cogent engineering

COMPUTER SCIENCE | RESEARCH ARTICLE

Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations

Javaid Iqbal[1]*

Received: 29 June 2016
Accepted: 20 January 2017
First Published: 27 January 2017

*Corresponding author: Javaid Iqbal, Department of Computer Sciences, University of Kashmir, Srinagar, 190006, India
E-mail: iamjavaid@gmail.com

Reviewing editor:
Duc Pham, University of Birmingham, UK

Additional information is available at the end of the article

Abstract: The software testing process basically aims at building confidence in the software for its use in real world applications. The reliability of a software system is always important to us. As we carry out the error detection and correction phenomenon on our software, the reliability of the software grows. With an aim to model this growth in the software reliability, many formulations in the form of Software Reliability Growth Models have been proposed. Many of these are based on Non-Homogeneous Poisson Process framework.In this paper, a parallel comparison of the performance of the proposed software reliability growth models is carried out, considering linear and exponential fault content functions for study of imperfect debugging situations. The performance of the proposed models has been compared with some famous existing software reliability models and the proposed models have been validated on some real-world datasets. Three goodness-of-fit criteria that include mean square error, predictive-ratio risk and predictive power are used to carry out the performance comparison of the models. Using these comparison criteria on six actual failure datasets, it is concluded that the proposed Model-2 which always outperforms Model-1, fits the actual failure data better and has better predictive power than other considered SRGMs for at least two data sets.

Subjects: Applied Mathematics; Software Engineering & Systems Development; Computer Science (General)

Keywords: software reliability; reliability growth; software reliability growth model (SRGM); non-homogeneous poisson process (NHPP); learning effect; fault detection rate (FDR); imperfect debugging

Javaid Iqbal

ABOUT THE AUTHOR

Javaid Iqbal received his BSc in Mathematics from the University of Kashmir, and Masters in Computer Applications from the same university in 2004. He completed his PhD from the University of Kashmir in 2014. His research interests are Software engineering, Software reliability engineering and reliability modeling, and software performance engineering. He is a member of ACM, CSI and IETE

PUBLIC INTEREST STATEMENT

Software reliability is one of the prime quality concerns. Reliable software is dependable. Software testing helps to improve the reliability of software. This paper explores the assumption of imperfect debugging under two possibilities i.e. linear and exponential fault content functions. This paper demonstrates using real world datasets that under given assumptions how the proposed model with exponential fault content function is better suited to use for the given datasets. The researchers in this field will benefit from the comparison of imperfect debugging situations carried out in this paper.

1. Introduction

The scope and importance of having highly reliable software solutions deployed in every walk of life is an accepted and acknowledged fact. Evaluation of a software before its deployment is a must. Reliability is an important quality characteristic of any software. Software Reliability is defined as the probability that software will provide failure-free operation in a fixed environment for a fixed interval of time (Musa, Iannino, & Okumoto, 1987). The reliability of software applications is measured by use of mathematical formulations called as a software reliability growth model (SRGM) which basically describes the error-detection and correction processes. The SRGMs find their applications in many practical areas which include safety-critical systems like Shuttle's on-board systems software (Schneidewind & Keller, 1992) and weapon systems (Carnes, 1997).

Over the past three decades many SRGMs have been formulated under Non-homogeneous poisson process (NHPP) modeling framework. Many researchers have used NHPP based SRGMs to capture the reliability growth of a software from the processes of testing and debugging. These models include Goel and Okumoto model (Goel & Okumoto, 1979), Yamada and Ohba model (Yamada, Ohba, & Osaki, 1983), Ohba model (Ohba, 1984) etc. All such models use some underlying assumptions. The general NHPP model is based on the following assumptions: (1) The testing/debugging process follows an NHPP and (2) a debugging procedure is invoked immediately when a software error is detected. Researchers perceive some realistic assumptions to make a model better fit the error-detection and correction process. For example, it is more realistic to consider imperfect debugging than perfect debugging. Goel and Okumoto (1979) proposed an exponential SRGM. The Goel and Okumoto model is one of the earliest NHPP models for software reliability and has been extensively used in literature. This has a concave-shaped mean value function and could not accommodate learning effects very well. As a result, many modifications and generalizations of G–O model have been proposed. Yamada and Ohba (1983) proposed delayed S-shaped SRGM while Ohba (1984) proposed inflection S-shaped SRGM to allow learning effects. These models Goel and Okumoto (1979), Yamada et al. (1983), Ohba (1984) etc. assume perfect debugging. Learning phenomenon has been considered in Yamada, Tokuno, and Osaki (1992), Pham and Zhang (2003) etc. under imperfect debugging assumption. Learning phenomenon and imperfect debugging are two important aspects of many NHPP models.

In this study, imperfect debugging is considered as an increasing function, with linear and exponential behaviors assumed separately for sake of comparisons. This function is called as total fault content (TFC) function. The motivation behind this paper is to study imperfect debugging by a parallel comparison of the performance of the proposed software reliability growth models that incorporate linear and exponential fault content functions for imperfect debugging and then compare the results with some famous existing models using some real datasets. Indeed, linear and exponential fault content functions have been extensively studied to model imperfect debugging situations. It may be noted that the difference between perfect and imperfect debugging is the nature of TFC function i.e. total number of faults. Perfect debugging has constant TFC while as imperfect debugging has a dynamic TFC which is a function of testing time.

The rest of the paper is organized as: In Section 2 the model development is discussed. The mean value functions of two models are considered, many famous existing models are tabulated, sources of six datasets used in this study are listed and estimated parameter values are tabulated for these six datasets. In Section 3 numerical example are given to illustrate the usefulness of these models and to carry out the performance comparison using three comparison criteria of the models. Finally, discussion and conclusion of the analysis made is presented in Sections 4 and 5 respectively.

2. Model development

The solution of the following differential equation representing relationship between failure intensity and the initial fault content is used to represent a general class of NHPP models (Pham, 2007; Pham, Nordmann, & Zuemei Zhang, 1999):

$$\frac{dm(t)}{dt} = r(t) \cdot [a(t) - m(t)] \tag{1}$$

where $a(t)$ is the time-dependent fault content function which represents total number of faults in the system including the initial and introduced faults due to imperfect debugging, $r(t)$ is the time-dependent fault detection rate function representing faults detected per unit time, $m(t)$ representing mean value function.

The general solution of differential Equation (1) is represented by the following equation:

$$m(t) = e^{-B(t)} \cdot \{m_0 + \int_{t0}^{t} a(\tau)r(\tau)e^{B(\tau)}d\tau\} \tag{2}$$

where $B(t) = \int_{t_0}^{t} r(\tau)d\tau$ with $m_0 = m(t_0)$ is the marginal condition of (2) and t_0 is the time at which debugging starts.

With this modeling methodology available in literature, our models are developed under common assumption of an increasing time-dependent fault detection rate function $r(t)$ that exhibits an S-shaped behavior to capture learning effects and is represented by the following equation, where α, represents autonomous errors and η represents the learning factor.

$$r(t) = (\alpha + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t} + \eta}\right) \tag{3}$$

However, as per the intent of this paper, we consider two cases of imperfect debugging. In case of model-1, a linear time-dependent fault content function $a(t)$ is assumed (Yamada et al., 1992) where a is the initial fault content and β is the constant fault introduction rate. As is evident, this imperfect debugging situation considers that the current number of faults $a(t)$ is proportional to the testing time.

$$a(t) = a(1 + \beta \cdot t) \tag{4}$$

and in case of model-2, an exponential the time-dependent fault content function is assumed (Yamada et al., 1992), where a is the initial fault content and faults can be introduced exponentially per detected fault.

$$a(t) = c + a(1 - e^{-\beta t}) \tag{5}$$

These two behaviors of a TFC function have been extensively used to represent imperfect debugging situations in literature available for NHPP models. With the assumed behavior of $r(t)$, and separately considering each behavior of $a(t)$, in the generalized mean value function $m(t)$ for NHPP models in Equation (2), we arrive at the following mean value functions for our models:

Model-1 (MVF):

$$m(t) = \frac{a}{1 + \frac{\eta}{\alpha e^{-(\alpha+\eta)t}}}\left(1 - e^{-(\alpha+\eta)t}\right)\left(1 - \frac{1}{\alpha + \eta}\right) + \beta \cdot t \tag{6}$$

Table 1. Listing of models, their mean value functions and fault detection rate functions

Model name	MVF and FDR functions
Goel–Okumoto (G–O) (Goel & Okumoto, 1979)	$m(t) = a(1 - e^{-bt})$
	$a(t) = a$
	$r(t) = b$
Yamada–Ohba delayed S-shaped (Yamada et al., 1983)	$m(t) = a(1 - (1 + bt)e^{-bt})$
	$a(t) = a$
	$r(t) = \frac{b^2}{(1+bt)}$
Ohba inflection S-shaped (Ohba, 1984)	$m(t) = \frac{a(1-e^{-bt})}{(1+ce^{-bt})}$
	$a(t) = a$
	$r(t) = \frac{b}{(1+ce^{-bt})}$
Yamada imperfect debugging (Yamada et al., 1992)	$m(t) = \frac{ab}{(\alpha+b)}(e^{\alpha t} - e^{-bt})$
	$a(t) = a \cdot e^{\alpha t}$
	$r(t) = b$
P–Z model (Pham & Zhang, 2003)	$m(t) = \frac{1}{1+\beta \cdot e^{-bt}}\left((a + c)(1 - e^{-bt}) - \frac{a \cdot b}{b-\alpha}(e^{-\alpha t} - e^{-bt})\right)$
	$a(t) = c + a(1 - e^{-\alpha t})$
	$r(t) = \frac{b}{(1+\beta e^{-bt})}$
Chiu–Huang–Lee learning (Chiu, Huang, & Lee, 2008)	$m(t) = a\left(1 - \frac{1+(\frac{\eta}{\alpha})}{(\frac{\eta}{\alpha})+e^{(\alpha+\eta)t}}\right)$
	$a(t) = a$
	$r(t) = (\alpha + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t}+\eta}\right)$
Model-1	$m(t) = \frac{a}{1+\frac{\eta}{\alpha}e^{-(\alpha+\eta)t}}(1 - e^{-(\alpha+\eta)t})\left(1 - \frac{1}{\alpha+\eta}\right) + \beta \cdot t$
	$a(t) = a(1 + \beta \cdot t)$
	$r(t) = (\alpha + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t}+\eta}\right)$
Model-2	$m(t) = \frac{\alpha}{\alpha+\eta \cdot e^{-(\alpha+\eta)t}}\left((a + c)(1 - e^{-(\alpha+\eta)t}) - \frac{a \cdot (\alpha+\eta)}{\alpha+\eta-\beta}(e^{-\beta t} - e^{-(\alpha+\eta)t})\right)$
	$a(t) = c + a(1 - \exp(-\beta \cdot t))$
	$r(t) = (\alpha + \eta)\left(1 - \frac{\eta}{\alpha e^{(\alpha+\eta)t}+\eta}\right)$

Model-2 (MVF):

$$m(t) = \frac{\alpha}{\alpha + \eta \cdot e^{-(\alpha+\eta)t}}\left((a + c)(1 - e^{-(\alpha+\eta)t}) - \frac{a \cdot (\alpha + \eta)}{\alpha + \eta - \beta}(e^{-\beta t} - e^{-(\alpha+\eta)t})\right) \tag{7}$$

Table 1 summarizes the proposed models and other famous existing NHPP models.

Table 2 presents the datasets by labels and the sources of datasets listed.

The parameters estimated for some of the model listed in Table 1 are presented in Table 3 for six datasets used.

Table 2. Sources of datasets and their description

Label/reference/data-set	Description
[1] Zhang and Pham (1998)/Failure data of misra system	Documented in Misra (1983) and also available in Zhang and Pham (1998), the data-set summarizes the number of failures per 1 h execution time interval and is recorded for 25 h with 136 cumulative failures
[2] Hossain and Dahiya (1993)/Failure data of NTDS System	Documented in Hossain and Dahiya (1993), core of NTDS is the development of software for the real –time, multi-computer system and consists of 38 different project schedules with each module supposed to having 3 phases of production, test and user. The failure data of this software reports time–between–failures in days. A total of 26 failures were found during the production phase and 5 during the test phase
[3] Pham and Zhang (2003)/Failure data of tandem software	Documented in Wood (1996) comprises four datasets obtained from four major releases (Release 1, 2, 3, 4) of software products at Tandem Computers company (Zhang, Teng, & Pham, 2003). Release-1 data-set comprises a total of 100 faults recorded in 20 weeks. The actual numbers of CPU hours used are also documented and a total of 10,000 CPU hours have been used. For sake of avoiding any confidentiality issues, the number of faults were normalized from 0 to 100, and the amount of testing effort consumption was proportionately translated into the range [0, 10,000] (Lin & Huang, 2008; Wood, 1996)
[4] Bai, Hu, Xie, and Ng (2005)/Failure data of space program	Documented in Bai et al. (2005) recorded from the testing of Space Program which consists of 9,564 lines of C code (6,218 are executable) and acts as an interpreter for an array definition language (ADL). Further details of this data-set are available in Rothermel, Untch, Chengyun Chu, and Harrold, (2001) and Bai et al. (2005). It records the 21 failures as per their order against their failure interval times
[5] Pham (2003)/Failure data of real time control system	Documented in Lyu (1996) recorded a total fault count of 136 against the time-between-failure (TBF) measured in seconds
[6] Jeske and Zhang (2005)/Failure data of wireless data service system	Documented in Jeske and Zhang (2005) where a description of its main functions as routing of voice channels and signaling messages to relevant radio resources and processing entities are also described. This data-set comprises three datasets (Release 1, 2, 3). R-1 had a life cycle of 13 months in the field with a cumulative exposure time of 58,633 system-days, recording a total of 33 failures, out of which 19 were unique. R-2 also had a life cycle of 13 months in the field with a cumulative exposure time of 167,900 system-days, recording a total of 115 failures, out of which 71 were unique. R-3 consists of failures recorded during feature testing and load testing. Exact specifications of this release can be found in Jeske and Zhang (2005). However, R3 field failure data for its first 6 months of deployment are recorded as 19 observations with cumulative observed failures equal to 22 and cumulative exposure time of 64,390

3. Numerical applications

In this section, using the comparison criteria like mean square error (MSE), predictive ratio risk (PRR) and predictive power (PP), a parallel comparison of our imperfect debugging models is carried out. The models are also compared to some famous pre-existing models. Six sets of actual software failure data obtained from actual projects have been used. Table lists comparison criteria used in this study. Formula notations include m_i is the total number of cumulated errors cumulated between 0 and t_i, $m(t_i)$ is the estimated number of errors at t_i obtained from fitting of the mean value function of the proposed model, n is the number of observations and k is the number of parameters (Table 4).

Each of these comparison criteria indicates a better fit of model than other when run on same data-set, if the criterion value is smaller. The following Tables 5–7 present criterion (MSE, PRR and PP) value and rank in a value (rank) combination format. Here and onwards, we have used shorter names of models listed in Table 1 and references are also available in Table 1 to avoid repetitions.

Table 3. Comparison of the estimated values of select model parameters

Model	**Parameters**					
	[1]	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**
Goel–Okumoto (G–O) (Goel & Okumoto, 1979)	$a = 135.974$, $b = 0.138$	$a = 33.6$, $b = 0.063$	$a = 133.761$, $b = 0.015$	$a = 18.257$, $b = 0.397$	$a = 124.44$, $b = 0.051$	$a = 23.092$, $b = 0.559$
Yamada–Ohba Delayed S-Shaped (Yamada et al., 1983)	$a = 124.673$, $b = 0.356$	$a = 26.715$, $b = 0.212$	$a = 101.909$, $b = 0.051$	$a = 17.007$, $b = 1.104$	$a = 112.43$, $b = 0.145$	$a = 21.371$, $b = 1.462$
Ohba inflection S-Shaped (Ohba, 1984)	$a = 135.974$, $b = 0.138$, $c = 1.000E\text{-}5$	$a = 24.821$, $b = 0.367$, $c = 14.34$	$a = 133.723$, $b = 0.146$, $c = 0.001$	$a = 18.25$, $b = 0.4$, $c = 0.01$	$a = 124.428$, $b = 0.051$, $c = 0.001$	$a = 22.252$, $b = 0.825$, $c = 0.675$
Yamada imperfect debugging (Yamada et al., 1992)	$a = 89.894$, $b = 0.289$, $\alpha = 0.021$	$a = 32.727$, $b = 0.065$, $\alpha = 0.001$	$a = 82.717$, $b = 0.259$, $\alpha = 0.044$	$a = 13.119$, $b = 0.843$, $\alpha = 0.023$	$a = 83.169$, $b = 0.097$, $\alpha = 0.007$	$a = 22.983$, $b = 0.563$, $\alpha = 0.001$
P–Z model (Pham & Zhang, 2003)	$a = 115.855$, $b = 0.741$, $c = 51.528$, $\alpha = 0.058$, $\beta = 0.239$	$a = 6.738$, $b = 0.387$, $c = 18.612$, $\alpha = 0.139$, $\beta = 12.86$	$a = 152.138$, $b = 0.857$, $c = 29.526$, $\alpha = 0.077$, $\beta = 0.001$	$a = 13.745$, $b = 2.91$, $c = 7.044$, $\alpha = 0.144$, $\beta = 0.01$	$a = 116.787$, $b = 1.014$, $c = 21.451$, $\alpha = 0.032$, $\beta = 0.507$	$a = 22.705$, $b = 20.452$, $c = 0.179$, $\alpha = 0.588$, $\beta = 6.758$
Chiu–Huang–Lee learning (Chiu et al., 2008)	$a = 135.965$, $\alpha = 0.138$, $\eta = 1.000E\text{-}4$	$a = 24.821$, $\alpha = 0.024$, $\eta = 0.343$	$a = 133.496$, $\alpha = 0.146$, $\eta = 0.001$	$a = 18.254$, $\alpha = 0.397$, $\eta = 0.001$	$a = 124.171$, $\alpha = 0.051$, $\eta = 0.001$	$a = 22.252$, $\alpha = 0.493$, $\eta = 0.332$
Model-1	$a = 126.965$, $\alpha = 0.016$, $\beta = 35.958$, $\eta = 0.079$	$a = 23.989$, $\alpha = 0.000$, $\beta = 1.785$, $\eta = 0.171$	$a = 85.605$, $\alpha = 0.009$, $\beta = 17.248$, $\eta = 0.323$	$a = 36.305$, $\alpha = 1.160$, $\beta = 0.482$, $\eta = 0.267$	$a = 204.833$, $\alpha = 0.036$, $\beta = 1.313$, $\eta = 1.222$	$a = 24.106$, $\alpha = 0.596$, $\beta = 1.668$, $\eta = 1.527$
Model-2	$a = 125.769$, $c = 22.914$, $\alpha = 12.03$, $\beta = 0.092$, $\eta = 5.599$	$a = 20.42$, $c = 4.956$, $\alpha = 0.057$, $\beta = 0.205$, $\eta = 0.3$	$a = 153.145$, $c=7.812$, $\alpha = 849.369$, $\beta = 0.099$, $\eta = 75.109$	$a = 13.735$, $c = 7.058$, $\alpha = 2.882$, $\beta = 0.143$, $\eta = 0.001$	$a = 118.970$, $c = 16.04$, $\alpha = 1.514$, $\beta = 0.035$, $\eta = 0.710$	$a = 22.701$, $c = 0.186$, $\alpha = 2.58$, $\beta = 0.588$, $\eta = 17.666$

Table 4. Comparison criteria

1.	Mean square error (MSE)
	$\text{MSE} = \frac{\sum_{i=1}^{n}(m_i - m(t_i))^2}{n-k}$
2.	Predictive-ratio risk (PRR) (Pham, 2007)
	$\text{PRR} = \sum_{i=1}^{n}\left(\frac{(m(t_i)-m_i)^2}{m(t_i)}\right)$
3.	Predictive power (PP) (Pham, 2007)
	$\text{PP} = \sum_{i=1}^{n}\left(\frac{(m(t_i)-m_i)^2}{m_i}\right)$

4. Discussion on results

As can be seen from the Table 5, on the basis of MSE criterion Model-2 that uses exponential fault content function outperforms Model-1 that uses linear fault content function. In fact Model-2 performs best among all pre-existing models for datasets 3 and 4 and ranks third for datasets 1 and 5. Rank 4 is the worst Model-2 has got for datasets 2 and 6. This means that from MSE values, Model-2 provides significantly better estimation than any other model for datasets 3 and 4.

The second comparison criterion used is PRR. The PRR values for Model-2 outrank Model-1 as can be seen from Table 6. Here also, Model-2 outranks all other models for datasets 3, 4 and 5. Model-2

Table 5. Comparison of MSE values

Tabulation of MSE values and their ranking for models comparison

Model/data-set	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**
G-O	33.309 (4)	4.952 (6)	8.62 (4)	2.685 (4)	35.102 (4)	0.717 (2)
Delayed S-shaped	134.233 (8)	1.464 (4)	45.783 (8)	6.593 (7)	129.457 (8)	1.828 (6)
Inflection S-shaped	34.834 (5)	0.504 (1)	9.129 (5)	2.851 (6)	35.397 (5)	0.651 (1)
Imperfect debugging	11.141 (2)	5.2 (7)	8.553 (3)	0.849 (2)	9.913 (1)	0.766 (3)
P–Z model	5.945 (1)	0.569 (2)	7.521 (2)	0.214 (1)	12.071 (2)	0.797 (4)
Chiu–Huang-learning	34.846 (6)	0.504 (1)	9.143 (6)	2.838 (5)	35.981 (6)	0.651 (1)
Model-1	52.888 (7)	2.849 (5)	16.666 (7)	0.964 (3)	63.632 (7)	1.408 (5)
Model-2	12.084 (3)	0.725 (3)	4.968 (1)	0.214 (1)	13.307 (3)	0.797 (4)

Table 6. Comparison of PRR values

Tabulation of PRR values and their ranking for models comparison

Model/data-set	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**
G–O	0.474 (4)	1.507 (4)	0.555 (4)	54.612 (5)	2,791.46 (4)	0.346 (5)
Delayed S-shaped	12.801 (6)	2.045 (6)	22.672 (7)	62,613.3 (8)	8.9E + 09 (8)	7.299 (6)
Inflection S-shaped	0.474 (4)	0.475 (1)	0.555 (4)	55.018 (7)	2,792.98 (5)	0.294 (3)
Imperfect debugging	0.055 (2)	1.511 (5)	0.365 (3)	18.27 (4)	1,674.55 (3)	0.346 (5)
P–Z model	0.474 (4)	0.475 (1)	0.557 (5)	54.677 (6)	2,816.48 (6)	0.294 (3)
Chiu–Huang-learning	0.015 (1)	0.773 (2)	0.18 (2)	2.978 (2)	499.805 (2)	0.166 (1)
Model-1	0.833 (5)	1.008 (3)	1.42 (6)	11.282 (3)	14,107.598 (7)	0.298 (4)
Model-2	0.065 (3)	2.721 (7)	0.024 (1)	2.959 (1)	165.156 (1)	0.168 (2)

Table 7. Comparison of PP values

Tabulation of PP values and their ranking for models comparison

Model/data-set	**[1]**	**[2]**	**[3]**	**[4]**	**[5]**	**[6]**
G–O	0.259 (4)	4.402 (6)	0.242 (4)	2.594 (5)	9.233 (4)	0.722 (4)
Delayed S-shaped	1.18 (6)	0.69 (4)	1.317 (6)	4.971 (8)	24.733 (8)	1.433 (6)
Inflection S-shaped	0.259 (4)	0.226 (1)	0.242 (4)	2.602 (7)	9.236 (5)	0.395 (3)
Imperfect debugging	0.046 (2)	4.418 (7)	0.181 (3)	1.525 (4)	6.184 (3)	0.726 (5)
P–Z model	0.259 (4)	0.226 (1)	0.242 (4)	2.595 (6)	9.288 (6)	0.395 (3)
Chiu–Huang- learning	0.015 (1)	0.305 (2)	0.111 (2)	0.595 (2)	2.502 (2)	0.165 (1)
Model-1	0.381 (5)	2.453 (5)	0.463 (5)	1.216 (3)	14.273 (7)	0.309 (2)
Model-2	0.082 (3)	0.547 (3)	0.023 (1)	0.594 (1)	2.195 (1)	0.165 (1)

ranks second for data-set 6 and ranks third for data-set 1. However, for data-set 2, Model-2 performs worst. Model-1 ranks third for datasets 2 and 4.

On the basis of PP values from Table 7, Model-2 predicts better than all other models for four out of six used datasets i.e. for datasets 3, 4, 5 and 6 and predicts third best for remaining two datasets. This is indicative of significant predictive power of Model-2 over rest of the models. Model-1 ranks second for data-set 6 and third for data-set 4.

For sake of drawing conclusion for parallel comparison of Model 1 and Model-2, it may be noted that Model-2 always outranks Model-1 for the given six datasets. Therefore for the datasets used in this study, it is observed that exponential imperfect debugging situations are better suited for use of model-2.

5. Conclusion

We present two new software growth models by considering two imperfect debugging situations represented by linear and exponential fault content functions. The mean value functions of the some famous existing and the two considered models have been used to present estimated parameters for six datasets. Three comparison criteria that include MSE, PRR and PP have been used to present a parallel comparison between the two considered models and among some famous existing models.

Based on the discussion on analysis of models using these comparison criteria on six actual failure datasets, it is concluded that the proposed Model-2 which always outperforms Model-1, fits the actual failure data better and has better predictive power than other considered SRGMs for at least two data sets (data-set 3 and 4). In fact Model-2 has better predictive power for four out of six data sets considered in this study. This indicates that for the given assumptions, given failure datasets and comparison criteria, Model-2 will perform better than other models for each comparison criterion for at least two datasets, which is significant enough to validate the models.

However, future research needs to be carried out on extending the existing forms of imperfect debugging situations or finding new such forms to better represent realistic imperfect debugging situations.

Funding
The author received no direct funding for this research.

Author details
Javaid Iqbal[1]
E-mail: iamjavaid@gmail.com
[1] Department of Computer Sciences, University of Kashmir, Srinagar, 190006, India.

Citation information
Cite this article as: Software reliability growth models: A comparison of linear and exponential fault content functions for study of imperfect debugging situations, Javaid Iqbal, *Cogent Engineering* (2017), 4: 1286739.

References

Bai, C., Hu, Q., Xie, M., & Ng, S. (2005). Software failure prediction based on a Markov Bayesian network model. *Journal of Systems and Software,* 275–282. http://dx.doi.org/10.1016/j.jss.2004.02.028

Carnes, P. (1997). Software reliability in weapon systems. *Proceedings of 8th International Symposium on Software Reliability Engineering*, pp. 114–115. http://dx.doi.org/10.1109/ISSRE.1997.630855

Chiu, K., Huang, Y., & Lee, T. (2008). A study of software reliability growth from the perspective of learning effects. *Reliability Engineering and System Safety, 93*, 1410–1421. http://dx.doi.org/10.1016/j.ress.2007.11.004

Goel, A. L., & Okumoto, K. (1979). Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability, R-28*, 206–211. http://dx.doi.org/10.1109/TR.1979.5220566

Hossain, S., & Dahiya, R. C. (1993, December). Estimating the parameters of a non-homogeneous Poisson process model for software reliability. *IEEE Transactions on Reliability, 42*, 604–612. http://dx.doi.org/10.1109/24.273589

Jeske, D., & Zhang, X. (2005). Some successful approaches to software reliability modeling in industry. *Journal of Systems and Software,* 85–99. http://dx.doi.org/10.1016/j.jss.2003.10.024

Lin, C., & Huang, C. Y. (2008). Enhancing and measuring the predictive capabilities of testing-effort dependent software reliability models. *Journal of Systems and Software, 81*, 1025–1038. http://dx.doi.org/10.1016/j.jss.2007.10.002

Lyu, M. R. (1996). *Handbook of software reliability engineering*. New York, NY: McGraw-Hill.

Misra, P. (1983). Software reliability analysis. *IBM Systems Journal, 22*, 262–270. http://dx.doi.org/10.1147/sj.223.0262

Musa, J., Iannino, A., & Okumoto, K. (1987). *Software reliability measurement, prediction, application*. New York, NY: McGraw-Hill.

Ohba, M. (1984). Inflexion S-shaped software reliability growth models. In *Stochastic models in reliability theory* (pp. 144–162). Berlin: Springer-Verlag.

Pham, H. (2003). Software reliability and cost models: Perspectives, comparison, and practice. *European Journal of Operational Research, 149*, 475–489. http://dx.doi.org/10.1016/S0377-2217(02)00498-8

Pham, H. (2007, October). An imperfect-debugging fault-detection dependent-parameter software. *International Journal of Automation and Computing, 4*, 325–328. http://dx.doi.org/10.1007/s11633-007-0325-8

Pham, H., Nordmann, L., & Zuemei Zhang, X. (1999, June). A general imperfect-software-debugging model with S-shaped fault-detection rate. *IEEE Transactions on Reliability, 48*, 169–175. http://dx.doi.org/10.1109/24.784276

Pham, H., & Zhang, X. (2003). NHPP software reliability and cost models with testing coverage. *European Journal of Operational Research, 145*, 445–454.

Rothermel, G., Untch, R., Chengyun Chu, C., & Harrold, M. (2001). Prioritizing test cases for regression testing. *IEEE*

Transactions on Software Engineering, 27, 929–948. http://dx.doi.org/10.1109/32.962562

Schneidewind, N. F., & Keller, T. W. (1992). Applying reliability models to the space shuttle. *IEEE Software, 9*, 28–33. http://dx.doi.org/10.1109/52.143099

Wood, A. (1996). Predicting software reliability. *Computer, 29*, 69–77. http://dx.doi.org/10.1109/2.544240

Yamada, S., Ohba, M., & Osaki, S. (1983). S-shaped reliability growth modeling for software error detection. *IEEE Transactions on Reliability,* 475–484. http://dx.doi.org/10.1109/TR.1983.5221735

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

Zhang, X., & Pham, H. (1998). A software cost model with warranty cost, error removal times and risk costs. *IIE Transactions,* 1135–1142.

Zhang, X., Teng, X., & Pham, H. 2003. Considering fault removal efficiency in software reliability assessment. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 33*, 114–120. http://dx.doi.org/10.1109/TSMCA.2003.812597