International Journal of Mathematical, Engineering and Management Sciences
Vol. 2, No. 3, 140–149, 2017
https://dx.doi.org/10.33889/IJMEMS.2017.2.3-013

Testing Domain Dependent Software Reliability Growth Models

Deepika[1], Ompal Singh[2], Adarsh Anand[*]
Department of Operational Research
University of Delhi, Delhi 110007, India
[1]deepika.sre@gmail.com, [2]drompalsingh1@gmail.com
[*]*Corresponding author*: adarsh.anand86@gmail.com

Jyotish N. P. Singh
Ramjas College
University of Delhi, Delhi 110007, India
jyotishdu@gmail.com

(Received October 31, 2016; Accepted January 1, 2017)

Abstract
Software Reliability Growth Models (SRGMs) are supporting software industries in expecting and scrutinizing quality of software. Numerous SRGMs have been proposed; majority of which concentrate on testing period of software. For testing, domain specific knowledge plays a very crucial role. Based on necessity condition, a set of programmes are in testing phase of software development. "Domain testing is a software technique in which small number of test cases is selected for trial. These sets of testing paths, all of which are to be eventually influenced by designed test cases are called the testing domain which expands with the progress of testing". Keeping this concept in mind, we propose SRGMs with the concept of testing domain with exponential coverage. Utility of proposed framework has been emphasized in this paper through some models pertaining to different distribution i.e Exponential, Logistic, Weibull and Rayleigh. Moreover, the data analysis is performed to find the estimates of parameters by fitting the models on authentic data sets.

Keywords: SRGMs, Detection rate of faults, Distribution function, Density function, Testing domain.

1. Introduction

Software is solitary essential medium which is motivating almost all electronics and industrial stuffs. In various fields, enormous software systems have been developed as computer systems have been utilized at the same time (Rafi et al., 2012). Most significant stage of Software Development Life Cycle (SDLC) is testing. One important ingredient for software is reliability where it is defined by quality (Fujiwara and Yamada, 2001). As the testing goes on, bugs are observed and disconnected from the software to make the quality enhancement. In order to perform software reliability assessment quantitatively, SRGM (Musa et al., 1987; Pham, 2000; Yamada, 1994) is well known as an effective tool. Mathematical models are constructed which are useful in describing the real time testing environment. Plenty of models are proposed to measure the software failure process successfully (Kapur et al., 2011). Some of them based on Non-Homogenous Poisson Process (NHPP) models are proposed to predict the future failures (Anand et al., 2016).

"In the testing phase of software development, there is a set of modules and functions in a software system to be influenced by executed cases (Othera et al., 1990). The test cases executed on the software in the testing phase are developed to influence the faults lying dormant in various modules/functions implemented in the software based on the requirement specifications. These test cases indeed influence a set of testing paths of these modules and functions. This set of testing paths, all of which are to be eventually influenced by designed test cases is called the testing domain" (Kapur et al., 2011). "The domain of testing which gets influenced by the test cases executed by any time in the testing phase is called the isolated testing domain by that time". An imperative role of testing domain is in reliability consideration and judgment related to the testing and operational phase (Yamada and Takahashi, 1993). The Measurement of software reliability with respect to the isolated testing domain is a region in SRGMs, which has not been greatly reconnoitred by the researchers. There are three types reliability examination of testing domain function studied in the prior research i.e. the basic testing domain, testing domain with skill factor and testing domain with imperfect debugging (Yamada and Fujiwara, 2001); and then these types of domain were very much connected to behaviour of fault recognition method of software testing (Kapur et al., 2011).

In this manuscript, SRGMs related to exponential growth curve have been discussed with testing domain. The quantity of bugs diverges with time and testing domain continues to stretch in software during the development. Further, we have taken fault detection rate dependent on time using four types of distributions i.e. Exponential, Logistic, Weibull and Rayleigh.

Respite of the manuscript is prearranged as follow: Segment 2 comprises notations related to SRGMs. In segment 3, modeling framework has been discussed. The parameters have been endorsed on factual data sets; consequently, obtained results and conclusion is accumulated in segment 4 and 5 respectively.

2. Notations

$z(t)$	Total number of bugs existing in the isolated testing domain at the testing time t
ω	Testing domain growth rate
κ	Total fault content
$\kappa(t)$	Total fault content dependent on time t
θ	Constant parameter
$\Psi(t)$	Mean value function
$\rho(t)$	Fault detection rate dependent on time t
$F(t)$	Cumulative distribution function for fault removal/ correction times
$f(t)$	Probability density function for fault removal/ correction times
ρ	Constant fault detection rate
ν	Learning parameter

3. Modeling Framework

3.1 Assumptions

These are following postulates which provide illustrative description of the proposed SRGMs (Kapur et al., 2011).

- In the testing domain, the dormant faults are disseminated evenly.
- The debugging process is perfect.
- The escalating rate of number of detectable faults is straight comparative to the number of faults remaining in the software outside of the isolated domain at randomly time.

With above assumption differential equation for testing domain can be written as

$$\frac{dz(t)}{dt} = \omega(\kappa - z(t)) \tag{1}$$

with the initial condition $z(0)=0$, we simplify Eq^{n} (1) then following expression for basic testing domain function is

$$z(t)=\kappa(1-e^{-\omega t}) \tag{2}$$

where $z(t)$ represent basic testing domain with the exponential growth curve and in equation (2), the quantity $(1-e^{-\omega t})$ signifies the isolated testing domain ratio in the software system at time *t*. $\Omega(t)$ representing the rate of the isolated testing domain growth in the software system is given as

$$\Omega(t)=\frac{d}{dt}z(t)=\kappa\omega e^{-\omega t} \tag{3}$$

3.2 Dynamic Testing Domain: Need and Importance

The definition of basic testing domain has been constructed under the postulation of perfect debugging. Introduction of new faults during the debugging progression is recurrently knowledgeable. “In the early stage of testing, correction of detected faults is simple and the influenced region of modules and functions by the fault correction is very conical” (Yamada and Fujiwara, 2001). With the evolution of testing, the exclusion of detected bugs becomes convoluted and prejudiced area extends broadly. This period requires much cautious and capable debugging deed for the rectification of additional complicated faults as compared to the early stage (Kapur et al., 2011).

While developing the basic isolated testing domain function, we assume that faults are constant over SDLC. However, in practical scenario it is also possible that the number of faults diverge with time and the testing domain continues to increase in software during the development (Kapur et al., 2011). In this circumstances κ of equation (1) is replaced by $\kappa(t)$.

$$\frac{dz(t)}{dt}=\omega(\kappa(t)-z(t)) \tag{4}$$

An exponential form of fault content is used to incarcerate the slow preface rate in the early phase and higher in the later phase i.e.

The following functional form of $\kappa(t)$ is used (Kapur et al., 2011)

$$\kappa(t)=\kappa e^{\theta t} \quad , \qquad \theta>0 \tag{5}$$

using equation (5) in equation (4), we see

$$\frac{dz(t)}{dt}=\omega(\kappa e^{\theta t}-z(t)) \tag{6}$$

with initial condition $z(0)=0$, we get

$$z(t)=\frac{\omega\kappa e^{\theta t}}{(\theta+\omega)}\left[1-e^{-(\theta+\omega)t}\right] \tag{7}$$

clearly basic testing domain is a special case of dynamic testing domain for $\theta=0$.
The growth rate (From equation (7) shows as

$$\Omega(t)=\frac{d}{dt}z(t)=\frac{\omega\kappa}{\omega+\kappa}\left[\theta e^{\theta t}+\omega e^{-\omega t}\right].$$

3.3 Testing Domain Dependent Model Development

These are following assumptions for SRGMs (Fujiwara and Yamada, 2001):

- The observed faults are considered to survive in the isolated domain.
- The proportion of isolated testing domain progresses with the time.
- Rate of fault detection is proportional to the number of bugs remaining domain time *t*.

Based on above assumption, the differential equation with respect to $\Psi(t)$ and testing domain function is

$$\frac{d\Psi(t)}{dt}=\frac{f(t)}{1-F(t)}\left[z(t)-\psi(t)\right] \tag{8}$$

where $\frac{f(t)}{1-F(t)}$ is the hazard rate function.

if $\frac{f(t)}{1-F(t)}=\rho(t)$ then equation (8),

$$\frac{d\Psi(t)}{dt}=\rho(t)\left[z(t)-\psi(t)\right] \tag{9}$$

In above equation (8), $F(t)$ follows different kind of distribution:

SRGM-1

$F(t)\sim$ Exponential distribution i.e. $1-e^{-\rho t}$

Thus $\frac{f(t)}{1-F(t)}=p(t)=p\approx constant$

using above calculation, we can write equation (8) as

$$\frac{d\Psi(t)}{dt}=\rho\left[z(t)-\psi(t)\right] \tag{10}$$

Now from the equation (7) value of $z(t)$ put in equation (10), we get

$$\frac{d\Psi(t)}{dt} = \rho\left[\frac{\omega\kappa e^{\theta t}}{(\theta+\omega)}(1-e^{-(\theta+\omega)t}) - \psi(t)\right] \tag{11}$$

using the initial condition $\psi(0)=0$, using testing domain function, is

$$\psi(t) = \frac{\omega\rho\kappa}{(\theta+\rho)(\rho-\omega)}\left[1-\frac{1}{(\theta+\omega)}\left\{(\theta+\rho)e^{(\rho-\omega)t} - (\rho-\omega)e^{(\theta+\rho)t}\right\}\right] \tag{12}$$

equation (12) shows expected number of faults using testing domain function with exponential form and constant rate (using hazard rate technique).

SRGM-2

$F(t) \sim$ Logistic distribution i.e. $\dfrac{(1-e^{-\rho t})}{(1+\nu e^{-\rho t})}$

$\dfrac{f(t)}{1-F(t)} = \rho(t) = \dfrac{\rho}{1+\nu e^{-\rho t}} \approx$ Logistic rate

using above logistic rate, equation (8),

$$\frac{d\Psi(t)}{dt} = \frac{\rho}{(1+\nu e^{-\rho t})}\left[z(t)-\psi(t)\right] \tag{13}$$

from the equation (7) and equation (13), we get

$$\frac{d\Psi(t)}{dt} = \frac{\rho}{(1+\nu e^{-\rho t})}\left[\frac{\omega\kappa e^{\theta t}}{(\theta+\omega)}(1-e^{-(\theta+\omega)t}) - \psi(t)\right] \tag{14}$$

Under initial condition $\psi(0)=0$,

$$\psi(t) = \frac{\omega\rho\kappa e^{-\rho t}}{(\theta+\rho)(\rho-\omega)(1+\nu e^{-\rho t})}\left[1-\frac{1}{(\theta+\omega)}\left\{(\theta+\rho)e^{(\rho-\omega)t} - (\rho-\omega)e^{(\theta+\rho)t}\right\}\right] \tag{15}$$

Above equation (15) integrates learning phenomenon in testing domain.

SRGM-3

$F(t) \sim$ Weibull distribution i.e. $(1-e^{-bt^{\gamma}}) \qquad ;\gamma > 0$

where γ is shape parameter (or slope).

$\dfrac{f(t)}{1-F(t)} = \rho(t) = \gamma\rho t^{\gamma-1}$

using above rate, equation (8) can be written as

$$\frac{d\Psi(t)}{dt} = \gamma\rho t^{\gamma-1}\left[z(t)-\psi(t)\right] \tag{16}$$

by equation (7), value of $z(t)$ put in equation (16)

$$\frac{d\Psi(t)}{dt} = \gamma\rho t^{\gamma-1}\left[\frac{\omega\kappa e^{\theta t}}{(\theta+\omega)}(1-e^{-(\theta+\omega)t})-\psi(t)\right] \tag{17}$$

Solving equation (17) using the seed value $\psi(0)=0$, we get

$$\psi(t) = \frac{\rho\omega\kappa}{(\theta+\omega)}\left[\frac{e^{\theta t}}{\frac{\theta}{\gamma}t^{(1-\gamma)}+\rho} - \frac{e^{-\omega t}}{\left(\frac{-\omega}{\gamma}\right)t^{(1-\gamma)}+\rho}\right] \tag{18}$$

above equation shows mean value function using testing domain function with Weibull Fault detection rate.

SRGM-4

$F(t) \sim$ Rayleigh distribution i.e. $(1-e^{-\rho\frac{t^2}{2}})$

$$\frac{f(t)}{1-F(t)} = \rho(t) = \rho t$$

above discussed Rayleigh rate put in equation (8)

$$\frac{d\Psi(t)}{dt} = \rho t\left[z(t)-\psi(t)\right] \tag{19}$$

putting the value of $z(t)$ in equation (19)

$$\frac{d\Psi(t)}{dt} = \rho t\left[\frac{\omega\kappa e^{\theta t}}{(\theta+\omega)}(1-e^{-(\theta+\omega)t})-\psi(t)\right] \tag{20}$$

using initial condition $\psi(0)=0$, below equation (21) for Rayleigh rate

$$\psi(t) = \frac{\rho\omega\kappa e^{-b\frac{t^2}{2}}}{(\theta+\omega)}\left[\begin{array}{r}\frac{e^{\theta t+\frac{\rho t^2}{2}}}{(\theta+\rho t)}\left(t-\frac{1}{(\theta+\rho t)}\right)-\frac{e^{-\omega t+\frac{\rho t^2}{2}}}{(-\omega+\rho t)}\left(t-\frac{1}{(-\omega+\rho t)}\right)\\ +\left(\frac{1}{\theta^2}-\frac{1}{\omega^2}\right)\end{array}\right] \tag{21}$$

Following different distributions are used in all SRGMs (Anand et al., 2014):

a) ***Exponential Distribution:*** This distribution has a constant rate and it is broadly used in modeling of software reliability. It designates the uniform distribution of faults.

b) ***Logistic Distribution:*** It has an S-shaped representation that is widely used in reliability. It looks like the normal distribution in contour.

c) ***Weibull Distribution:*** This distribution is much used in reliability engineering. “It is a versatile distribution in that it can take on the characteristics of other type of distributions, based on the value of shape parameters”. We can say that “Weibull distribution is a generalization of the exponential distribution due to its flexible environment”.

d) ***Rayleigh Distribution*****:** It is the distribution of the magnitude of a two-dimensional random vector whose coordinates are independent, identically distributed.

4. Data Analysis, Validation and Comparison Criteria

To illustrate the estimation, we have taken out the data analysis of authentic software data sets. The parameters have been evaluated using Statistics Analytical Software (SAS). Data set:1 (DS-1) has been compiled for 19 weeks in which 42 bugs were observed (Wood, 1996). In second data set (DS-2), the software size was about 300KB and it was written in assembly language (Kanoun et al., 1991; Anand et al., 2017). In the entire execution period, a total of 461 faults have been removed in 81 weeks. SRGMs are measured through the capability of fitting the previous failure data and expecting the coming performance of faults (as shown in Fig. 1 and 2).

Estimation of parameters and judgment criteria results for DS-1and DS-2 of all models under deliberation can be sight through Table 1, Table 2, Table 3 and Table 4. It is clear from the tables that the value of R^2 for SRGM-1 is higher and value of SSE, MSE and Root MSE is lower in comparison with other models and provides better goodness of fit for DS-1. Similarly we can see that the value of R^2 for SRGM-3 is higher and value of SSE, MSE and Root MSE is lower in comparison with other models and provides better goodness of fit for DS-2.

Models	κ	ω	ρ	θ	ν	γ
SRGM-1	70.0	0.000164	0.117575	0.003000	-	-
SRGM-2	71.4	0.331913	0.088956	0.114635	6	-
SRGM-3	62.0	0.065743	0.756506	0.002000	-	1.087
SRGM-4	50.0	0.060000	0.000059	0.004659	-	-

Table 1. Estimated results for dataset 1

Models	SSE	MSE	Root MSE	R^2	Adj. R^2
SRGM-1	19.801	1.164	1.079	0.994	0.994
SRGM-2	21.729	1.358	1.165	0.993	0.993
SRGM-3	39.999	2.500	1.581	0.988	0.987
SRGM-4	80.608	4.741	2.177	0.977	0.976

Table 2. Comparison results for data set 1

Models	κ	ω	ρ	θ	ν	γ
SRGM-1	650	0.10097	0.1153	0.1835	-	-
SRGM-2	500	0.0600	0.0160	0.1045	1.04	-
SRGM-3	566	0.0243	0.990	0.00085	-	1.012
SRGM-4	500	0.0212	0.000074	0.00685	-	-

Table 3. Estimated results for data set 2

Models	SSE	MSE	Root MSE	R^2	Adj. R^2
SRGM-1	18429.3	236.0	15.371	0.987	0.986
SRGM-2	10405.4	131.7	11.476	0.993	0.993
SRGM-3	7068.7	91.80	9.581	0.995	0.993
SRGM-4	10328.0	132.4	11.507	0.993	0.993

Table 4. Comparison results for data set 2

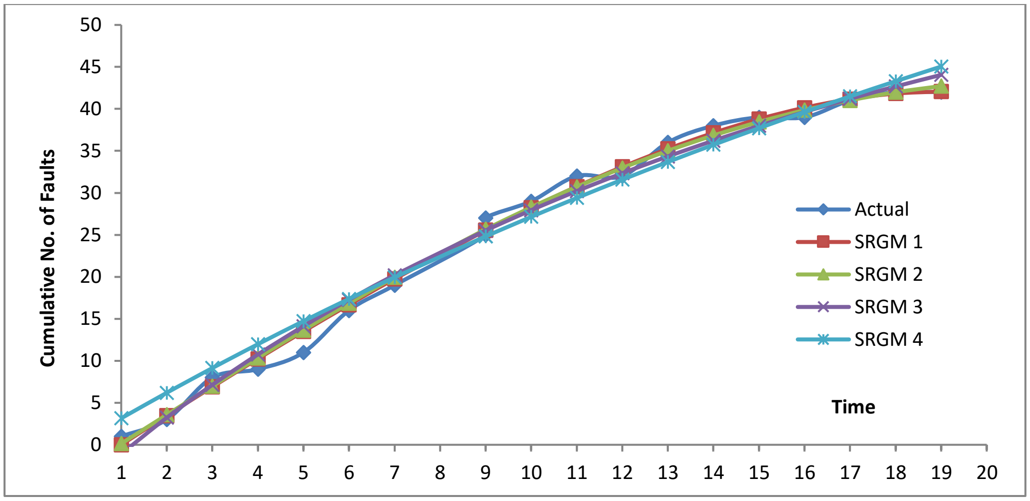

Fig. 1. Goodness of fit curve for data set 1

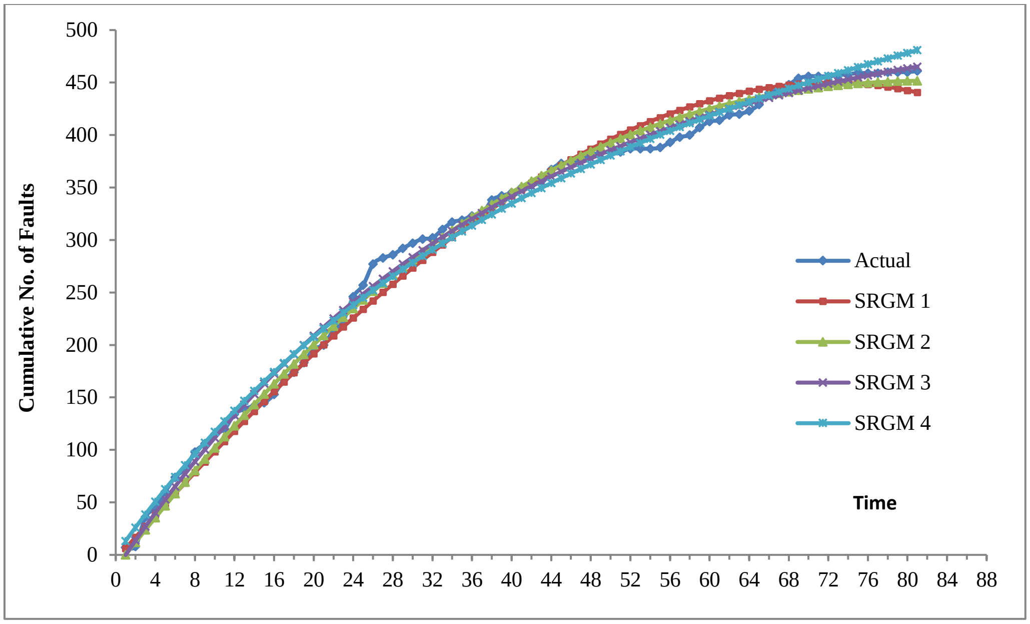

Fig. 2. Goodness of fit curve for data set 2

5. Conclusion

In this article, SRGMs based on the concept of testing domain with exponential coverage have been discussed. In the mathematical formulation, fault detection rate dependent on time has been taken. Standard distributions have used in this paper e.g. Exponential, Logistic, Weibull, Rayleigh for fault removal phenomenon. These SRGMs have been authenticated and validated on two software failures data sets. As can be viewed through Table 1 and Table 3, the obtained results are quite encouraging. For comparison analysis, we have different kind of comparison criteria as can be seen in Table 2 and Table 4. In future times the same concept can be applied to multi-release framework for software reliability growth modelling.

References

Anand, A., Bhatt, N., Aggrawal, D., & Papic, L. (2017). Software reliability modelling with impact of beta testing on release decision. In *Advances in Reliability and System Engineering* (pp. 121-138). Springer International Publishing.

Anand, A., Deepika, Singh, N. & Dutt, P. (2016). Software reliability growth modeling based on in house testing and field testing. *Communication in Dependability and Quality Management: An International Journal*, *19*(1), 74-84.

Anand, A., Kapur, P. K., Agarwal, M., & Aggrawal, D. (2014, October). Generalized innovation diffusion modeling & weighted criteria based ranking. In *Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2014 3rd International Conference on* (pp. 1-6). IEEE.

Fujiwara, T., & Yamada, S. (2001). Software reliability growth modeling based on testing-skill characteristics: Model and application. *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, *84*(6), 42-49.

Kanoun, K., de Bastos Martini, M. R., & De Souza, J. M. (1991). A method for software reliability analysis and prediction application to the TROPICO-R switching system. *IEEE Transactions on Software Engineering*, *17*(4), 334-344.

Kapur, P. K., Pham, H., Gupta, A., & Jha, P. C. (2011). *Software reliability assessment with OR applications*. London: Springer.

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

Othera, H., Yamada, S., & Narihisa, H. (1990). Software reliability growth model for testing-domain. *Trans. IEICE J73-D-1*, 170-174.

Pham, H. (2000). Software reliability. *Springer- Verlag, Singapur.*

Rafi, S. M., Rao, K. N., Setty, S. P., & Akthar, S. (2012). Joint effect of learning and testing effort in SRGM with fault dependent correction delay. *International Journal of Computer Science and Information Technologies*, *3*(5), 4961- 4967.

Wood, A. (1996). Predicting software reliability. *Computer*, *29*(11), 69-77.

Yamada, S., & Fujiwara, T. (2001). Testing-domain dependent software reliability growth models and their comparisons of goodness-of-fit. *International Journal of Reliability, Quality and Safety Engineering*, *8*(3), 205-218.

Yamada, S., & Takahashi, M. (1993). Introduction to software management model. *Kyoritsu-Shuppan, Tokyo, 284*.

Yamada, S., Software Reliability Models. (1994). Fundamental and application. *The Union of Japanese Scientists and Engineers (JUSE) Press, Tokyo.*