

SRGM with TEF for Imperfect Debugging in Software Systems

Dr. M.Jahir Pasha¹, V.Satish Kumar², Md.Asim³

¹ Associate Professor in department of Computer Science & Engineering, Dr. K.V.Subba Reddy College of Engineering For Women, Kurnool, A.P

² Assistant Professor in department of Computer Science & Engineering, Dr.K.V.Subba Reddy College of Engineering For Women, Kurnool, A.P., India

³ Assistant Professor in department of Computer Science & Engineering, Dr.K.V.Subba Reddy College of Engineering For Women, Kurnool, A.P., India

Article Info

Volume 83

Page Number: 8970 - 8976

Publication Issue:

March - April 2020

Article History

Article Received: 24 July 2019

Revised: 12 September 2019

Accepted: 15 February 2020

Publication: 09 April 2020

Abstract

The main goal of this proposed work is for analyzing FDP and FCP by incorporating two important processes namely TEF and imperfect debugging in software systems. To attain high quality reliability, it is required to test the system rigorously so that failures can be identified and correction can be done by applying tools called debuggers. As we know, testing effort function is predicted during this time by allocating the resources which influences considerably only for the fault identification rate and also for the correction of such faults.

Additionally, new faults may be included for evaluating as the feedback. In this technique, first it is proposed to demonstrate for the inclusion of TEF and fault introduction into FDP and later develop FCP as delayed FDP with a proper effort for correction. The FCP as well FCP as paired specific models which are extracted based on the basis of types of assumptions of introducing fault introduction as well as correction effort. In addition, the optimal policy of software release for different criteria with examples was also presented in this work.

Keywords; FDP, FCP, TEF, Fault.

I. INTRODUCTION

New generation computers and advanced networking technologies are transformed human day to day life. They are responsible for so many things changing drastically in the field of electronics, telecommunication and computer networks and technological changes created revolution in many areas like automation of manufacturing sectors, hospital management, transportation and office automation as well. In addition, the role of computers are diversified into many applications namely,

- In avionics for the control of air traffic
- Nuclear plants

- To get real time solutions in military
- Used for measurement and control in process industries.
- Used in Automotive industries for safety assessments
- Used in medical equipment for patient monitoring systems

In world's economy computer and computer networks play important and major roles in terms product manufacturing and services.

In this regard, the operations and maintenance of becomes tedious and time consuming in this modern world. The critical applications and functionality of software size and their complexities also leads to several challenges for the users and manufactures. Hence, design and development of reliability software is an important issue for finding software systems faults.

It is the fact that, two major components are incorporated in computer system computer systems they are hardware and software components. As on today, there is a tremendous research work carried out on testing of hardware reliability but there is no much focus for the development of SRGMs even though increase of software applications. As per the expert's opinion, reliability of software is entirely different from reliability of hardware in terms design and usability etc. If the software flaws and design were incorporated in the software system itself, human will not experience any software failures.

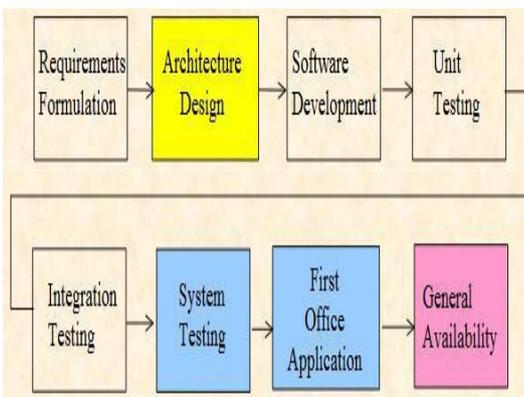


Figure 1.1 Architecture Based Reliability Models.

It is observed that, conventional SRGMs are applied to system test dataset to get the software failure information that can be monitored in the field. Perfect debugging assumption is an acceptable assumption. Therefore, the below challenges are taken into account while designing the SRGMs.

Utilization of different profile compare to testing profile.

Removal of faults in the instantaneous environment. Providing quality data for software

reliability. Commonly, software faults are more tedious and time consuming to identify the faults as compared to hardware or physical faults. In principle, the software will be made available free from faults, unlike components of hardware systems. However, it is noticed that, all design failures were there during the installation of the software into the hardware system [1] and [2]. For critical applications, computer programs vary between thousands and millions and this will make the fault decision. The above circumstances and other similar events have made it inevitable to predict and evaluate the software reliability [3] and [4] before applying for different tasks.

1.2 Fault $m_d(t) = aF(t)$ Introduction using FDP and FCP Models

Number of failures predicted during the time t can be expressed by the function $a(t)$ which is known as fault function. This function is equal to sum initial failures number in the software and number of faults introduction at the interval $[0, t]$. In this case, $w(t)$ = Testing effort rate which depends on time and also for the CTE (cumulative testing effort) consumed till time t .

Let $d(t)$ = Mean value function for depiction of the predicted faults

Let $d(t)$ = Mean value function for depiction of the predicted faults detected at time t .

Let $\lambda d(t) = dm(t) / dt$ = Fault Intensity function.

For, the time interval $[t, t + \Delta t)$ with

testing effort function can be expressed as,

$$\lambda d(t) = dm(t) / d(t) = b(t) w(t) (a(t) - md(t))$$

where, $b(t)$ = Current fault detection rate per unit of testing effort at time t .

$w(t)$ = Current testing effort expenditure at time t .

Put $md(0) = 0$ in equation (1.1) we get,

$$m d(t) = a(t) - \exp \left\{ - \int_0^t b(x) w(x) dx \right\}$$

$$-\exp\{-\int_0^t a_1(x) \exp\{\int_0^x b(y) w(y) dy\}$$

dx

where, $a_1(x) = \frac{dm_d(t)}{dt} = aF(t)$.

dx

of $a(t), b(t)$ and $w(t)$. $\lambda_d(t)$ can be obtained by substituting (1.2) into the equation (1.1).

$$\lambda_d(t) = \frac{dm_d(t)}{dt} = a b(t) w(t) \exp\{-\int_0^t b(x) w(x) dx\} (1 + \int_0^t a_1(x) \exp\{\int_0^x b(y) w(y) dy\} dx) \quad (1.3)$$

Functions of the SRGMs were predicted in terms of these datasets. The value of mean function $m_r(t)$ is applied to define the removal of faults which was expected till the interval t and λ

where, $r(t) = \frac{dm_r(t)}{dt}$ = Fault removal intensity function [5] and [6].

In general, correcting various failures may need different quantity of investigating materials [7], [8]. Hence, the PDF and the CDF were used as random variables for correction effort.

$$m_r(t) = \int_0^t \lambda_d(y) F(W(t) - W(y)) dy \quad (1.4)$$

here, $F(W(t) - W(y))$ = Probability of fault identified at y is corrected before. Different $m_r(t)$ will be calculated using $m_d(t)$ for different $f(x)$ respectively. Then these parameters can be expressed by the following equation.

$$\begin{aligned} \lambda_r(t) &= \int_0^t \lambda_d(y) f(W(t) - W(y)) w(t) dy \\ &= \int_0^{W(t)} \lambda_d(W^{-1}(W(t) - x)) f(x) w(t) d(W^{-1}(W(t) - x)) \\ &= \int_0^{W(t)} \lambda_d(W^{-1}(W(t) - x)) f(x) w(t) \frac{dx}{w(W^{-1}(W(t) - x))} \end{aligned} \quad (1.5)$$

1.3 FDP and FCP Multi-release Modeling

The modeling method of FDP is single version software where NHPP of SRGM is incorporated with the cumulative faults.

Let $N(t)$ = De-tested faults and is assumed to follow a Poisson distribution

and $m_d(t)$ = MVF.

$$P\{N(t) = n\} = \frac{m_d^n(t)}{n!} e^{-m_d(t)}.$$

(1.6)

The, the Mean Value Function can be expressed by the following equation.

$$\begin{cases} \frac{dm_d(t)}{dt} = \lambda_d(t) = \frac{F'(t)}{1 - F(t)} [a - m_d(t)] \\ m_d(0) = 0 \end{cases}$$

here, $\lambda_d(t)$ = represents fault rate at the time of process test and

$F(t)$ = CDF and by computing the equation (1.7), the MVF may be expressed as,

$$m_d(t) = a[1 - \exp(-\gamma t)].$$

Assuming $F(t)$ as an exponential distribution, it is considered as popular Goel model.

Mean value function which is delayed (MVF) m_c^* can be defined by the following expressions.

$$\lambda_c^* = \begin{cases} \lambda_d(t - \Delta t), & \Delta t \leq t \\ 0, & \Delta t > t \end{cases}$$

$$m_c^* = \begin{cases} m_d(t - \Delta t), & \Delta t \leq t \\ 0, & \Delta t > t \end{cases}$$

Dai et al. [12], expressed the approach as expectation of the delayed failure rate. Accordingly, $\lambda_c(t)$ can be written as,

$$m_c(t) = \int_0^t \lambda_c(\tau) d\tau = \int_0^t \int_0^\tau \lambda_d(\tau-x) \cdot g(x) dx d\tau$$

$$= \int_0^t \int_x^t \lambda_d(\tau-x) \cdot g(x) d\tau dx$$

$$= \int_0^t m_d(t-x) \cdot f(x) dx = E[m_c^*]$$

$$m_d(t) = aF(t)$$

$$\lambda_c(t) = E[\lambda_c^*] = \int_0^t \lambda_d(t-x) \cdot g(x) dx$$

It has been justified that, $mc(t)=E[m^*c]$. Commonly applied TEF are discussed in following paragraph.

Constant TEF: In this case, $W(t)$ assumed to be constant value and defined

as,

$$w(t) = w \tag{1.15}$$

$$W(t) = wt \tag{1.16}$$

It is observed that, the overall testing effort utilized to be the positive infinity, when time t reaches to positive infinity. If TEF is valued then it is awarded as $w(t)=1$.

TEF in Weibull: Well fit most data will be studied which are often used in SRGM is called Weibull TEF. They are very flexible and TEF $W(t)$ with cumulative value is defined by the following equation.

$$W(t) = N(1 - \exp\{-\beta t^m\}) \tag{1.17}$$

here, N = Total testing effort which is expected and it is necessary for testing of software.

β, m = Scale and shape parameters respectively.

It must be noticed that, there is finite consumption of CTE and tends to N when t reaches positive infinity. Equation (1.17) is differentiated we get,

$$w(t) = N \beta m t^{m-1} \exp\{-\beta t^m\} \tag{1.18}$$

$$W(t) = N \int_0^t w(x) dx = N [1 - \exp\{-\eta t\}] \tag{1.19}$$

TEF Logistic: As an alternative Rayleigh curve or Logistic curve was first studied by Parr et al. [13]. This TEF logistics gives similar characteristics as Rayleigh curve, other than project initial step. The logistic cumulative TEF $W(t)$ can be defined by the following equation.

here, A = Constant value and η = rate of consumption for the expenditure of testing effort.

Identical as Weibull case, there is finite consumption of CTE consumed resulting in N when t reaches to the value of positive infinity. Applying derivatives to the equation (1.19) we get,

$$m_r(t) = \frac{NA\eta}{\exp\left\{\frac{\eta}{2} \left(\frac{t}{A} + \exp\left\{\frac{\eta}{2} t\right\}\right)\right\}} \tag{1.20}$$

\when, $t = \ln A/\eta$ when $w(t)$ gets its highest value.

1.4 Results and Discussion

The proposed SRGM uses dataset from the system T1

data of the RADC[14] This work analyses various TEF software dependent FCP and FDP with imperfect debugging. It is noticed that, during software testing stage, the resources were not allocated.

Table I illustrate the values of faults identification and faults corrected for 1st 21weeks during the first week of cumulative numbers. For the entire time span, totally 302.1CPU hours are utilized.

Table.I Data Set.

Weeks	CPU time in Hours	Number of cumulative faults detected	Number of cumulative faults corrected
1	4.10	2.0	1.0
2	4.31	2.0	2.0
3	2.01	2.0	2.0
4	0.61	3.0	3.0
5	2.31	4.0	4.0
6	1.30	6.0	4.0
7	1.81	7.0	5.0
8	14.71	16.0	8.0
9	25.20	29.0	24.0
10	4.6	31.0	18.0
11	9.6	42.0	19.0
12	8.6	44.0	32.0
13	29.6	55.0	38.0
14	22.2	69.0	56.0
15	39.7	87.0	75.0
16	26.02	99.0	86.0
17	25	111.0	98.0
18	31.3	126.0	118.0
19	30.01	132.0	129
20	12.67	135.0	133.0
21	5.02	136.0	137.0

In the testing process, total of 136 faults are identified and corrected all of them. If more amount of testing effort is consumed, debuggers may influence for the introduction of more faults because of the software code has experienced more changes.

The TEF and the fault introduction effect can be incorporated simultaneously on both FDP and FCP to get the influences of testing resource allocation. On the basis different assumptions on correction effort as well as fault introduction various paired FCP and FDP models are obtained.

It is also observed that, the proposed SRGM model is quite common and it is very flexible. Parameters of various types of TEF were computed and predicted by using LSE which is defined in the previous section in order select a TEF that best fits this.

Table II. Comparative analysis of experimental results for different TEF with estimated parameters.

TEF	Parameters	Bias	RMSE	RMSPE	Variance
Constant	W = 14.29	0.00048	12.11	12.11	12.11
	N = 407.0830				
Weibull	$\beta = 2.064E-4$	7.86	7.86	7.86	7.76
	M=2.923				
Logistic	N = 321.482	6.7838	6.7828	6.7818	6.7570
	$\eta = 0.3826$				
	A=423.788				

The reliability for the exposure time will be defined with respect to the following parameters.

Time of calendar

Time of Clock

Execution time of CPU

Number of test runs

Different types of TEF with estimated parameters were compared and recorded in Table II.

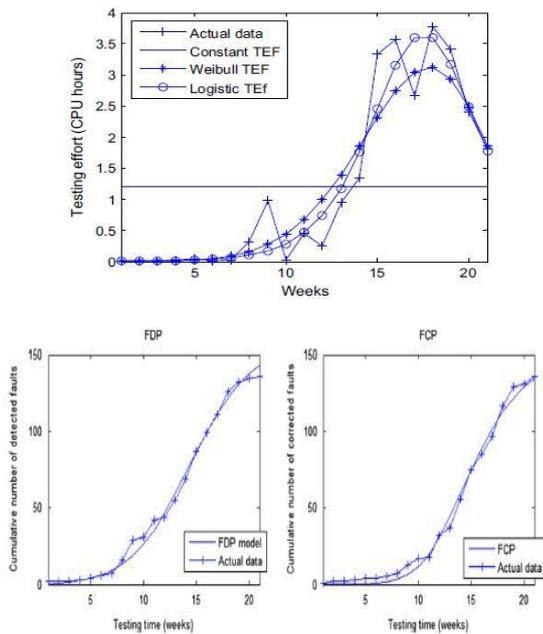


Figure 1.3 Paired model fitted against real database.

The fig. 1.2 shows graphical demonstration of different TEF. According to the predicted values, about 101.93 failure values at the starting of the experiment.

Total values of faults for time duration t reaches to 197.08. Figure 1.3 is plotted for graphical demonstration of the proposed model. In order to obtain influences for the allocation of resources for testing as well as for the introduction of faults on both FCP and FDP. The release time for optimal software $T = 39.626$. Figure 1.4 plotted for software release policy versus total reliability cost for graphical demonstration.

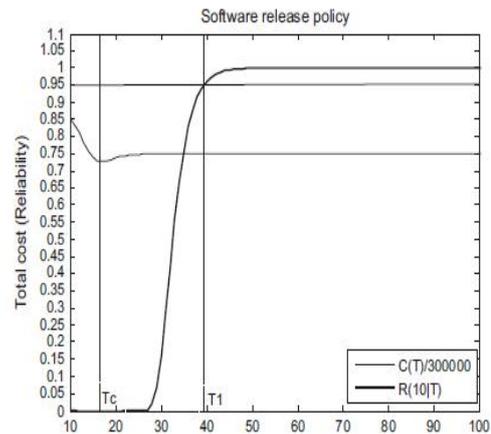


Figure 1.4 plotted for software release policy versus total reliability cost for graphical demonstration.

REFERENCES

1. Amin A, Grunske, Colman A, “An approach to software reliability prediction based on time series modeling,” *Journal of Systems Software*, 86 (7), pp. 1923–1932, 2013.
2. Rana R , Staron M , Berger C, Hansson J, Nilsson M, Törner F, Meding W, Höglund C, “Selecting software reliability growth models and improving their predictive accuracy using historical projects data,” *Journal of systems software*, 98, pp. 59–78. 2014.
3. Schneidewind, N. F, “Modeling the fault correction process. In: *Proceedings of the 12th International Symposium on Software Reliability Engineering*, pp. 185–190, 2001.
4. Smidts, C, Stutzke, M, Stoddard, R. W, “Software reliability modeling: an approach to early reliability prediction,” *IEEE Trans. Reliability*, 47(3), pp. 268–278, 2006.
5. Tamura, Y, Yamada, S, “Reliability assessment based on hazard rate model for an embedded OSS porting-phase,” *Software. Test Verification Reliability*, 23(1), pp. 77–88, 2013.
6. Wang J, Wu. Z, Shu. Y, Zhang Z, “An imperfect software debugging model considering log-logistic

distribution fault content function. *J. Syst. Software*, 100, pp. 167–181, 2015.

7. 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. Reliability*, 56 (4), pp.629–642, 2005.

8. Dai, Y, Xie, M, Long, Q, Ng, S. H, “Uncertainty analysis in software reliability modeling by Bayesian approach with maximum-entropy principle,” *IEEE Trans. Software Eng.* 33 (11), 781–795, 2007.