Arab J Sci Eng (2016) 41:5009–5025
DOI 10.1007/s13369-016-2189-0

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Modeling and Analysis of Software Fault Detection and Correction Process Through Weibull-Type Fault Reduction Factor, Change Point and Imperfect Debugging

Subhashis Chatterjee[1] · **Ankur Shukla**[1]

Received: 22 June 2015 / Accepted: 11 May 2016 / Published online: 25 May 2016

Abstract Fault reduction factor (FRF) is one of the most important factors which plays a vital role in software reliability growth. In the past, few studies on the influence of different environmental factors into FRF have been carried out. In these studies, FRF has been defined using some particular functions such as constant, increasing, decreasing and inflection S-shaped. These functions may not be realistic and reasonable to represent the actual behavior of FRF. Therefore, in this study, it has been tried to represent the realistic behavior of FRF using Weibull curve. Moreover, a new approach of software reliability modeling has been proposed in which FRF has been incorporated in fault detection and correction process. Thus, in this paper, a general frame work of software reliability growth model (SRGM) has been proposed considering the fault detection and correction process. The concepts of imperfect debugging and change point have also been incorporated in the present study. Different parameters of the proposed SRGM are estimated using the SPSS and 'R' software. Different comparison criteria have been used for comparison of the proposed SRGM with other existing SRGMs. Chi-square goodness-of-fit test has been used for validation of the proposed SRGM.

Keywords Software reliability · Non-homogeneous Poisson process · Fault reduction factor · Imperfect debugging · Change point · Sensitivity analysis

✉ Subhashis Chatterjee
chatterjee_subhashis@rediffmail.com

1 Department of Applied Mathematics, Indian School of Mines, Dhanbad 826004, India

1 Introduction

With rapid and continuous changes in the area of computer technology, a revolutionary change has been taken place in software development process. Nowadays software and software-driven systems are used in various area starting from simple data processing and finance to real-time control system. Due to this reason in today's automated world, the modern society has become more software dependent. Consequently, the demand of high-quality software is increasing day by day. As the software development process has become more complex, producing quality software and its maintenance has become a challenging task to the software engineers. The important characteristics of a quality software are reliability, security, safety, compatibility, performability, etc. Reliability is the most dynamic characteristics of a quality software, since it quantifies the software faults during software development process [1–3]. Software reliability is defined as the failure-free operation of software under specified time and environment [4]. It is a difficult task to asses the reliability of the software being released. The models used to asses the reliability of software quantitatively are known as SRGMs. SRGMs help to make decision when to stop the testing to achieve desired level of reliability. It is also used to estimate the remaining number of faults, failure intensity, initial faults and other decision-making processes like cost and release time estimation. In the past few decades, numerous studies have been carried out by researchers to assure the quality of software [5–15].

There are many factors that affect the reliability growth of software. FRF is one of the key factors which profiles the software development process, first proposed by Musa [4]. Therefore, it has much influence on the reliability growth of software. FRF is defined as the net number of faults removed in proportion to the failures experienced [4,16]. In the generalization of basic execution time model, Musa [4] assumed that FRF may affect the fault detection and correction process and defined it as a proportionality constant. Musa considered the FRF as a constant. In reality, FRF can be influenced by different factors such as imperfect debugging, resource allocation, fault dependency and environment. Considering this fact, some researchers have discussed the impact of these factors on FRF and defined different characteristics of FRF. In this sequence, Hsu et al. [17] considered the effect of environmental factor on FRF and defined three patterns of FRF that are constant, increasing and decreasing. Later, Pachauri et al. [18] redefined the Hsu et al. SRGM and defined FRF as inflection S-shaped function. In the above-mentioned articles, authors have considered particular nature of FRF, which is not realistic. In reality, FRF has no single or definite pattern for different data sets as such; it solely depends on different environmental factors. Therefore, it is essential to consider the actual behavior of FRF. Hence, Weibull-type FRF has been considered in the present study. The advantage of considering Weibull-type FRF can be assumed that by varying the parameters one can obtain an appropriate FRF for a particular failure data. The detail study about FRF and formulation of Weibull-type FRF has been done in Sects. 2.1.1, 2.2 and 3.1.

During the fault removal process, a considerable time is consumed for fixing and removing of the detected faults. This happens because of different reasons such as complexity of detected faults, skill of testing team, testing effort and testing environment. Therefore, the time delay between fault detection and correction process cannot be avoided. Previously, some researchers have made effort to model the fault detection and correction process [1,19–23]. In these SRGMs, various functions, such as constant, linear and exponential, have been used to denote the time lag between fault detection and correction process [1,21,23], which is not realistic since these time lag functions have been formulated based on the authors assumptions and valid for only specific conditions. To overcome this problem, a different approach of modeling has been proposed in the proposed study. In this study, the relationship between fault detection and correction process has been defined by FRF. Since FRF is the average ratio of the rate of reduction of faults to the rate of failure occurrence, therefore it can be represented as a function of detected and corrected faults [16].

It is often observed that the fault occurrence phenomenon in software is not always the same during entire testing process. At different time points during the entire testing phase, it changes due to variations in testing strategy, testing environment, testing effort, resource allocation, team constitution, etc. These points are known as change point [24]. Therefore, due to change point the number of detected and corrected faults changes during testing. In past, some researchers discussed the importance of change point in software reliability growth modeling [1,25–33]. Hence, it is appropriate to consider change point in fault detection and correction modeling.

During the debugging process, there is a possibility of introducing new faults when detected faults are removed. This phenomenon is known as imperfect debugging. This was first proposed by Goel [34] incorporating the probability of imperfect debugging in Jelinski and Moranda model [35]. In past, some researchers have been proposed different SRGMs considering imperfect debugging phenomenon [1,14,36–40]. Imperfect debugging phenomenon has an effect on the number of faults detected, corrected and total number of faults. Consequently, FRF will be affected also. Therefore, imperfect debugging has been incorporated in the proposed study.

The objective of this paper is to develop a general frame for modeling the fault detection and correction process which incorporates the actual behavior of FRF. In this paper, FRF has been represented by Weibull curve which provides more flexible and realistic behavior of FRF, while it is considered as a particular behavior in recent studies. Moreover, a different prospective of software reliability growth model has been proposed by representing the dependency between fault detection and correction process through FRF. As discussed above, imperfect debugging and change point are essential part of a SRGM. Therefore, these factors have been incorporated in the proposed SRGM. Some numerical examples using real software failure data have been discussed to validate and illustrate the applications of the proposed SRGM.

Rest of the article is organized as follows: a description of basic concepts and related work about FRF- and NHPP-based SRGMs has been presented in Sect. 2. The detail development of the proposed SRGM has been presented in Sect. 3. Comparison and analysis of the performance of the proposed SRGM with some well-known existing SRGMs have been presented in Sect. 4. Sensitivity analysis has been carried out in Sect. 5. Finally, Sect. 6 presents conclusion.

2 Basic Concepts and Related Work

This section presents the basic concepts and related works about FRF, NHPP-based SRGMs and fault detection and correction process.

2.1 Basic Concepts

2.1.1 Fault Reduction Factor

The net number of faults removed is only a portion of the failure experienced, expressed by fault reduction factor B. In other words, it is the ratio of net fault reduction to failures experienced as time of operation approaches infinity.

Expected values are taken for both the faults and failures. If m is the number of failure experienced in correcting n faults, then B can be defined as [4,16]:

$$B = n/m \tag{1}$$

It is also defined as:

$$B = \frac{\lambda_0}{Kfm} \tag{2}$$

where λ_0 is the initial failure intensity, K is the fault exposure ratio and f is the linear execution frequency of the program. According to Musa [16], "FRF is usually positive and less than one but it can be negative or zero. It can be greater than one in the case that finding the fault that produced a particular failure causes other faults to be found and fixed as well".

2.1.2 NHPP-Based Software Reliability Growth Modeling

Software reliability modeling based on NHPP is quite popular tool, which is widely used by several researchers in past four decades [1,13–15]. In these SRGMs, cumulative number of failures is represented by a counting process and it is assumed that the counting process follows NHPP. One of the most common assumptions of these SRGMs is that the failure occurrence rate is the proportional to the number of remaining faults in the software which are not detected. The main aim of these SRGMs is to find out the realistic and reasonable expression of mean value function (MVF), which represents the cumulative number of failures experienced up to certain time.

Initially, Schneidewind [42] has made an attempt to model fault detection and correction process based on NHPP. He assumed that faults detected in an interval is independent of the faults detected in other intervals and proportional to the number of faults within the interval. In this SRGM, failure intensity is defined as a exponentially decreasing function of time. Goel–Okumoto [43] assumed that the counting process $\{N(t), t \geqslant 0\}$ which represents the cumulative number of failure follows NHPP with mean value function $m(t)$, i.e.,

$$P[N(t) = n] = \frac{[m(t)]^n}{n!} \exp(-m(t)) \tag{3}$$

where $n = 0, 1, 2, \ldots$.

Also, they have assumed that the failure intensity is proportional to the number of remaining faults, i.e.,

$$\lambda(t) = \frac{\mathrm{d}m(t)}{\mathrm{d}t} = b(a - m(t)) \tag{4}$$

and

$$m(t) = \int_0^t \lambda(t)\mathrm{d}t \tag{5}$$

where a represents the total number of faults present in software before testing, b is the proportionality constant defined as fault detection rate and the mean value function (MVF) $m(t)$ represents the expected number of faults by time t.

This SRGM was an innovative effort made in NHPP-based software reliability growth modeling. It is also very popular and conventional SRGM. Many NHPP-based SRGMs have been proposed with various modifications of the assumptions of Goel–Okumoto model [1,2,13–15,41]. In this article, a NHPP-based SRGM has been proposed considering fault detection and correction modeling technique.

2.1.3 Fault Detection and Correction Process

During the testing process, possible fault sites are sensitized to detect the faults. Several conventional SRGMs are developed based on one common assumption that detected faults are removed immediately, which may not be realistic or reasonable. In reality, fault correction process is a complex and time-consuming process. Fault correction process goes through the different stages such as fault detection, fault localization, fault reporting, fault isolation, fault correction and verification. So it is important to consider time lag between fault detection and correction process in software reliability growth modeling. The time lag mainly depends on fault complexity, testing skill, testing effort, testing environment, etc.

Schneidewind [42] first modeled the fault detection and correction process assuming the possibility of time lag between fault detection and correction process. If fault detection process follows the NHPP with failure intensity $\lambda_{\mathrm{d}}(t)$, then MVF $m_{\mathrm{d}}(t)$ of the detected number of faults is:

$$m_{\mathrm{d}}(t) = \int_0^t \lambda_{\mathrm{d}}(s)\mathrm{d}s \tag{6}$$

It is assumed that the rate of fault correction is proportional to the number of remaining uncorrected faults. After fault detection, if the correction of fault is delayed by time Δt, then expected number of faults corrected during time Δt will be $(m_{\mathrm{d}}(t) - m_{\mathrm{c}}(t))$. Thus, the MVF of corrected number of faults can be obtained from $m_{\mathrm{d}}(t)$ as follows:

$$m_{\mathrm{c}}(t) = m_{\mathrm{d}}(t - \Delta t) \tag{7}$$

Schneidewind [42] assumed that all detected faults are corrected with equal time delay, which is not reasonable. Since in the beginning of testing process, it is easy to correct the detected faults, while it becomes more difficult at the end of testing process as the complexity of faults increases. Considering this fact, Xie and Zhao [44] extended the Schneidewind [42] model and defined the time delay as a increasing function of time. Moreover, they [44] presented another approach to model the fault detection and correction process in which both the fault detection and correction processes were modeled separately. In this SRGM [44], it is considered that the fault detection rate is proportional to the number of remaining number of undetected faults, and the fault correction rate is proportional to the remaining number of uncorrected faults. The MVF of detected number of faults ($m_{\mathrm{d}}(t)$) and corrected number of faults ($m_{\mathrm{c}}(t)$) can be obtained from following differential equations:

$$\frac{\mathrm{d}m_{\mathrm{d}}(t)}{\mathrm{d}t} = b(t)(a - m_{\mathrm{d}}(t)) \tag{8}$$

$$\frac{\mathrm{d}m_{\mathrm{c}}(t)}{\mathrm{d}t} = c(t)(m_{\mathrm{d}}(t) - m_{\mathrm{c}}(t)) \tag{9}$$

where $b(t)$ is the fault detection rate and $c(t)$ is the fault correction rate.

These two SRGMs proposed in [42,44] are the key models in the area of fault detection correction process modeling. Previously, some studies have been carried out to show the importance of fault detection and correction process in software reliability growth modeling which are mainly extension or modification of these SRGMs [42,44]. In these studies, researchers mainly considered the various possible time function for the fault detection and correction rate, while some of them considered the different patterns of time delay function [1,2,19,21–23,45].

2.2 Related Work

2.2.1 Musa Basic Execution Time Model

Musa [15,16] proposed basic execution time model by assuming that the failure rate function at the execution time is proportional to the number of remaining faults in the software, which is represented by the following differential equation:

$$\lambda(t) = \frac{\mathrm{d}m(t)}{\mathrm{d}t} = z(t) = \phi(a - m(t)) \tag{10}$$

where $\lambda(t)$ is the failure rate function at time t, ϕ is the per fault hazard rate, $m(t)$ represents cumulative number of failures at execution time t and $z(t)$ is the hazard rate function.

Musa [15,16] generalized the basic execution time model by defining a fault reduction factor B to show the relationship between faults and failure. Now, the basic execution model becomes:

$$\lambda(t) = \frac{\mathrm{d}m(t)}{\mathrm{d}t} = Bz(t) = B\phi(a - m(t)) \tag{11}$$

MVF can be obtained by solving Eq. (11) with initial conditions $m(0) = 0$ as follows:

$$m(t) = a\left(1 - \mathrm{e}^{-B\phi t}\right) \tag{12}$$

With assumptions of Musa's generalized basic execution model, some SRGMs are proposed considering different behaviors of FRF [17,18].

2.2.2 Hsu et al. SRGM

As shown in Musa basic execution model, FRF has been defined as a constant and less than 1. Hsu et al. [17] observed that FRF can be influenced by different environmental factor such as imperfect debugging, resource allocation, fault dependency and environment. Therefore, FRF cannot be constant in all cases. As the learning process increases, the impact of environmental factor on FRF decreases, and hence, FRF increases. On other hand, as the impact of environmental factor increases continuously, FRF deceases. Considering these facts, Hsu et al. [17] redefined the generalized form of the Musa basic execution time model by considering the time-dependent FRF, which is defined by the following differential equations:

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t} = r(t)\,(a - m(t)) \tag{13}$$

and

$$r(t) = r \times B(t) \tag{14}$$

where $r(t)$ is the fault detection rate and $B(t)$ is FRF. They studied the behavior of FRF and concluded that FRF has three patterns, which are:

(i) Constant FRF, i.e., $B(t) = B$ with $0 < B \leqslant 1$, where B is constant FRF.
(ii) Increasing FRF curve, i.e., $B(t) = 1 - (1 - B_0)\mathrm{e}^{-kt}$.
(iii) Decreasing FRF curve, i.e., $B(t) = B_0\mathrm{e}^{-kt}$.

where B_0 is the initial FRF with $B_0 \leqslant B(t) \leqslant 1$ and k is the constant parameter, $0 \leqslant k \leqslant 1$. By solving Eqs. (13) and (14) with initial condition $m(0) = 0$, MVFs of the detected number of faults are given as follows:

(i) For constant FRF, $m(t) = a\left(1 - \mathrm{e}^{-Brt}\right)$.

(ii) For increasing FRF curve, i.e.,

$$m(t) = a\left(1 - e^{-r\left(\frac{(B_0-1)(1-e^{-kt})}{k}+t\right)}\right).$$

(iii) For decreasing FRF curve,

$$m(t) = a\left(1 - e^{-r\left(\frac{B_0(1-e^{-kt})}{k}\right)}\right).$$

2.2.3 Pachauri et al. SRGM

As discussed previously, Hsu et al. [17] defined the three patterns of FRF for single release software. Pachauri et al. [18] observed that FRF may not follow the similar pattern for multi release software and defined FRF as an inflection S-shaped curve, which is:

$$B(t) = \frac{\alpha}{1+\beta e^{-\alpha t}}, \tag{15}$$

where α and β are the shape and scale parameters. MVF of the detected faults can be obtained by solving the Eqs. (13)–(15) simultaneously with initial condition $m(0) = 0$, which is given as follows:

$$m(t) = a\left(1 - \left(\frac{1+\beta}{1+\beta e^{-\alpha t}}\right)^r e^{-\alpha r t}\right). \tag{16}$$

Moreover, Pachauri et al. [18] extended their proposed model by considering the possibility of introduction of new faults during imperfect debugging process., i.e.,

$$\frac{da(t)}{dt} = \gamma\frac{dm(t)}{dt} \tag{17}$$

where $a(t)$ is the fault content function and γ is the fault introduction rate. MVF can be obtained by solving Eqs. (13), (14) and (17) simultaneously with Eq. (15) as follows:

$$m(t) = \frac{a}{1-\gamma}\left(1 - \left(\frac{1+\beta}{1+\beta e^{-\alpha t}}\right)^{r(1-\gamma)} e^{-\alpha r(1-\gamma)t}\right). \tag{18}$$

2.3 Motivation of the Proposed SRGM

FRF can be influenced by different environmental factors such as testing effort, testing environment, tester skill, testing coverage and imperfect debugging. Previously, some researchers defined the nature of FRF using various pattern such as constant, exponentially increasing curve, exponentially decreasing curve and S-shaped curve [16–18]. However, these curves are not suitable for better representation of the realistic behavior of FRF, because it may not be smooth and follow a particular pattern. Therefore, it is very important to formulate the realistic behavior of FRF for accurate reliability estimation and prediction. In the proposed study, an attempt has been made to model the exact behavior of FRF using Weibull curve.

As discussed in the previous Sect. 2.1.3, different time lag functions have been used to develop the relationship between fault detection and correction process. In reality, these functions are considered based on the assumptions of the authors and only true for some specific conditions. Therefore, these function are not sufficient to develop the dependency between fault detection and correction process. In this regard, a new methodology has been proposed to develop the dependency between fault detection and correction process through FRF in this paper, and both the fault detection and correction process are modeled separately. Moreover, reliability growth of the software can be influenced by different factor such as imperfect debugging and change point and it is necessary to consider the effect these factors in a SRGM [1,14,15]. Therefore, the concept of imperfect debugging and change point has been incorporated in the proposed SRGM to improve the accuracy and flexibility.

3 Model Development

This section presents the development of the proposed SRGM in detail and systematic manner. Also, the effect of FRF, time-dependent fault introduction and detection rate along with change point on the proposed SRGM has been discussed following subsections in detail.

3.1 Weibull FRF

In software reliability modeling, trustworthiness of data sets is necessary [46,47]. In SRGMs, different real software failure data sets have been used to estimate and predict the reliability growth of software. As discussed previously, FRF curves defined in [16–18] may not always the best selection for a particular data set. To overcome this situation, first FRF for a particular data set has been estimated using Weibull curve in the proposed study, and then, the expected curve of FRF has been incorporated in the proposed SRGM. FRF $B(t)$ can be represented as follows:

$$\begin{aligned} B(t) = w(t) &= \frac{dW(t)}{dt} \\ &= Nm\,\eta\,t^{m-1}\exp(-\eta t^m),\ N>0,\ m>0,\ \eta>0 \end{aligned} \tag{19}$$

where $w(t)$ is the probability distribution function and $W(t) = N(1 - \exp(-\eta t^m))$ is the cumulative distribution function for Weibull distribution. N is constant, η is the scale parameter and m is the shape parameter.

Weibull distribution is the well-known and widely used distribution for modeling reliability data due to its flexibility and versatility [1]. Varying the different parameters, it is possible to derive various patterns like

(a) *Exponential Curve:* When $m = 1$, Weibull curve represents the exponential curve. Hence, FRF can be defined as follows:

$$B(t) = w(t) = \frac{\mathrm{d}W(t)}{\mathrm{d}t} = N\eta \exp(-\eta t), \quad N > 0, \ \eta > 0 \tag{20}$$

(b) *Rayleigh Curve:* When $m = 2$, Weibull curve represents the Rayleigh curve. Hence, FRF can be defined as follows:

$$B(t) = w(t) = \frac{\mathrm{d}W(t)}{\mathrm{d}t} = 2N\eta\, t \exp(-\eta t^2), \quad N > 0, \ \eta > 0 \tag{21}$$

This shows that the Weibull curve is the best possible selection for modeling the time varying FRF, as one can get suitable pattern of FRF for a particular data set by varying the parameters.

3.2 Dependency Between Fault Detection and Correction Process

As discussed previously, most of the SRGMs have considered that the detected faults are removed immediately and perfectly [1,14,15]. In practical, fault removal process is a difficult and time-consuming process. Some researchers have made efforts to show the dependency between fault detection and correction process through different time lag functions such as constant, linear and exponential. [1,14,21,23,42], while these functions are considered based on the assumptions made during the development of SRGM and not verified for real testing process. Therefore, these functions are not enough to represent the relationship between fault detection and correction process. Since FRF is defined as the ratio of rate of reduction of faults to the rate of failure occurrence [4], hence fault detection and correction process can be defined by FRF. Therefore, fault detection and correction process can be defined by FRF. In this subsection, a different methodology has been developed to represent the dependency between fault detection and correction process through FRF.

As defined in Sect. 2.1.1, FRF can be represented as:

$$B = n/m \tag{22}$$

where m is the number of failure experienced in correcting n faults.

Since $m_\mathrm{d}(t)$ denotes the MVF of the detected faults, i.e., total number of faults detected at time t and $m_\mathrm{c}(t)$ denotes the MVF of the corrected faults, i.e., total number of faults corrected at time t. Hence, FRF can be redefined as follows:

$$B(t) = \frac{m_\mathrm{c}(t)}{m_\mathrm{d}(t)} \tag{23}$$

For example, let if total 100 faults are detected for a system. During the debugging process, when detected faults are removed, it is possible to introduce new faults. Let 5 faults are introduced due to imperfect debugging in correcting 100 faults. Finally, total number of corrected faults are 100, but the net number of faults is 95. Therefore, FRF would be 0.95.

Equation (23) implies

$$m_\mathrm{c}(t) = B(t)m_\mathrm{d}(t) \tag{24}$$

where $B(t)$ can be best represented by Weibull curve as discussed previously.

3.3 Proposed Model

3.3.1 Assumptions

The assumptions made to develop the proposed SRGM are as follows:

(i) Software failure process follows a NHPP.
(ii) The mean number of faults detected in the time interval $(t + \Delta t)$ is proportional to the remaining number of faults in the system.
(iii) As the learning and maturity of software engineers increase with time, the fault detection rate increases. Therefore, the fault detection rate, $b(t)$, can be better represented as power function of time as follows [1]:

$$b(t) = bt^k \tag{25}$$

where b and k are constants.
(iv) The software debugging process is imperfect. It means, during debugging process when detected faults are removed, it is possible to introduced new faults with fault introduction rate $\beta(t)$. During the testing process, number of faults increases at the beginning due to more faults introduced, and as the learning and maturity of software engineers increase, fault introduction rate decreases later. Therefore, to satisfy this condition the fault introduction rate $\beta(t)$ can be considered as follows:

$$\beta(t) = \beta t(1 - \alpha t) \tag{26}$$

where β and α are constants.

(v) The detected faults are not corrected immediately and perfectly.

(vi) The dependency between fault detection and correction process is represented by FRF, which is a Weibull curve.

3.3.2 Formulation and Solution

Based on the above assumptions, the proposed NHPP-based SRGM can be easily represented by the differential equations derived as follows:

As mentioned in assumption (ii) and (v), the mean number of faults detected in the time interval $(t + \Delta t)$ is proportional to the remaining number of faults in the system, and the detected faults are not removed immediately and perfectly. It means the number of detected faults in interval $(t + \Delta t)$ is proportional to the number of uncorrected faults in that interval. Hence, the MVF of the detected number of faults can be formulated using assumption (i) as follows:

$$\frac{\mathrm{d}m_{\mathrm{d}}(t)}{\mathrm{d}t} = b(t)(a(t) - m_{\mathrm{c}}(t)) \tag{27}$$

where $m_{\mathrm{d}}(t)$ and $m_{\mathrm{c}}(t)$ are the MVF of fault detection and correction process, respectively, and $b(t)$ is the proportionality constant defined as the fault detection rate.

From assumption (iv), expected initial fault content $a(t)$ at time t can be formulated as follows:

$$\frac{\mathrm{d}a(t)}{\mathrm{d}t} = \beta(t)\frac{\mathrm{d}m_{\mathrm{d}}(t)}{\mathrm{d}t} + m_{\mathrm{d}}(t)\frac{\mathrm{d}\beta(t)}{\mathrm{d}t} \tag{28}$$

As discussed in Sect. 3.2 and presented in assumption (vi), $m_{\mathrm{c}}(t)$ can be represented as $m_{\mathrm{c}}(t) = B(t)m_{\mathrm{d}}(t)$. Putting the value of $m_{\mathrm{c}}(t)$ in Eq. (27), the following equation is obtained:

$$\frac{\mathrm{d}m_{\mathrm{d}}(t)}{\mathrm{d}t} = b(t)(a(t) - B(t)m_{\mathrm{d}}(t)) \tag{29}$$

(a) MVF of Fault Detection Process

Using Eqs. (25) and (26), MVF of detected number of faults can be obtained by solving Eqs. (28) and (29) simultaneously with initial conditions $m_{\mathrm{d}}(0) = 0$ and $a(0) = a$ as follows:

$$\begin{aligned} m_{\mathrm{d}}(t) = ab_1\Bigg[&\frac{t^{k_1+1}}{k_1+1} - b_1\beta_1\left\{\frac{t^{(2k_1+3)}}{(2k_1+3)(k_1+2)} - \alpha_1\frac{t^{(2k_1+4)}}{(k_1+3)(2k_1+4)}\right\} \\ &+ b_1 N m\eta\left\{\frac{t^{(m+2k_1+1)}}{(m+k_1)(m+2k_1+1)} - \eta\frac{t^{(2m+2k_1+1)}}{(2m+k_1)(2m+2k_1+1)}\right\}\Bigg] \\ &\times \exp\left[b_1\beta_1\left\{\frac{t^{k_1+2}}{(k_1+2)} - \alpha_1\frac{t^{k_1+3}}{(k_1+3)}\right\} - b_1 N m\eta\left\{\frac{t^{m+k_1}}{(m+k_1)} - \eta\frac{t^{2m+k_1}}{(2m+k_1)}\right\}\right] \end{aligned} \tag{30}$$

(b) MVF of Fault Correction Process

Since FRF represents the relationship between fault detection and correction process. Therefore, from Eq. (24) MVF of the corrected number of faults can be written as follows:

$$m_{\mathrm{c}}(t) = B(t)m_{\mathrm{d}}(t)$$

Notes 1 The failure intensity function $\lambda(t)$ can be obtained by differentiating w.r.t. t, i.e.,

$$\lambda(t) = \frac{\mathrm{d}m(t)}{\mathrm{d}t} \tag{31}$$

Notes 2 The conditional reliability of the proposed SRGM can be obtained using the following equation:

$$R(x|t) = \mathrm{e}^{-[m(t+x)-m(t)]} \tag{32}$$

Theorem 1 *Let* $m'_{\mathrm{d}}(t) = b(t)(a(t) - \phi(t)m_{\mathrm{d}}(t))$ *and* $a'(t) = \beta(t)m'_{\mathrm{d}}(t) + m_{\mathrm{d}}(t)\beta'(t)$ *with* $a(0) = a$ *and* $m_{\mathrm{d}}(0) = 0$ *on* $[0, \infty)$*, then MVF* $m_{\mathrm{d}}(t)$ *will be*

$$m_{\mathrm{d}}(t) = \left(\int_0^t b(x)a(x)e^{B(x)}dx\right)e^{-B(t)} \tag{33}$$

where $B(t) = -\int_0^t b(t)\phi(t)dt$.

Theorem 2 *Let* $m'_{\mathrm{d}}(t) = b(t)(a(t) - \phi(t)m_{\mathrm{d}}(t))$ *and* $a'(t) = \beta(t)m'_{\mathrm{d}}(t) + m_{\mathrm{d}}(t)\beta'(t)$ *with* $a(0) = a$ *and* $m_{\mathrm{d}}(0) = 0$ *on* $[0, \infty)$*, then fault content function* $a(t)$ *will be*

$$a(t) = a + \left(\int_0^t \beta(x)b(x)a(x)e^{B(y)}dx\right)e^{-B(t)} \tag{34}$$

where $B(t) = -\int_0^t b(t)\phi(t)dt$.

3.3.3 Proposed SRGM with a Change Point

Due to change in different factors such as testing strategy, testing environment, testing effort and defect density, change is possible at certain point in different parameters of SRGMs. In the proposed SRGM, fault introduction and detection rate are two crucial parameters and it can be changed whenever a change point occurs. Therefore, it is very important to consider the influence of change point on these parameters. Fault introduction and detection rate can be redefined with change point as follows:

$$\beta(t) = \begin{cases} \beta_1 t(1-\alpha_1 t), & 0 \le t \le \tau \\ \beta_2 t(1-\alpha_2 t), & t > \tau \end{cases} \tag{35}$$

where β_1, β_2, α_1 and α_2 are constants and τ is the change point and

$$b(t) = \begin{cases} b_1 t^{k_1}, & 0 \le t \le \tau \\ b_2 t^{k_2}, & t > \tau \end{cases} \tag{36}$$

where b_1, b_2, k_1 and k_2 are constants, τ is the change point.

Using Eqs. (35) and (36), MVF of detected number of faults can be obtained for interval $(0 \le t \le \tau)$ and $(t > \tau)$, by solving Eqs. (28) and (29) simultaneously with initial conditions $m_\mathrm{d}(0) = 0$ and $a(0) = a$ as follows:

$$\begin{aligned} m_\mathrm{d}(t) = ab_1 \Bigg[& \frac{t^{k_1+1}}{k_1+1} - b_1\beta_1 \left\{ \frac{t^{(2k_1+3)}}{(2k_1+3)(k_1+2)} \right. \\ & \left. - \alpha_1 \frac{t^{(2k_1+4)}}{(k_1+3)(2k_1+4)} \right\} + b_1 N m\eta \\ & \left\{ \frac{t^{(m+2k_1+1)}}{(m+k_1)(m+2k_1+1)} \right. \\ & \left. -\eta \frac{t^{(2m+2k_1+1)}}{(2m+k_1)(2m+2k_1+1)} \right\} \Bigg] \\ & \times \exp\left[b_1\beta_1 \left\{ \frac{t^{k_1+2}}{(k_1+2)} - \alpha_1 \frac{t^{k_1+3}}{(k_1+3)} \right\} \right. \\ & \left. - b_1 N m\eta \left\{ \frac{t^{m+k_1}}{(m+k_1)} - \eta \frac{t^{2m+k_1}}{(2m+k_1)} \right\} \right] \quad \text{for} \quad 0 \le t \le \tau \end{aligned} \tag{37}$$

and

$$\begin{aligned} m_\mathrm{d}(t) = a \Bigg[\Bigg\{ & 1 + b_1 N mc \left\{ \frac{\tau^{m+k_1}}{(m+k_1)} - c \frac{\tau^{2m+k_1}}{(2m+k_1)} \right\} \\ & - b_1\beta_1 \left\{ \frac{\tau^{k_1+2}}{(k_1+2)} - \alpha_1 \frac{\tau^{k_1+3}}{(k_1+3)} \right\} \\ & + b_2\beta_2 \left\{ \frac{\tau^{k_2+2}}{(k_2+2)} - \alpha_2 \frac{\tau^{k_2+3}}{(k_2+3)} \right\} \\ & - b_2 N mc \left\{ \frac{\tau^{m+k_2}}{(m+k_2)} - c \frac{\tau^{2m+k_2}}{(2m+k_2)} \right\} \Bigg\} \\ & \frac{b_2 \left\{ t^{k_2+1} - \tau^{k_2+1} \right\}}{(k_2+1)} + {b_2}^2 N mc \\ & \left\{ \frac{t^{m+2k_2+1} - \tau^{m+2k_2+1}}{(m+k_2)(m+2k_2+1)} - c \frac{t^{2m+2k_2+1} - \tau^{2m+2k_2+1}}{(2m+k_2)(2m+2k_2+1)} \right\} \\ & -{b_2}^2 \beta_2 \left\{ \frac{t^{2k_2+3} - \tau^{2k_2+3}}{(k_2+2)(2k_2+3)} \right. \\ & \left. - \alpha_2 \frac{t^{2k_2+4} - \tau^{2k_2+4}}{(k_2+3)(2k_2+4)} \right\} + m(\tau) \Bigg] \times \exp\{-b_1 N m \\ & \left\{ \frac{\tau^{m+k_1}}{(m+k_1)} - c \frac{\tau^{2m+k_1}}{(2m+k_1)} \right\} + b_1\beta_1 \left\{ \frac{\tau^{k_1+2}}{(k_1+2)} - \alpha_1 \frac{\tau^{k_1+3}}{(k_1+3)} \right\} \\ & + b_2\beta \left\{ \frac{t^{k_2+2} - \tau^{k_2+2}}{(k_2+2)} - \alpha_2 \frac{t^{k_2+3} - \tau^{k_2+3}}{(k_2+3)} \right\} \\ & - b_2 N mc \left\{ \frac{t^{m+k_2} - \tau^{m+k_2}}{(m+k_2)} - c \frac{t^{2m+k_2} - \tau^{2m+k_2}}{(2m+k_2)} \right\} \Bigg\} \\ & for\ t > \tau \end{aligned} \tag{38}$$

Similarly, MVF of corrected number of faults can be obtained using Eq. (24). The failure intensity and conditional reliability of the proposed SRGM can be obtained using Eqs. (31) and (32).

4 Numerical Example

4.1 Software Failure Data and Comparison Criteria

To validate the proposed SRGM, two data sets have been used here. First data set (Data Set I) has been published by Wu et al. [23]. It represents the failure pattern of detected and corrected faults of a middle-sized software project. The software was tested for 17 weeks. Totally, 144 faults were detected and 143 faults were corrected. Second data set (Data Set II) has been published by Musa et al. [4] for System 2. This data set represents the failure pattern of detected and corrected faults. It was tested for 17 weeks. Totally, 54 faults were detected and 54 faults were corrected. The following comparison criteria have been used to compare the performance of proposed SRGM with some existing SRGMs proposed in [17,18,41,44,45]:

4.1.1 Mean Square Error

It is defined as [13,14]:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{39}$$

where y_i and $\hat{y}_i$ are the observed and predicted faults, respectively, and n is the total number of observations. MSE for models based on the concept of fault detection and correction process has been calculated as the average of MSE of detected faults, i.e., MSE_d and MSE of corrected faults MSE_c. Hence,

$$\text{MSE} = \frac{\text{MSE}_d + \text{MSE}_c}{2} \tag{40}$$

4.1.2 Bias

It is defined as the sum of the deviation of the estimated curve from the actual data, as given [48,49]:

$$\text{Bias} = \frac{1}{n}\sum_{k=1}^{n}(m(t_k) - m_k) \tag{41}$$

lower value of bias is better goodness of fit.

4.1.3 Variance

It is defined as follows [48,49]:

$$\text{Variance} = \sqrt{\frac{1}{n-1}\sum\nolimits_{i=1}^{n}(m_k - m(t_k) - \text{Bias})^2} \tag{42}$$

lower value of variance is better goodness of fit.

4.1.4 The Root Mean Square Prediction Error (RMSPE)

It measure the closeness with which a model predicts the observation, as shown [48,49]:

$$\text{RMSPE} = \sqrt{\text{Variance}^2 + \text{Bias}^2} \tag{43}$$

lower value of RMSPE is better goodness of fit.

4.1.5 Confidence Interval for m(t)

It is defined as follows [50]:

$$\hat{m}(t) + \eta_p\sqrt{\hat{m}(t)} \quad \text{and} \quad \hat{m}(t) - \eta_p\sqrt{\hat{m}(t)}$$

The bounds of $m(t)$ approximately as follows:

$$\hat{m}(t) + \eta_p\sqrt{\hat{m}(t)} \geqslant m(t) \geqslant \hat{m}(t) - \eta_p\sqrt{\hat{m}(t)}$$

where $\hat{m}(t)$ is the estimate of and η_p is the $\frac{(1+p)}{2} \times 100$ percentile of the standard normal distribution.

4.1.6 Chi-Squared (χ^2) Goodness of Fit

To validate the proposed SRGM, Chi-squared (χ^2) goodness-of- fit test has been used. It is a very powerful test for testing the significance of the discrepancy between experiment and theory. It is defined as:

$$\chi^2 = \sum_{i=1}^{n}\frac{(O_i - E_i)^2}{E_i} \tag{44}$$

where O_i, $(i = 1, 2, 3, \ldots, n)$ is the set of observed frequency and $E_i (i = 1, 2, 3, \ldots, n)$ is the corresponding set of expected frequency.

4.2 Performance Analysis

In this section, performance of the proposed SRGM has been analyzed in comparison with the some well-known and widely accepted SRGMs, which are mainly based on FRF, and fault detection and correction process. First, model parameters have been estimated for fault detection process and compared with some well-known SRGMs, which are based on fault detection process and FRF [17,18]. Next, MVF of the proposed SRGM have been estimated for fault detection and correction process and compared with some SRGMs, which are developed considering fault detection and correction process [22,42,44,45]. The unknown parameters of the proposed SRGM have been estimated using least square method. The computation of the parameters of the proposed SRGM has been carried out using Statistical Package for Social Sciences (SPSS) software. Change point has been estimated using 'changepoint' package in 'R' software [51].

4.2.1 Data Set I

(a) Fault Detection Process

For the fault detection process, estimated parameters of the proposed SRGM and other SRGMs are presented in Table 3. The change point has been detected at position $\tau = 7$ weeks for this data set. The actual values of FRF for each time point of first data set are shown in Table 1. Using these values, the pattern of the FRF curve is estimated with Weibull curve, which is shown in Fig. 1. As shown in this figure, estimated FRF curve follows the similar pattern of actual FRF curve. This establishes the (vi)th assumption of the proposed SRGM, i.e., FRF can be represented by Weibull curve. The estimated parameters of the FRF are tabulated in Table 2.

As shown in Table 3, the estimated number of initial faults using the proposed SRGM is 144.02, which is very close to the actual number of faults detected in the software at the end of testing, i.e., 144, while the estimated number of faults at the end of testing is 130.6114. It means 10 faults still present in the software which are unable to detect. This establishes

Table 1 Data Set I

Weeks	Detected faults	Corrected faults	Fault reduction factors
1	12	3	0.25
2	23	3	0.130435
3	43	12	0.27907
4	64	32	0.5
5	84	53	0.630952
6	97	78	0.804124
7	109	89	0.816514
8	111	98	0.882883
9	112	107	0.955357
10	114	109	0.95614
11	116	113	0.974138
12	123	120	0.97561
13	126	125	0.992063
14	128	127	0.992188
15	132	127	0.962121
16	141	135	0.957447
17	144	143	0.993056

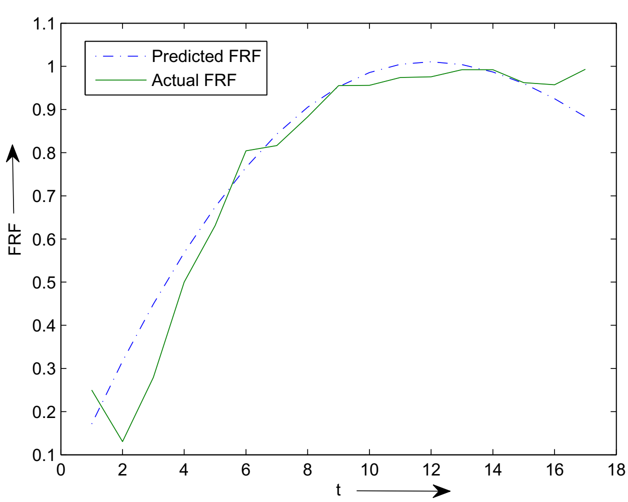

Fig. 1 Prediction of fault reduction factor for Data Set I with using Weibull curve

Table 2 Estimated parameters of FRF for Data Set I

Fault reduction factor	Estimated parameter		
	N	m	η
	21.486	1.905	0.0042

the fact that these faults will be detected in the operational phase. On the other hand, the values of b_2 are greater than the b_1, and also the value of k_2 is greater than the k_1. This implies that the fault detection rate increases after the change point since the learning process and maturity of the testing team increases as the testing time proceeds. Therefore, it is reasonable that the fault detection rate increases. Similarly, the value of β_1 is greater than the β_2, while α_1 and α_2 have the similar value. It means fault introduction rate decreases after the change point. In general, at the beginning of the testing process more faults introduced, and as the learning and maturity of the testing team increase, it decreases later. This established the (iv)th assumption of the proposed SRGM, i.e., fault introduction rate decreases as testing time increases. Moreover, estimated faults using the proposed SRGM and the actual faults present in the software with 95 % confidence bound are shown in Fig. 2. This figure illustrates that the estimated number of faults is very close to the actual faults present in the software as well as it lies between the confidence limits.

As shown in Table 3, the proposed SRGM produces the lower value of MSE along with bias, variance, etc., compared with the other SRGMs [17,18]. Moreover, the estimated result is acceptable at 1 % level of significance as the computed value of Chi-square for proposed SRGM is less than the tabulated value at 1 % level of significance, i.e., $\chi^2_{\text{computed}} = 5.2037 < \chi^2_{\text{tabulated}} = 11.341$, for 3 degrees of freedom. This validates the proposed SRGM.

(b) Fault Detection and Correction Process

From the above result, it is clear that the proposed SRGM performs better for fault detection process. The fault correction behavior of the proposed SRGM has been computed using the results tabulated in Table 4. The results obtained using the proposed SRGM has been compared with some existing SRGMs [22,41,44,45], which are mainly based on the concept of fault detection and correction process.

The estimated parameters of SRGMs along with different comparison criteria are tabulated in Table 4. As shown in this table, the proposed SRGM produces the lower value of MSE for detected fault, corrected fault and the average of both detected and corrected faults than the other SRGMs [22,42,44,45]. Graphical representation of estimated corrected number of faults by proposed SRGM with 95 % confidence bound is shown in Fig. 3, which represents the graphical comparison of the estimated corrected faults and actual corrected faults. From this figure, it can be seen that the estimated faults by the proposed SRGM are closer to the actual faults present in the software and also lie within the confidence limits. This establishes the fact that the proposed SRGM has better goodness of fit for both the fault detection and correction process.

Overall, it can be concluded that the performance of the proposed SRGM is better than the other SRGMs.

Table 3 Estimated parameters of the proposed SRGM, other SRGMs and their comparison for Data Set I

S. no.	SRGMs	Estimated parameters		MSE	Bias	Variance	RMSPE
		a	144.02				
		b_1	0.002				
		b_2	0.302				
		β_1	8.500				
1	Proposed SRGM	β_2	0.042	11.7485	-0.1343	3.5304	3.5329
		k_1	0.080				
		k_2	0.110				
		α_1	0.020				
		α_2	0.020				
		a	154.21				
2	Hsu SRGM Case 1 [17]	B_0	0.386	48.8116	0.9526	7.1342	7.1975
		k_1	0.365				
		a	144.309				
		r	0.189				
3	Hsu SRGM Case 2 [17]	B_0	0.500	33.6805	0.5097	5.9589	5.9807
		k	0.543				
		a	154.390				
		r	0.154				
4	Hsu SRGM Case 3 [17]	B_0	0.914	48.8282	1.0318	7.1238	7.1981
		k	0.000029				
		a	138.767				
		r	0.150				
5	Pachauri SRGM 1 [18]	α	1.536	24.1731	0.0056	5.0679	5.0679
		β	5.966				
		a	143.847				
		r	0.248				
6	Pachauri SRGM 2 [18]	γ	0.00001	24.0789	0.1618	5.0552	5.0578
		α	0.772				
		β	0.998				

4.2.2 Data Set II

(a) Fault Detection Process

For this data set, the change point has been detected at position $\tau = 9$ weeks.

The actual values of FRF for each time point of Data Set II are shown in Table 5. The pattern of the FRF curve is estimated with Weibull curve, which is shown in Fig. 4. As shown in this figure, estimated FRF curve follows the pattern of actual FRF curve. This establishes the (vi)th assumption of the proposed SRGM, i.e., FRF can be represented by Weibull curve. The estimated parameters of the FRF are tabulated in Table 6. For the fault detection process, estimated parameters of the proposed SRGM and other SRGMs are presented in Table 7.

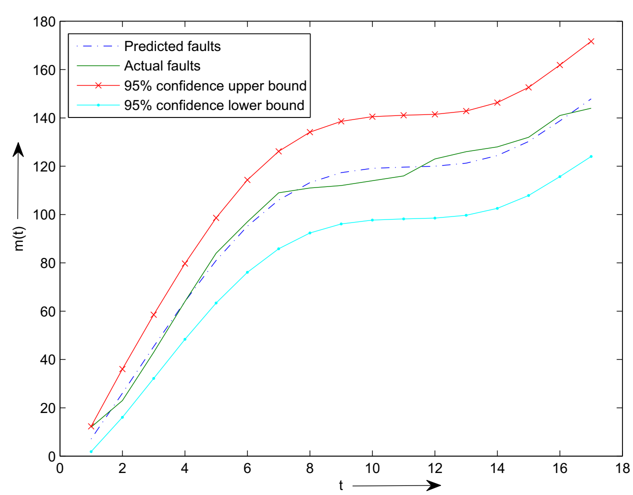

Fig. 2 Prediction of cumulative detected faults for Data Set I along with 95 % confidence bound using proposed SRGM

Table 4 Estimated parameters of the proposed SRGM and other SRGMs for fault correction process for Data Set I

S. no.	SRGMs	Estimated parameters		MSE
1	Proposed SRGM	As mentioned in Table 3		$MSE_d = 13.767$ $MSE_c = 43.912$ $MSE = 28.839$
2	Lo SRGM and Xie SRGM [22,45]	a b μ	156.34 0.1404 0.5810	$MSE_d = 50.635$ $MSE_c = 59.748$ $MSE = 55.192$
3	Xie and Zhao SRGM [44]	a b c	168.36 0.1193 0.0277	$MSE_d = 58.0822$ $MSE_c = 151.7077$ $MSE = 104.8949$
4	Schneidewind SRGM [41]	a b $\triangle$	153.01 0.1487 1.9390	$MSE_d = 41.00$ $MSE_c = 52.01$ $MSE = 30.00$

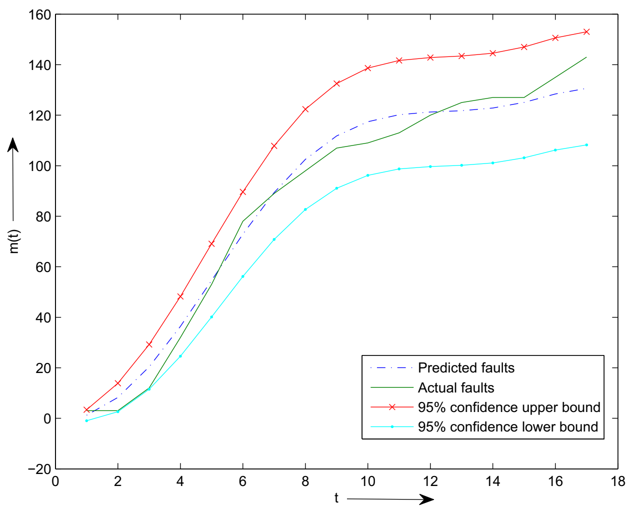

Fig. 3 Prediction of cumulative number of corrected faults and its 95 % confidence interval for Data Set I

Table 5 Data Set II

Weeks	Detected faults	Corrected faults	Fault reduction factors
1	1	0	0
2	2	2	1.000
3	4	3	0.75
4	5	5	1.000
5	13	12	0.92307
6	22	18	0.81818
7	28	25	0.892857
8	35	33	0.94285
9	39	36	0.923076
10	42	36	0.857142
11	42	39	0.928571
12	46	42	0.913043
13	47	46	0.978123
14	47	47	1.000
15	49	48	0.97959
16	51	50	0.980392
17	54	54	1.000

As shown in Table 7, the estimated number of initial faults using the proposed SRGM is 58.900, which is very close to the actual number of faults detected in the software at the end of testing, i.e., 54, while the estimated number of faults at the end of testing is 51.5487. It means 7 faults still present in the software which are unable to detect. This establishes the fact that these faults will be detected in the operational phase. On the other hand, the values of b_2 is greater than the b_1, also the value of k_2 is grater than the k_1. This implies that the fault detection rate increases after the change point. Since, the learning process and maturity of the testing team increases as the testing time proceeds. Hence, it is reasonable to assume that the fault detection rate increases. Similarly, the value of β_1 is greater than the β_2, and the value of α_1 is grater than α_2. It means fault introduction rate decreases after the change point. In general, in the beginning of the testing process more faults introduced, and as the learning and maturity of the testing team increase, it decreases later. This established the (iv)th assumption of the proposed SRGM, i.e., fault introduction rate decreases as testing time increases. Moreover, estimated faults using the proposed SRGM and the actual faults present in the software with 95 % confidence bound are shown in Fig. 5. This figure illustrates that the estimated number of faults is very close to the actual faults present in the software as well as it lies between the confidence limits.

As shown in Table 7, the proposed SRGM produces the lower value of MSE along with bias, variance, etc.,

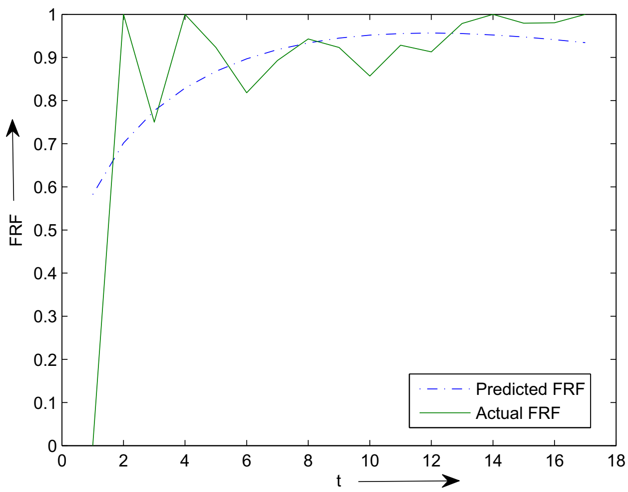

Fig. 4 Prediction of fault reduction factor for Data Set II with using Weibull curve

Table 6 Estimated parameters of FRF for Data Set II

Fault reduction factor	Estimated parameter		
	N	m	η
	48.004	1.289	0.010

compared with the other SRGMs [17,18]. Moreover, the estimated result is acceptable at 1 % level of significance as the computed value of Chi-square for proposed SRGM is less than the tabulated value at 1 % level of significance, i.e., $\chi^2_{\text{computed}} = 4.9043 < \chi^2_{\text{tabulated}} = 11.341$, for 3 degrees of freedom. This validates the proposed SRGM.

(b) Fault Detection and Correction Process

Estimated parameters of the different SRGMs [22,42,44,45], based on the concept of fault detection and correction process, and their comparison with the proposed SRGM are tabulated in Table 8. From this table, it is clear that MSE produced by proposed SRGM for detected fault, corrected fault and the average of the both detected and corrected faults is also lower than the other SRGMs [22,42,44,45]. Graphical representation of estimated corrected number of faults by proposed SRGM with 95 % confidence bound is shown in Fig. 6, which represents the graphical comparison of the estimated corrected faults and actual corrected faults. From this figure, it can be seen that the estimated faults by the proposed SRGM are closer to the actual faults present in the software and also lies within the confidence limits. This establishes the fact that the proposed SRGM has better goodness of fit for both the fault detection and correction process.

Finally, it can be concluded that the performance of the proposed SRGM is better than the other SRGMs as well as the realistic for Data Set II also.

5 Sensitivity Analysis

Parameters play very important role to make a model accurate, realistic and flexible. A model mainly contains two types of parameters, i.e., more sensitive parameters and less sensitive parameters. More sensitive parameters have greater impact on model than less sensitive parameters. MVF of the SRGM is highly dependent on its parameters such as fault detection rate and fault introduction rate. Hence, in this section sensitivity analysis [29,52–54] has been carried to study the influence of the change in the parameters of the proposed SRGM. The sensitivity analysis for parameters has been performed on total number of faults.

5.1 Effect of Variation in Fault Detection Rate

Fault detection rate is one of the important parameters of the proposed SRGM. Hence, sensitivity analysis of the fault detection rate has been carried out to study the effect of change in fault detection rate on total number of faults estimated, by varying the parameters of the fault detection rate. The value of the fault detection rate has been increased 30 %. For the proposed SRGM, fault detection rate function has been defined with four constant parameters, i.e., b_1, b_2, k_1 and k_2. If constant parameter of fault detection rate function before change point, i.e., b_1, and after change point, i.e., b_2, is increased by 30 % one at a time, then the estimated value of total number of faults decreases from 147.8693 to −26.4897 and from 147.8693 to 118.4493, respectively, for Data Set I. It decreases from 51.5487 to 45.7716 and from 51.5487 to 50.3339, respectively, for Data Set II. If the values of both the parameters b_1 and b_2 are increased together by 30 %, then the estimated value of total number of faults decreases from 147.8693 to −4.1662 and from 51.5487 to 43.6823 for Data Set I and Data Set II, respectively.

Next, if constant parameters of fault detection rate function before change point, i.e., k_1, and after change point, i.e., k_2, are increased by 30 % one at a time, then the estimated value of total number of faults decreases from 147.8693 to 126.4281 and from 147.8693 to 139.9109, respectively, for Data Set I. It decreases from 51.5487 to 51.5393 and from 51.5487 to 51.4673, respectively, for Data Set II. If both the values of k_1 and k_2 are increased by 30 % together, then the estimated value of total number

Table 7 Estimated parameters of the proposed SRGM, other SRGMs and their comparison for Data Set II

S. no.	SRGMs	Estimated parameters		MSE	Bias	Variance	RMSPE
		a	58.900				
		b_1	0.00176				
		b_2	0.20				
		β_1	9.528				
1	Proposed SRGM	β_2	0.0355	3.2387	0.0460	1.7990	1.7996
		k_1	0.001				
		k_2	0.044				
		α_1	0.080				
		α_2	0.0001				
		a	167.440				
2	Hsu SRGM Case 1 [17]	B_0	0.174	23.7416	1.0045	4.9146	5.0162
		k_1	0.140				
		a	90.437				
		r	0.062				
3	Hsu SRGM Case 2 [17]	B_0	0.500	18.1203	0.9036	4.2878	4.3819
		k	0.317				
		a	167.440				
		r	0.085				
4	Hsu SRGM Case 3 [17]	B_0	0.0001	23.9128	0.6018	5.00225	5.0383
		k	0.024				
		a	59.842				
		r	0.233				
5	Pachauri SRGM 1 [18]	α	0.697	5.6539	0.3360	2.4263	2.4495
		β	10.00				
		a	56.293				
		r	0.363				
6	Pachauri SRGM 2 [18]	γ	0.0001	5.3207	0.4912	2.3066	2.3583
		α	0.577				
		β	10.000				

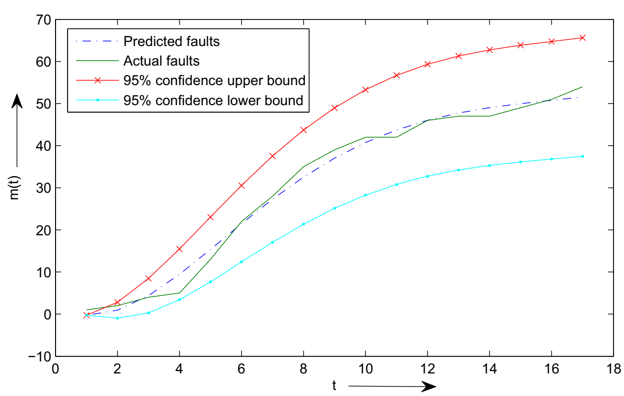

Fig. 5 Prediction of cumulative detected faults for Data Set II along with 95 % confidence bound using proposed SRGM

of faults decreases from 147.8693 to 120.3092 and from 51.5487 to 51.4576 for Data Set I and Data Set II, respectively.

5.2 Effect of Variation in Fault Introduction Rate

In this subsection, sensitivity analysis has been carried out on fault introduction rate. Fault introduction rate has been increased by 30 %. For the proposed SRGM fault introduction rate function has been defined with four important constant parameters, i.e., β_1, β_2, α_1 and α_2. If constant parameter of fault introduction rate function before change point, i.e., β_1, and after change point, i.e., β_2, is increased by 30 % one at a time, then the estimated value of total number of faults decreases from 147.8693 to −25.6575 and increases from 147.8693 to 219.7630, respectively, for Data Set I. It increases from 51.5487 to 59.6831 and from 51.5487 to 62.4612, respectively, for Data Set II. If both the values of β_1 and β_2 are increased by 30 % together, then the estimated value of total number of faults decreases from 147.8693 to −53.6695 and increases from 51.5487 to 72.0073 for Data Set I and Data Set II, respectively.

Table 8 Estimated parameters of the proposed SRGM and other SRGMs for fault correction process for Data Set II

S. no.	SRGMs	Estimated parameters		MSE
				$MSE_d = 3.2387$
1	Proposed SRGM	As mentioned in Table 7		$MSE_c = 6.1209$
				$MSE = 4.6798$
		a	76.400	$MSE_d = 37.7854$
2	Lo SRGM and Xie SRGM [22,45]	b	0.0780	$MSE_c = 10.8105$
		μ	0.5700	$MSE = 24.2979$
		a	71.000	$MSE_d = 32.4071$
3	Xie and Zhao SRGM [44]	b	0.0740	$MSE_c = 30.9609$
		c	0.0060	$MSE = 31.684$
		a	69.000	$MSE_d = 33.4663$
4	Schneidewind SRGM [41]	b	0.0780	$MSE_c = 32.0467$
		$\triangle$	0.0070	$MSE = 32.7565$

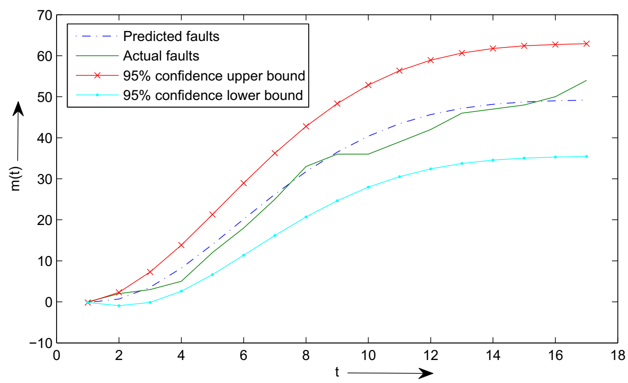

Fig. 6 Prediction of cumulative number of corrected faults and its 95 % confidence interval for Data Set II

Next, if constant parameter of fault introduction rate function before change point, i.e., α_1, and after change point, i.e., α_2, is increased by 30 % one at a time, then the estimated value of total number of faults increases from 147.8693 to 151.8066 and decreases from 147.8693 to 128.6038, respectively, for Data Set I. It decreases from 51.5487 to 41.3063 and from 51.5487 to 51.5349, respectively, for Data Set II. If both the values of α_1 and α_2 are increased by 30 % together, then the estimated value of total number of faults decreases from 147.8693 to 131.9560, from 51.5487 to 41.2956 for Data Set I and Data Set II, respectively.

5.3 Effect of Variation in Fault Reduction Factor

In this subsection, sensitivity analysis has been carried out on fault reduction factor. Fault reduction factor has been increased by 30 %. For the proposed SRGM fault reduction function has been defined as Weibull curve with parameters N, m, η. If constant parameter of fault reduction factor, 'N,' is increased by 30 %, then the estimated value of total number of faults decreases from 147.8693 to 41.2551 and from 51.5487 to 23.6748 for Data Set I and Data Set II, respectively. If the value of m is increased by 30 %, then the estimated value of total number of faults decreases from 147.8693 to −7.6598 and from 51.5487 to 11.0494 for Data Set I and Data Set II, respectively. If the value of η is increased by 30 %, then the estimated value of total number of faults increases from 147.8693 to 270.4204 and decreases from 51.5487 to 44.7188 for Data Set I and Data Set II, respectively.

5.4 Summary of The Result of Sensitivity Analysis for Initial Faults

From the above results, it is clear that parameters of the fault detection rate function impose similar effect on total number of faults as it is decreasing, while parameters of the fault introduction rate and fault reduction factor impose different effect on total number of fault as it is either increasing or decreasing. In fault detection rate function, the impact of the parameters b_1 and b_2 is greater than k_1 and k_2 because these parameters cause greater change in total number of faults. Similarly, for fault introduction rate function the impact of the parameters β_1 and β_2 is greater than α_1 and α_2. In fault reduction factor, function parameters N and m impose similar effect on total number of faults as it is decreasing, while η imposes different effects on total number of faults as it is either increasing or decreasing. Hence, it can be interpreted that the parameters of fault introduction rate function and fault reduction factor are more sensitive and it should be estimated very carefully.

6 Conclusion

A SRGM will be more realistic and accurate in prediction if each estimated parameter of the SRGM follows their real nature. In this article, it has been tried to predict the exact nature of FRF using Weibull curve first, and then it has been incorporated in the proposed SRGM. Concept of imperfect debugging and change point has been also incorporated in the proposed SRGM to make it more flexible and realistic. The number of faults estimated for two different data sets is very close to the number of actual faults presents in the two software. Experimental results prove that the predicted values of the detected and corrected faults by the proposed SRGM are better fit to the actual data set. Based on the above discussion, it can be concluded that the proposed SRGM is better and it will be useful for researchers and software engineers.

Acknowledgments The authors are thankful to referees and editor for their valuable suggestions and comments. The authors are also thankful to Indian School of Mines, Dhanbad, India, for providing necessary facilities to complete the proposed work.

References

1. Kapur, P.K.; Pham, H.; Gupta, A.; Jha, P.C.: Software Reliability Assessment with OR application. Springer, New York (2011)
2. Pham, H.: System Software Reliability. Springer, New York (2006)
3. Wohlin, C.; Runeson, P.; Hst, M.; Ohlsson, M.C.; Regnell, B.; Wessln, A.: Experimentation in Software Engineering. Springer, New York (2012)
4. Musa, J.D.; Iannino, A.; Okumoto, K.: Software Reliability Measurement, Prediction, Application. McGraw-Hill, New York (1987)
5. Kitchenham, B.A.: Software quality assurance. Microprocess. Microsyst. **13**(6), 373–381 (1989)
6. Kitchenham, B.; Pfleeger, S.L.: Software quality: the elusive target. IEEE Softw. **1**, 12–21 (1996)
7. Xie, M.; Hong, G.Y.; Wohlin, C.: Software reliability prediction incorporating information from a similar project. J. Syst. Softw. **49**(1), 43–48 (1999)
8. Wohlin, C.: Engineering and Managing Software Requirements. Springer, New York (2005)
9. Chatterjee, S.; Nigam, S.; Singh, J.B.; Upadhyaya, L.N.: Application of fuzzy time series in prediction of time between failures & faults in software reliability assessment. Fuzzy Inf. Eng. **3**(3), 293–309 (2011)
10. Chatterjee, S.; Nigam, S.; Singh, J.B.; Upadhyaya, L.N.: Software faults predication using NARX network. Appl. Intell. **37**(1), 121–129 (2012)
11. Chatterjee, S.; Nigam, S.; Singh, J.B.; Upadhyaya, L.N.: Transfer function modeling in software reliability. Computing **92**(1), 33–48 (2011)
12. Chatterjee, S.; Singh, J.B.; Nigam, S.; Upadhyaya, L.N.: A Study of a single multiplicative neuron (SMN) model for software reliability prediction. Innovations in Intelligent Machines-3 Contemporary Achievements in Intelligent Systems. Book Edited by Springer-Verlag, pp. 89–103 (2012)
13. Lyu, M.R.: Handbook of Software Reliability Engineering. IEEE Computer Society Press/McGraw-Hill, New York (1996)
14. Pham, H.: Software Reliability. Springer, New York (2000)
15. Xie, M.: Software Reliability Modeling. World Scientific Press, Singapore (1991)
16. Musa, J.D.: A theory of software reliability and its application. IEEE Trans. Soft. Eng. SE-1 **3**, 312–327 (1975)
17. Hsu, C.J.; Huang, C.Y.; Chang, J.R.: Enhancing software reliability modeling and prediction through the introduction of time variable fault reduction factor. Appl. Math. Model. **35**(1), 506–521 (2011)
18. Pachauri, B.; Dhar, J.; Kumar, A.: Incorporating inflection S-shaped fault reduction factor to enhance software reliability growth. Appl. Math. Model **39**(5), 1463–1469 (2014)
19. Hu, Q.P.; Xie, M.; Ng, S.H.; Levitin, G.: Robust recurrent neural network modeling for software fault detection and correction prediction. Reliab. Eng. Syst. Saf. **92**(3), 332–340 (2007)
20. Huang, C.Y.; Huang, W.: Software reliability analysis and measurement using finite and infinite server queueing models. IEEE Trans. Reliab. **57**(1), 192–203 (2008)
21. Huang, C.Y.; Lin, C.: Software reliability analysis by considering fault dependency and debugging time lag. IEEE Trans. Reliab. **55**(3), 436–450 (2006)
22. Lo, J.; Huang, C.: An integration of fault detection and correction processes in software reliability analysis. J. Syst. Soft. **79**(9), 1312–1323 (2006)
23. Wu, Y.P.; Hu, Q.P.; Xie, M.; Ng, S.H.: Modeling and analysis of software fault detection and correction process by considering time dependency. IEEE Trans. Reliab. **56**(4), 629–642 (2007)
24. Zhao, M.: Change-point problems in software and hardware reliability. Commun. Stat. Theory Methods **22**(3), 757–768 (1993)
25. Chatterjee, S.; Nigam, S.; Bahadur, J.; Upadhyaya, L.N.: Effect of change point and imperfect debugging in software reliability and its optimal release policy. Math. Comput. Model. Dyn. Syst. **18**(5), 1–13 (2012)
26. Huang, C.Y.; Lyu, M.R.: Estimation and analysis of some generalized multiple change-point software reliability models. IEEE Trans. Reliab. **60**(2), 498–514 (2011)
27. Inoue, S.; Yamada, S.: Optimal software release policy with change point. In: Proceedings of 2008 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 157–163 (2008)
28. Kapur, P.K.; Kumar, A.; Yadav, K.; Khatri, S.K.: Software reliability growth modelling for errors of different severity using change point. Int. J. Reliab. Qual. Sof. Eng. **14**(4), 311–326 (2007)
29. Li, X.; Xie, M.; Ng, S.H.: Sensitivity analysis of release time of software reliability models incorporating testing effort with multiple change-points. Appl. Math. Model. **34**(11), 3560–3570 (2010)
30. Shyur, H.J.: A stochastic software reliability model with imperfect debugging and change point. J. Syst. Soft. **66**(2), 135–141 (2003)
31. Chang, I.P.: An analysis of software reliability with change-point models. NSC 85-2121-M031-003, National Science Council, Taiwan, (1997)
32. Chang, Y.P.: Estimation of parameters for non-homogeneous Poisson process software reliability with change-point model. Commun. Stat. Simul. Comput. **30**(3), 623–635 (2001)
33. Kapur, P.K.; Singh, V.B.; Anand, S.; Yadavalli, V.S.S.: Software reliability growth model with change-point effort control using a power function of the testing time. Int. J. Prod. Res. **46**(3), 771–787 (2008)
34. Goel, A.L.: Software reliability models: assumptions, limitations and applicability. IEEE Trans. Soft. Eng. **SE-11**(12), 1411–1423 (1985)
35. Jelinski, Z.; Moranda, P. : Software reliability research. In: Freiberger, W (ed.) Statistical Computer Performance Evaluation, pp. 465–484. Academic Press, New York (1972)
36. Chatterjee, S.; Misra, R.B.; Alam, S.S.: A generalized shock model for software reliability. Comput. Electr. Eng. Int. J. **24**, 363–368 (1998)
37. Chatterjee, S.; Singh, J.B.: A NHPP based software reliability model and optimal release policy with Logistic-Exponential test coverage under imperfect debugging. Int. J. Syst. Assur. Eng. Manag. **5**(3), 399–406 (2014)

38. Pham, H.: Software reliability assessment: imperfect debugging and multiple failure types in software development. EG&G-RAMM-10737. Idaho National Engineering Laboratory, Idaho Falls, ID (1993)
39. Yamada, S.; Tokuno, K.; Osaki, S.: Imperfect debugging models with fault introduction rate for software reliability assessment. Int. J. Syst. Sci. **23**(12), 2241–2252 (1992)
40. Kapur, P.K.; Pham, H.; Anand, S.; Yadav, K.: 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 (2011)
41. Schneidewind, N.: Analysis of error processes in computer software. SigPlan Proc. Proc. Int. Conf. Reliab. Softw. **10**(6), 337–346 (1975)
42. Goel, A.L.; Okumoto, K.: A time-dependent error detection rate model for software reliability and other performance measure. IEEE Trans. Reliab. **R-28**, 206–211 (1979)
43. Yamada, S.; Ohba, M.; Osaki, S.: S-shaped reliability growth modeling for software error detection. IEEE Trans. Reliab. **32**(5), 475–484 (1983)
44. Xie, M.; Zhao, M.: The Schneidwind software reliability model revisited. In: Proceeding of the 3rd International Symposium on Software Reliability Engineering, pp. 184–192 (1992)
45. Xie, M.; Hu, Q.; Wu, Y.; Ng, S.: A study of the modeling and analysis of software fault-detection and fault-correction processes. J. Qual. Softw. Reliab. Eng. Int. **23**(4), 459–470 (2007)
46. Kitchenham, B.; Hughes, R.T.; Linkman, S.G.: Modeling software measurement data. IEEE Trans. Softw. Eng. **27**(9), 788–804 (2001)
47. Wohlin, C.; Hst, M.; Runeson, P.; Wessln, A.: Software Reliability. Encycl. Phys. Sci. Technol. **15**, 25–39 (2001)
48. Huang, C.Y.; Kuo, S.Y.: Analysis of incorporating logistic testing effort function into software reliability modeling. IEEE Trans. Reliab. **51**(3), 261–270 (2002)
49. Pillai, K.; Nair, V.S.S.: A model for software development effort and cost estimation. IEEE Trans. Softw. Eng. **23**(8), 485–497 (1997)
50. Yin, L.; Trivedi, K.S.: Confidence interval estimation of NHPP-based software reliability models In: Proceedings of the 10th International Symposium on Software Reliability Engineering, pp. 6–11 (1999)
51. Killick, R.; Eckley I., changepoint: An R package for change point analysis. J. Stat. Softw. **58**(3), 1–19 (2014)
52. Huang, C.Y.; Lyu, M.R.: Optimal testing resource allocation, and sensitivity analysis in software development. IEEE Trans. Reliab. **54**(4), 592–603 (2005)
53. Karnava, W.J.; Sanchez, P.J.; Terry Bahill, A.: Sensitivity analyses of continuous and discrete systems in the time and frequency domains. IEEE Trans. Syst. Man Cybern. **23**(2), 488–501 (1993)
54. Lo, J.H.; Huang, C.Y.; Chen, I.Y.; Kuo, S.Y.; Lyu, M.R.: Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure. J. Syst. Softw. **76**(1), 3–13 (2005)