\ \frac{a}{(1-\alpha_2)}G_2(t); \ t > \tau \end{cases} \tag{12}$$

SRGM-3: Weibull Testing Coverage

$$r(t) = \frac{c\prime(t)}{1 - c(t)} = \begin{cases} b_1\theta t^{\theta-1}; \ 0 \le t \le \tau \\ b_2\theta t^{\theta-1}; \ t > \tau \end{cases} \tag{13}$$

Now using Eqs. (6) and (13)

$$m_{srgm3}(t) = \begin{cases} \frac{a}{(1-\alpha_1)}\left(1 - e^{\left(-(1-\alpha_1)b_1t^\theta+\frac{\sigma^2 t}{2}\right)}\right); \ 0 \le t \le \tau \\ \frac{a}{(1-\alpha_2)}\left(1 - e^{\left(-(1-\alpha_1)b_1\tau^\theta-(1-\alpha_2)b_2(t^\theta-\tau^\theta)+\frac{\sigma^2 t}{2}\right)}\right) + \frac{(\alpha_1-\alpha_2)}{1-\alpha_2}S(\tau); \ t > \tau \end{cases} \tag{14}$$

$$m_{srgm3}(t) = \begin{cases} \frac{a}{(1-\alpha_1)}F_3(t); \ 0 \le t \le \tau \\ \frac{a}{(1-\alpha_2)}G_3(t); \ t > \tau \end{cases} \tag{15}$$

For simplification, $\alpha_1 = \alpha_2 = \alpha$ has been considered.

3 Multi-release model

In addition to ensuring reliability, developers must stay up with the market's expanding needs such as problems in the current release, requirements for new software updates, market competition, different company policies, the need for new functionalities in the code, and so on. As a result, software development organizations continue to improve their products by adding new features, and codes, and correcting previously recognized faults. Swift advances in hardware and favorable financial conditions have compelled companies to constantly innovate and improve their goods to preserve their market position and overcome competitor products. The improvements are introduced into the marketplace at various periods. Integration of both detected faults of the previous version and the additional faults of the new release is required for modeling the reliability improvement of multi-release software. Although new versions are developed to keep up with the growth and changing demands of customers, they can additionally increase the failure rate and fault content of the software system. As a result, the number of problems eliminated over each release is identified differently. During our research, we discovered that the new faults are fixed in the present release and the unresolved faults are fixed in the next release, implying that the next release takes into account the remaining faults of the recently released version of the software. However, in the case of obstacles, the remaining faults from Release 1 are resolved in Releases 2, 3, and 4. It indicates that in the software development environment, residual faults from previous versions are passed on to subsequent releases (up to the following three or four). We assume here that lingering faults from Release 1 can pass up to Release 4. The multi-release model has been developed as follows:

A. **Release 1:**

Before launching the software in the market, testers test the software very precisely. The testers aim to remove all the faults before launching it which is practically not possible. So, when the software is launched it contains few faults and mathematically, we can represent the MVF of the very first release as:

$$m_{1k}(t) = \begin{cases} \frac{a_{1k}}{1-\alpha_{1k}}F_{1k}(t); \ 0 \le t \le \tau_1 \\ \frac{a_{1k}}{1-\alpha_{1k}}G_{1k}(t); \ \tau_1 < t > t_1 \end{cases} \tag{16}$$

B. **Release 2:**

To preserve their reputation in the market, software firms bring up some additional features to the code of the

Table 1 Tandem computers dataset

Releases	Testing time (in weeks)	Cumulative no. of faults
1	20	100
2	19	120
3	12	60
4	19	42

software. This new addition to code generates complexity by upsurging the number of faults. Also, it is reasonable to consider the fact that some faults get carried forward from the first release which could not get removed. So, the MVF corresponding to the second release is represented as:

$$m_{2k}(t) = \begin{cases} \left[\frac{a_{2k}}{1-\alpha_{2k}} + \frac{a_{1k}}{1-\alpha_{1k}}\left(1 - F_{1k}\left(t_1\right)\right)\right] F_{2k}(t - t_1);\ t_1 \le t \le \tau_2 \\ \left[\frac{a_{2k}}{1-\alpha_{2k}} + \frac{a_{1k}}{1-\alpha_{1k}}\left(1 - G_{1k}\left(t_1\right)\right)\right] G_{2k}(t - t_1);\ \tau_2 < t \le t_2 \end{cases} \tag{17}$$

C. **Release 3:**

Along the same line, we consider the MVF of the third release to be affected by the previous release. So, the MVF corresponding to the third release can be represented as:

$$m_{3k}(t) = \begin{cases} \left[\begin{array}{c} \frac{a_{3k}}{1-\alpha_{3k}} + \frac{a_{2k}}{1-\alpha_{2k}}\left(1 - F_{2k}\left(t_2\right)\right) \\ + \frac{a_{1k}}{1-\alpha_{1k}}\left(1 - F_{1k}\left(t_1\right)(1 - F_{2k}\left(t_2\right)\right) \end{array}\right] F_{3k}(t - t_2);\ t_2 \le t \le \tau_3 \\ \left[\begin{array}{c} \frac{a_{3k}}{1-\alpha_{3k}} + \frac{a_{2k}}{1-\alpha_{2k}}\left(1 - G_{2k}\left(t_2\right)\right) \\ + \frac{a_{1k}}{1-\alpha_{1k}}\left(1 - G_{1k}\left(t_1\right)\right)\left(1 - G_{2k}\left(t_2\right)\right) \end{array}\right] G_{3k}(t - t_2);\ \tau_3 < t \le t_3 \end{cases} \tag{18}$$

D. **Release 4:**

Similarly, the MVF corresponding to the fourth release can be represented as:

Table 2 Estimated parameters

Models	Releases	a	b_1	b_2	α	σ	θ
SRGM 1	Release 1	103.382	0.081	0.209	0.316	0.0012	-
	Release 2	123.436	0.0698	0.163	0.512	0.0347	-
	Release 3	65.099	0.054	0.189	0.001	0.0223	-
	Release 4	44.7	0.0472	0.091	0.001	0.014	-
SRGM 2	Release 1	102.384	0.217	0.231	0.412	0.00018	-
	Release 2	122.901	0.188	0.199	0.048	0.0231	-
	Release 3	64.365	0.172	0.183	0.001	0.001	-
	Release 4	42.739	0.164	0.162	0.013	0.017	-
SRGM 3	Release 1	103.741	0.139	0.127	0.537	0.002	1.349
	Release 2	121.93	0.412	0.489	0.0683	0.0199	1.624
	Release 3	62.421	0.248	0.241	0.092	0.083	2.103
	Release 4	40.784	0.153	0.207	0.011	0.009	2.321

Table 3 Performance criteria values

Models	Releases	R^2	MSE	MAE	Bias	Variance	RMPSE
SRGM 1	Release 1	0.996	19.065	3.6975	3.1525	7.173029	7.83515
	Release 2	0.998	14.657	2.9936	2.415	5.821	6.3010
	Release 3	0.989	12.767	2.898	2.251	5.524	5.965
	Release 4	0.997	4.8274	1.8168	1.555	3.5721	3.89627
SRGM 2	Release 1	0.995	8.83413	2.583	1.847	4.4802	4.8459
	Release 2	0.995	10.178	2.1105	0.912	3.658	3.7702
	Release 3	0.998	8.34573	2.53	2.22	5.023	5.49
	Release 4	0.998	2.355	1.18978	− 0.95	2.3118	2.44993
SRGM 3	Release 1	0.998	4.74613	1.514	0.247	2.27778	2.2911
	Release 2	0.997	9.9975	2.3221	− 0.6521	3.467	3.5343
	Release 3	0.999	4.6515	1.7166	0.6683	2.556	2.642
	Release 4	0.998	1.46713	0.95473	− 0.262	1.3289	1.354

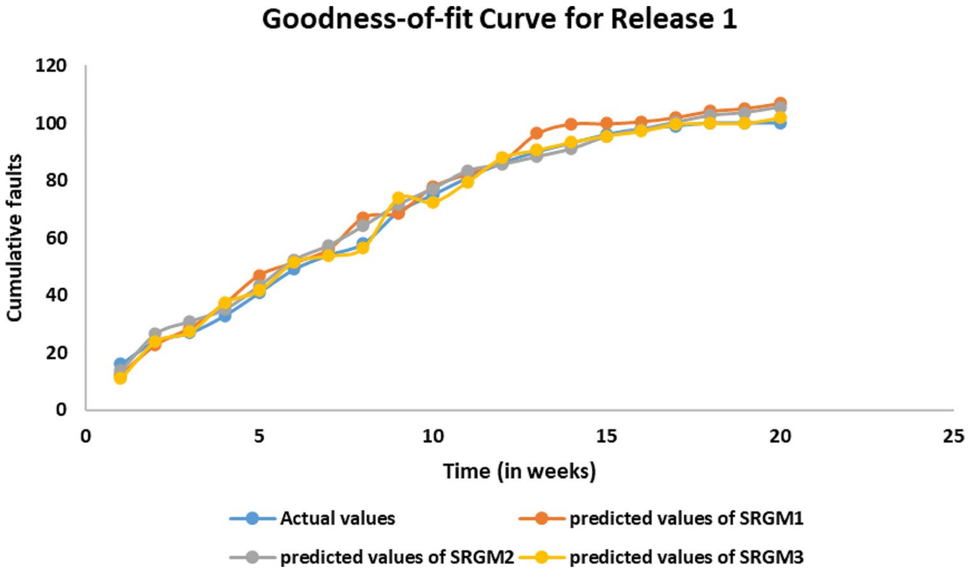

Fig. 2 Goodness of fit curve for release 1

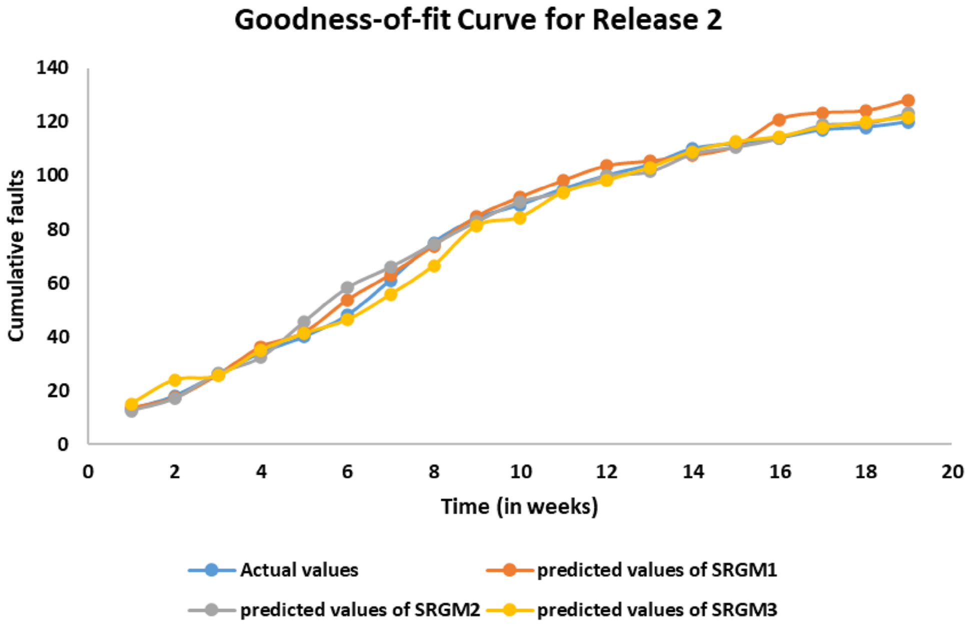

Fig. 3 Goodness of fit curve for release 2

$$m_{4k} = \begin{cases} \begin{bmatrix} \frac{a_{4k}}{1-\alpha_{4k}} + \frac{a_{3k}}{1-\alpha_{3k}}\left(1 - F_{3k}\left(\tau_3\right)\right) \\ +\frac{a_{2k}}{1-\alpha_{2k}}\left(1 - F_{2k}\left(\tau_2\right)\right)\left(1 - F_{3k}\left(\tau_3\right)\right) \\ +\frac{a_{1k}}{1-\alpha_{1k}}\left(1 - F_{3k}\left(\tau_3\right)\right)\left(1 - F_{2k}\left(\tau_2\right)\right)\left(1 - F_{1k}\left(\tau_1\right)\right) \end{bmatrix} F_{4k}(t-\tau_3) \quad ; t_3 \le t \le \tau_4 \\ \begin{bmatrix} \frac{a_{4k}}{1-\alpha_{4k}} + \frac{a_{3k}}{1-\alpha}\left(1 - G_{3k}\left(t_3\right)\right) \\ +\frac{a_{2k}}{1-\alpha_{2k}}\left(1 - G_{2k}\left(t_2\right)\right)\left(1 - G_{3k}\left(t_3\right)\right) \\ +\frac{a_{1k}}{1-\alpha_{1k}}\left(1 - G_{3k}\left(t_3\right)\right)\left(1 - G_{2k}\left(t_2\right)\right)\left(1 - G_{1k}\left(t_1\right)\right) \end{bmatrix} G_{4k}(t-t_3) \quad ; \tau_4 \ < t \ < t_4 \end{cases} \tag{19}$$

4 Numerical illustration with results

The numerical illustration and the results of the proposed models have been provided in this section**:**

4.1 Estimation and performance criteria

The proposed three models are validated using a multi-release fault dataset from Tandem Computers (Wood 1996) for four subsequent releases. The description of the dataset is provided in Table 1. Corresponding to each release, the real data curve was created to identify any kinks that are merely change points. On observation, we concluded that the change points (τ) were observed in the 8th, 26th, 42nd, and 56th week of the four releases respectively. We have proposed three models using three different TCFs namely, Exponential, Delayed S-shaped, and Weibull TCF. For the estimation let's assume $\alpha_1 = \alpha_2 = \alpha$, and the parameter estimation values are provided in Table 2.

After, the estimation of parameters using SPSS software for the proposed three models, we compare the models by using performance criteria. The criteria such as R^2, Mean Square Error (MSE), Mean Absolute Error (Otoom, Al-Shdaifat, Hammad, & Abdallah), Bias, Variance, and Root Mean Percentage Square Error (RMPSE) have been evaluated. The values for the criteria for the three proposed models have been provided in Table 3. The R^2 is highest for release 3 of the SRGM 3 and lowest for the release of the SRGM 1. The R^2 of the model SRGM 3

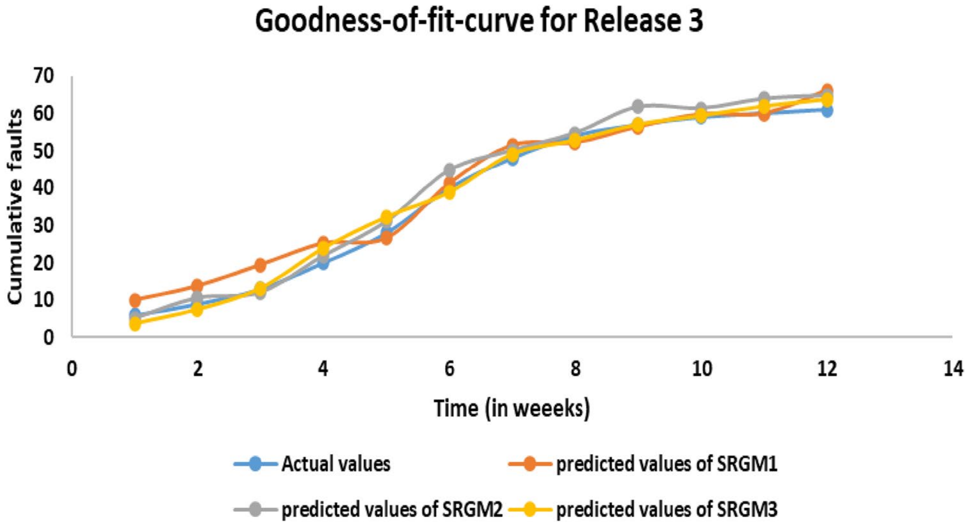

Fig. 4 Goodness of fit curve for release 3

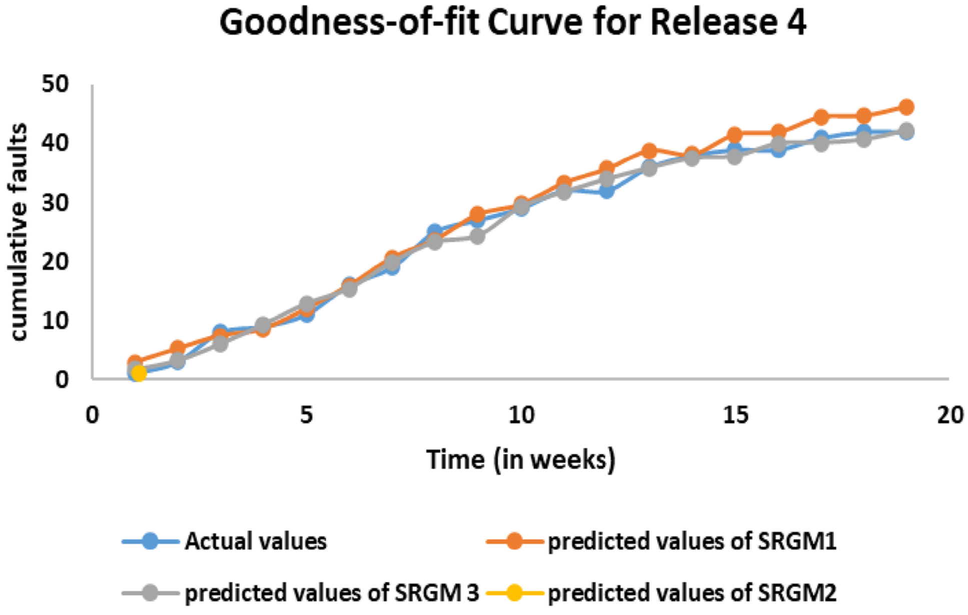

Fig. 5 Goodness of fit curve release 4

with four releases is comparatively high as compared to other proposed models. By this, it can be suggested that the prediction of the percentage of variance in actual faults is significantly good for SRGM 3, in which we have used Weibull TCF. The performance criteria values are better for the SRGM 3 as compared to other models.

4.2 Goodness-of-fit (GoF)

Now, by using the actual and estimated values of the faults, the GoF curve is plotted for the three proposed models as shown in Figs. 2, 3, 4, 5 corresponding to each release. The testing time in weeks is presented on the x-axis, while cumulative faults are on the y-axis. It is said that if the distance is minimum between the actual and estimated points, then it indicates good fitting. By plotting the curves, we have attempted to show how accurately the model estimation has been done. In our research, all three models have efficiently shown good fitting as the estimated values are near the actual values.

The novelty of capturing irregular fluctuations in the fault detection rate lies in its ability to improve sensitivity, adaptability, early fault detection, uncovering unforeseen patterns, and enhancing fault diagnosis. By going beyond traditional approaches, this method offers unique advantages for fault detection and can contribute to more reliable and efficient systems across various domains.

5 Conclusions

The proposed SRGMs are the generalized model with the combination of the idea of SDE-based modeling, testing coverage-based fault observation rate, error generation, and change points. We have presented error generation with the three TCF namely, Exponential, Delayed S-shaped, and Weibull. The fault content observed during the testing procedure is proportional to the size of the code of the software. Whereas, the fault content removed or corrected in the process of debugging is inversely proportional to the size of the code of the software. This nature of fault observation/removal rate is studied by considering the fault detection process as a stochastic process. Later, the multi-release model has been proposed for the four consequent releases of the software. The failure dataset given by Wood (1996) for four releases has been used. The results presented in Sect. 5 of the paper reveal significant estimation values of all three models. On comparing the three models, it is observed that the Weibull testing coverage-based SRGM has performed significantly well. To estimate the accuracy of the model, various performance criteria have been evaluated such as R^2, MSE, MAE, Bias, Variance, and RMPSE. These criteria depict that the Weibull TCF-based SRGM fits the data appropriately and is showing better results as compared to other proposed models. The results are deeply analyzed to reach to following conclusions:

For a software firm, the release of the software is a significant part of the software evolution cycle. So, the testers test the software code for a longer time to disclose the faults perfectly. Here, because of the impact of error generation in the first release, the upcoming version has some residual faults along with new faults due to updates in code. The residual faults from the previous release should be considered on a prior basis as they can be exploited easily by users because they are uncovered in the market. So, while the testing procedure, fixing them becomes highly important. The firm cannot take any risk with these faults, as if they fail to disclose these faults even after consequent testing might result in a negative impact on the market. The third and fourth release shows similar phenomena, but this time testers are now accustomed to the software. Hence, the fault removal process becomes faster.

The study implies that capturing irregular fluctuations in the fault detection rate can provide valuable insights into system stability, fault tolerance, external factors, and overall performance. By understanding the underlying causes of these fluctuations and implementing appropriate measures, organizations can enhance system reliability, optimize resource allocation, and improve fault detection and response processes. Irregular fluctuations in the fault detection rate can pose challenges in fault diagnosis and root cause analysis. When the fault detection rate deviates from expected patterns without apparent reasons, it may be difficult to pinpoint the specific cause or trigger of the fluctuations. This has been considered a limitation as it becomes harder to identify and address underlying issues, potentially leading to longer resolution times and increased downtime.

6 Future work

The proposed stochastic differential-based SRGM opens up many prospects for the extension.

- The release time optimization model can be formulated to increase the reliability of the model and decrease the overall software development cost.
- The extension can be done for different types of faults based on the severity of the software.
- More environmental impact factors can be considered to generate SRGM.

Funding All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript. No funding was received to assist with the preparation of this manuscript.

Declarations

Conflicts of interest There are no potential conflicts of interest.

Ethical approval This is an observational study. The study does not require any ethical approval.

References

Aggarwal AG, Gandhi N, Verma V, Tandon A (2019) Multi-release software reliability growth assessment: an approach incorporating fault reduction factor and imperfect debugging. Int J Math Oper Res 15(4):446–463

Aggarwal AG, Kapur P, Nijhawan N (2018) A discrete SRGM for multi-release software system with faults of different severity. Int J Oper Res 32(2):156–168

Aggarwal AG, Nijhawan N, Kapur P (2015) A discrete SRGM for multi-release software system with imperfect debugging and related optimal release policy. Paper presented at the 2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)

Ahmad N, Khan MG, Rafi LS (2010) A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging. Int J Quality Reliab Manag 27(1):89–110

Anand S, Verma V, Aggarwal AG (2018) 2-Dimensional multi-release software reliability modelling considering fault reduction factor under imperfect debugging. Ingeniería Solidaria 14(25):1–12

Bevia V, Calatayud J, Cortés J-C, Jornet M (2023) On the generalized logistic random differential equation: theoretical analysis and numerical simulations with real-world data. Commun Nonlinear Sci Numer Simul 116:106832

Bibyan R, Anand S, Aggarwal AG, Kaur G (2023) Multi-release software model based on testing coverage incorporating random effect (SDE). MethodsX 10:102076

Chang Y-P (2001) Estimation of parameters for nonhomogeneous Poisson process: software reliability with change-point model. Commun Stat Simul Comput 30(3):623–635

Dhaka V Nijhawan N (2022) Effect of change in environment on reliability growth modeling integrating fault reduction factor and change point: a general approach. Ann Oper Res 1–35

Goel AL (1985) Software reliability models: assumptions, limitations, and applicability. IEEE Trans Softw Eng 12:1411–1423

Goel AL, Okumoto K (1979) Time-dependent error-detection rate model for software reliability and other performance measures. IEEE Trans Reliab 28(3):206–211

Gokhale SS, Philip T, Marinos PN, Trivedi KS (1996) Unification of finite failure non-homogeneous Poisson process models through test coverage. Paper presented at the Proceedings of ISSRE'96: 7th international symposium on software reliability engineering

Huang C-Y, Lin C-T (2006) Software reliability analysis by considering fault dependency and debugging time lag. IEEE Trans Reliab 55(3):436–450

Hwang S, Pham H (2008) Quasi-renewal time-delay fault-removal consideration in software reliability modeling. IEEE Trans Syst Man Cybern a Syst Humans 39(1):200–209

Kapur P, Aggarwal AG, Garmabaki AS (2011a) Imperfect debugging software reliability growth model for multiple releases

Kapur P, Anand S, Yadavalli V, Beichelt F (2007a) A generalised software growth model using stochastic differential equation. Communication in dependability and quality management belgrade, Serbia, 34

Kapur P, Anand S, Yamada S, Yadavalli VS (2009). Stochastic differential equation-based flexible software reliability growth model. Math Probl Eng

Kapur P, Garg R, Aggarwal AG, Tandon A (2010) Two dimensional flexible software reliability growth model and related release policy. Paper presented at the proceedings of the 4th national conference, INDIACom-2010, New Delhi, India

Kapur P, Gupta A, Jha P (2006) On modeling failure phenomenon of multiple release of a software in operational phase for product and project type software: a theoretical approach. Paper presented at the Proceedings of international conference on quality, reliability and infocom technology

Kapur P, Gupta D, Gupta A, Jha P (2008a) Effect of introduction of fault and imperfect debugging on release time. Ratio Mathematica 18(1):62–90

Kapur P, Kumar A, Yadav K, Khatri SK (2007b) Software reliability growth modelling for errors of different severity using change point. Int J Reliab Qual Saf Eng 14(04):311–326

Kapur P, Pham H, Aggarwal AG, Kaur G (2012) Two dimensional multi-release software reliability modeling and optimal release planning. IEEE Trans Reliab 61(3):758–768

Kapur P, Pham H, Anand S, Yadav K (2011b) A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation. IEEE Trans Reliab 60(1):331–340

Kapur P, Sachdeva N, Singh JN (2014) Optimal cost: a criterion to release multiple versions of software. Int J Syst Assur Eng Manag 5(2):174–180

Kapur P, Singh V, Anand S, Yadavalli V (2008b) Software reliability growth model with change-point and effort control using a power function of the testing time. Int J Prod Res 46(3):771–787

Kareer N, Kapur P, Grover P (1990) An S-shaped software reliability growth model with two types of errors. Microelectron Reliab 30(6):1085–1090

Lee CH, Kim YT, Park DH (2004) S-shaped software reliability growth models derived from stochastic differential equations. IIE Trans 36(12):1193–1199

Li Q, Pham H (2017) NHPP software reliability model considering the uncertainty of operating environments with imperfect debugging and testing coverage. Appl Math Model 51:68–85

Lyu MR (1996) Handbook of software reliability engineering, vol. 222. IEEE computer society press Los Alamitos

Malaiya YK, Li MN, Bieman JM, Karcich R (2002) Software reliability growth with test coverage. IEEE Trans Reliab 51(4):420–426

Malaiya YK, Li N, Bieman J, Karcich R, & Skibbe B (1994) The relationship between test coverage and reliability. Paper presented at the proceedings of 1994 IEEE international symposium on software reliability engineering

Mishra G, Kapur P, Aggarwal AG (2023) A generalized multi-upgradation SRGM considering uncertainty of random field operating environments. International Journal of System Assurance Engineering and Management, pp. 1-9

Nijhawan N, Aggarwal AG (2015) On development of change point based generalized SRGM for software with multiple releases. Paper presented at the 2015 4th international conference on reliability, infocom technologies and optimization (ICRITO) (trends and future directions).

Ohba M (1984a) Inflection S-shaped software reliability growth model. In: Stochastic models in reliability theory. Springer, pp 144–162

Ohba M (1984b) Software reliability analysis models. IBM J Res Dev 28(4):428–443

Ohba M, Chou X-M (1989) Does imperfect debugging affect software reliability growth? Paper presented at the Proceedings of the 11th international conference on software engineering

Øksendal B (2003) Stochastic differential equations. In: Stochastic differential equations. Springer, pp 65–84

Otoom AF, Al-Shdaifat D, Hammad M, Abdallah EE (2016) Severity prediction of software bugs. Paper presented at the 2016 7th international conference on information and communication systems (ICICS).

Pham H (1993) Software reliability assessment: imperfect debugging and multiple failure types in software development. EGandG-RAAM-10737, Idaho National Engineering Laboratory

Pham H (2000) Software reliability. Springer, Berlin

Pham H (2014) Loglog fault-detection rate and testing coverage software reliability models subject to random environments. Vietnam J Comput Sci 1(1):39–45

Pham H, Nordmann L, Zhang Z (1999) A general imperfect-software-debugging model with S-shaped fault-detection rate. IEEE Trans Reliab 48(2):169–175

Pham H, Zhang X (1997) An NHPP software reliability model and its comparison. Int J Reliab Qual Saf Eng 4(03):269–282

Pham H, Zhang X (2003) NHPP software reliability and cost models with testing coverage. Eur J Oper Res 145(2):443–454

Pham T, Pham H (2019) A generalized software reliability model with stochastic fault-detection rate. Ann Oper Res 277:83–93

Shigeru Y, Akio N (2003) A stochastic differential equation model for software reliability assessment and its goodness-of-fit. Int J Reliab Appl 4(1):1–12

Tamura Y, Yamada S (2006) A flexible stochastic differential equation model in distributed development environment. Eur J Oper Res 168(1):143–152

Tandon A, Aggarwal AG (2020) Testing coverage based reliability modeling for multi-release open-source software incorporating fault reduction factor. Life Cycle Reliabil Saf Eng 9(4):425–435

Tandon A, Aggarwal AG, Nijhawan N (2016) An NHPP SRGM with change point and multiple releases. International Journal of Information Systems in the Service Sector (IJISSS) 8(4):57–68

Verma V, Anand S, Kapur P, Aggarwal AG (2022) Unified framework to assess software reliability and determine optimal release time in presence of fault reduction factor, error generation and fault removal efficiency. Int J Syst Assur Eng Manag 13(5):2429–2441

Wood A (1996) Predicting software reliability. Computer 29(11):69–77

Xie M, Hu Q, Wu Y, Ng SH (2007) A study of the modeling and analysis of software fault-detection and fault-correction processes. Qual Reliab Eng Int 23(4):459–470

Yamada S, Kimura M, Tanaka H, Osaki S (1994) Software reliability measurement and assessment with stochastic differential equations. IEICE Trans Fundam Electron Commun Comput Sci 77(1):109–116

Yamada S, Ohba M, Osaki S (1983) S-shaped reliability growth modeling for software error detection. IEEE Trans Reliab 32(5):475–484

Yamada S, Ohba M, Osaki S (1984) S-shaped software reliability growth models and their applications. IEEE Trans Reliab 33(4):289–292

Yamada S, Ohtera H, Ohba M (1992a) Testing-domain dependent software reliability models. Comput Math Appl 24(1–2):79–86

Yamada S, Tokuno K, Osaki S (1992b) Imperfect debugging models with fault introduction rate for software reliability assessment. Int J Syst Sci 23(12):2241–2252

Zhang C, Cui G, Liu H, Meng F, Wu S (2014) A unified and flexible framework of imperfect debugging dependent SRGMs with testing-effort. J Multimed 9(2)

Zhao J, Liu H-W, Cui G, Yang X-Z (2006) Software reliability growth model with change-point and environmental function. J Syst Softw 79(11):1578–1587

Zhao M (1993) Change-point problems in software and hardware reliability. Commun Stat Theory Methods 22(3):757–768

Zhu M, Pham H (2022) A generalized multiple environmental factors software reliability model with stochastic fault detection process. Ann Oper Res 1–22

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.