ISSN 2278 – 0211 (Online)

Generalized Modeling for Multiple Release of Two Dimensional Software Reliability Growth Model

A. K. Shrivastava
Research Development Center, Asia Pacific Institute of Management, New Delhi, India
Ruchi Sharma
Department of Computer Engineering, Netaji Subhash Institute of Technology, New Delhi, India

Abstract:
The amount of time spent on testing may not give a correct idea about the number of faults removed in the software. applied to remove the fault is also an important aspect. Therefore there is need of combining these aspects for modeling of software reliability growth models. This problem is solved by considering two dimensional framework. But In the existing literature for two dimensional software reliability growth model it was assumed that there is immediate correction of the failure which is not true. There is always a time lag between detection and correction of faults. Therefore, a generalized framework for modeling multi release of two dimensional software reliability growth models is proposed in this paper. A generalized approach using hazard rate function for developing the multi release models. Various continuous failure distributions are used to derive the models. The models developed are validated on real data set of tandem computers.

Keywords: Software Reliability Growth model (SRGM), Multi Release, Generalized framework, Two Dimension, fault detection process (FDP),fault correction process(FCP).

1. Introduction

Umpteen number of one dimensional SRGMs have been developed in the past [2, 10]. One dimensional software reliability framework depends either on testing coverage or testing time. Initially it was assumed that fault is corrected immediately after its detection i.e. fault detection and correction was considered as one stage phenomenon. Firstly, Scheindewind [7] explained the process of testing as a two-stage operation with a constant delay between fault detection and the process of correction. He also gave the notion of distinct modeling of FCP and FDP. Shanti Kumar [3] proposed a generalized birth process model. Xie and Wu et. al [5,6] generalized this idea and recommended NHPP based model for Fault detection process. In addition, he established the phenomenon of fault correction as a delayed process having random time lag. Later, Yamada et.al [15] also described testing as a two-stage process. Lo and Huang, [4] proposed a general framework from the correction perspective where some existing NHPP models were evaluated again. Then, Recently Peng et al. [13] proposed testing dependent software reliability model for imperfect debugging process considering both detection and correction. However, such SRGMs, which consider the effect of reliability growth factors on software reliability growth processes show dependence on the time spent on testing of the software. These models take into account the number of faults and hence reliability as a function of time but the depletion rate and pattern of resources such as manpower, computer time, and number of executed test cases etc is not considered. Therefore, a two dimensional modeling strategy can give a more realistic view of the process. In literature, Ishii and Dohi [17, 18] proposed a two-dimensional software reliability growth model and examined the dependence of test-execution time on the software reliability assessment. They quantitatively validated the SRGM with two-time scales. Inoue and Yamada [15, 16] also worked on two-dimensional software reliability growth models. However, their modeling framework does not showcase the use of mean value functions for representing the process of fault removal. They proposed a software reliability assessment method by using two-dimensional Weibull type SRGM. Later, Kapur et. al [8] worked towards a generalized approach for modeling two dimensional SRGM.

The firms now a days are adopting the practice of releasing the product functionalities in parts rather than unveiling them at once. This on one hand helps in maintaining the interest of customers and handles the stiff market contention on the other. Almost every software in the market today is following this trend. Be it android, Microsoft Office or Windows, one can see the new versions releasing periodically, sometimes to satisfy the needs of customers while sometimes to address the past errors. But in all, they are aimed at enriching the customer experience. Kapur et.al [12] proposed a dual dimension framework for modeling multiple up-gradations of

software. In their paper [12] they assume that detection and correction is one stage process which do not seems to be true in reality. We have improved this approach by proposing a more realistic scenario by considering detection and correction as two stage process.
In this paper, we present a two stage hazard rate based approach to propose a generalized framework for software reliability growth modeling for multiple releases of a software. Organization of the paper is as follows: section II describes the modelling framework. It includes the assumptions and notations used in the paper. In the subsequent part of this section, the unified framework for testing dependent software reliability growth model is described while considering the time differentiation between failure detection and fault removal processes. The modeling framework for software with multiple releases is suggested in section III. The parameter estimation and validation of data of the proposed model is depicted in section IV. Conclusion and future scope in this field is discussed in section V.

2. Modeling Framework

The development of a two dimensional model gives a more realistic view of the process of software reliability growth. It considers the influence of both testing effort and testing time on the number of bugs removed. As evident from the literature, the traditional one dimensional model showed dependence on the testing effort, testing time or testing coverage. But, a software reliability growth model based on the time spent by the testing team to test the software or the fraction of code covered during the process of testing does not fulfill the need of high precision software reliability. For this, we propose a two dimensional software reliability growth model which models the number of faults removed by combining the effect of testing time and testing effort. The basis of model suggested in this paper is Cobb Douglas production function. Mathematically, it can be represented as follows:

$$Y = AL^{v}K^{1-v}$$

Where:
Y = total production (the monetary value of all goods produced in a year), L = labor input, K = capital input, A = total factor productivity v is elasticity of labor. This value is constant and determined by technology available.

2.1. Notations

a	Total no. of faults present in the software at the beginning of testing.
$m(t)$	Expected no. of faults removed in the time interval (0, t).
b	Constant rate of fault detection/isolation/correction.
$f(t)/g(t)$	Probability density function for failure detection/correction
$F(t)/G(t)$	Probability distribution function for failure detection/correction
α_1/β_1	Shape/Scale parameter.
μ	Mean parameter
σ	Standard deviation parameter

2.2. Assumptions

a) Software system has a tendency to fail during execution due to remaining number of faults in the system.
b) Failure of the software is influenced equally by the faults in the software.
c) On encountering a failure, an instantaneous repair effort start sand the fault removal takes place with certainty.
d) From failure detection point of view, all the faults are mutually independent.
e) The process of fault detection/ correction are modeled by non-homogeneous poison process (NHPP).
f) There are finite number of faults at the initiation of testing phase.
g) There is a lag between failure detection and its correction.

In this paper, we suggest a growth model by combining the effect of testing effort and testing time. Two dimensional reliability models depicts a combined effect of two testing resources on the software reliability metric. Probability distribution functions govern the scheme proposed in this paper. The existing models and some new NHPP models can be obtained using our proposed approach. The model in our study is formulated using Cobb Douglas production function which shows the effect on number of faults removed owing to the testing time and testing effort of the software.
On the basis of above assumptions we get the differential equation as follows:

$$\frac{dm(t)}{dt} = h(t)[a - m(t)] \qquad (1)$$

Where $h(t) = \frac{(f * g)(t)}{[1-(F \otimes G)(t)]}$ is the failure observation/detection-fault removal/correction rate. On solving (1) under the initial condition of $m(t=0)=0$ we get mean value function given by Kapur et al. [12]

$$m(t) = a(F \otimes G)(t) \qquad (2)$$

In section 4, we presented various continuous statistical distributions and obtained various existing and new software reliability models with respect to testing resources using our proposed approach. These models are two dimensional in nature and depends both on testing time and testing effort.

2.3. Two-Dimensional Modeling

We used the Cobb–Douglas functional form of production functions which is widely used to represent the relationship inputs to output. In this paper, testing time and testing effort are taken as inputs affecting the output testing resources whenever a percentage change in the testing time or testing effort is encountered. So, in the 2D software reliability growth models, the number of faults removed depends on the testing resources τ . Testing resources are defined as follows:

$$\tau \cong s^{\alpha} u^{1-\alpha} \qquad 0 \le \alpha \le 1$$

Where

τ : testing resources

s : testing time

u : testing

α : Output elasticity of testing time

Using the value of τ we can write equation (2) as follows:

$$m(\tau) = a\ F\left(s^{\alpha} u^{1-\alpha}\right) \qquad 0 \le \alpha \le 1 \tag{3}$$

A. Modeling Multi Release of software

a) Release 1

Modeling of first release is same as equation (3) and it is given by

$$m_1\left(\tau\right) = a_1 F_1 \otimes G_1(\tau) \qquad 0 \le \tau < \tau_1 \tag{4}$$

$$\tau = s^{\alpha} u^{1-\alpha}$$

b) Next Release (i=2,3,4)

Considering the fault remaining from just previous release mean value function for next releases will be given by

$$m_i\left(\tau\right) = (a_i + a_{i-1}\left(1 - F_{i-1} \otimes G_{i-1}\left(\tau_{i-1}\right)\right) F_i \otimes G_i\left(\tau - \tau_{i-1}\right) \tag{5}$$

$$\tau_i = s_i^{\alpha_i} u_i^{1-\alpha_i} \quad ; \tau_{i-1} \le \tau < \tau_i$$

C. Derivation of Existing and New SRGM

For random failure detection/ correction, following probability distributions functions are used.

a) Exponential

Exponential distribution is one of the most widely used distribution in reliability engineering modeling because of its constant rate nature. Each and every fault has same probability for its removal and specifies a uniform distribution of faults in the software code.

b) Gamma / Erlang Gamma and Erlang

They are extensions of Exponential distribution where the process of fault removal takes place in a number of steps which include failure report generation, report analysis and correction time followed by verification and validation.

c) Normal

Various factors affects the process of fault correction. These factors can be internal or external. Internal factors can be defect density, fault complexity, internal structure of the software. The external factors can arise owing to the environment under which testing takes place. They could include test cases designs, tester skillset, availability of testers etc. This distribution can define the correction times for the cases where correction time is dependent on various factors.

Model	F(t)	G(t)	m(t)
SRGM-1	$t \sim \exp(b)$	$1(t)$	$a\left[1-e^{-bt}\right]$
SRGM-2	$t \sim \exp(b)$	$t \sim \exp(b)$	$a\left[1-\left((1+bt)e^{-bt}\right)\right]$
SRGM-3	$t \sim \exp(b_1)$	$t \sim \exp(b_2)$	$\left[1-\left\{\frac{1}{b_1-b_2}\left(b_1e^{-b_2t}-b_2e^{-b_1t}\right)\right\}\right]$
SRGM-4	t ~Erlang-2 (b)	$t \sim \exp(b)$	$a\left[1-\left(\left(1+bt+\frac{b^2t^2}{2}\right)e^{-bt}\right)\right]$
SRGM-5	$t \sim \exp(b)$	$t \sim N(\mu,\sigma^2)$	$a\left[\left(\varphi(t,\mu,\sigma)-e^{\left(-bt+\mu b+\frac{(b\sigma)^2}{2}\right)}\varphi(t,\mu+b\sigma^2,\sigma)\right)\right]$
SRGM-6	$t \sim \exp(b)$	$t \sim \gamma(\alpha,\beta)$	$a\left[\left(\Gamma(t,\alpha,\beta)-\frac{e^{-bt}}{(1-b\beta)^{\alpha}}\Gamma\left(t,\alpha,\frac{\beta}{1-b\beta}\right)\right)\right]$

Table 1: Mean Value function for different failure detection and correction distribution function

3. Numerical Example

The parameters of the proposed model are estimated using the dataset derived from wood [1]. It contains fault data of four subsequent releases of a software at Tandem Computers. These four software versions were tested for 20, 19, and 12 and 19 weeks and as a result 100, 120, 61 and 42 faults respectively were detected. Parameters are estimated using the nonlinear regression technique of least square using Statistical Package for Social Sciences. The estimated parameters presented in table II to VI. The results for comparison of various models are shown in table VII to XI. Goodness of fit curves for each release are given in the figures 1-4.

Release	a	b	α
1	103.852	.111	.863
2	122.449	0.046	.740
3	62.625	0.018	.510
4	41.220	.097	.877

Table 2: Estimation Results of SRGM1

Release	a	b_1	b_2	α
1	123.008	.018	.498	.735
2	134.861	.005	.067	.557
3	54.625	.018	.018	.510
4	48.761	.053	.153	.861

Table 3: Estimation Results of SRGM2

Release	a	b	α
1	98.918	.348	.967
2	118.624	.254	.927
3	63.072	.239	.868
4	41.760	.361	.953

Table 4: Estimation Results of SRGM3

Release	a	b	μ	σ	α
1	132.200	.003	1.224	.002	.475
2	150.501	.002	1.734	0.083	.472
3	70.550	.001	2.450	.083	.268
4	53.883	.010	7.259	.018	.673

Table 5: Estimation Results of SRGM4

Release	a	b	α_1	β_1	α
1	149.941	0.075	.877	.001	.269
2	124.298	0.014	1.377	.012	.622
3	65.312	.045	1.978	0.025	.570
4	44.518	0.066	1.821	.078	.864

Table 6: Estimation Results of SRGM5

A. Goodness of Fit criteria

The goodness of fit for a software reliability growth model suggests that "how well it fits a list of observed data". Different measures of goodness of fit summarize the disagreement between the values obtained with the proposed model and the values recorded from real software data.

a) Mean Square Error (MSE)

The difference between the actual data values, $\hat{m}(t_i)$ and the predicted values y_i is measured by MSE using the following formula.

$$MSE = \sum_{i=1}^{k} \frac{(\hat{m}(t_i) - y_i)^2}{k}$$

Where k represents the total number of observations.

b) Bias

The difference between the observed and predicted number of faults at any instant of time i is known as $PE_{i.}$ (Prediction error). The mean value of predictions errors is termed as bias. Lower the value of bias, better is the goodness of fit.

c) Variation

The standard deviation of prediction errors is termed as variation. A lower value indicates less fitting error.

$$Variation = \sqrt{\left(1/_{N-1}\right)\sum\left(PE_i - Bias\right)^2}$$

d) Root Mean Square Prediction Error (RMSPE)

It is a measure of closeness with which a model predicts the observation. A lower RMSPE value indicates less fitting error.

$$RMSPE = \sqrt{\left(Bias^2 + Variation^2\right)}$$

In all the above comparison criteria lower is the value better is the goodness of fit...

e) Coefficient of Multiple Determination (R^2):

The ratio of the sum of squares resulting from the trend model to that from constant model subtracted from 1 is denoted as the coefficient of multiple determination...

$$R^2 = 1 - \frac{Residual\ sum\ of\ square}{Corrected\ sum\ of\ square}$$

Higher the value of R^2 better is the goodness of fit.

SRGM-1	Release 1	Release 2	Release 3	Release 4
M.S.E.	26.826	17.164	6.128	1.234
R^2	.967	.990	.990	.995
Bias	-1.756	-2.331	-1.038	-.175
Variation	4.998	3.476	2.347	1.127
RMSPE	5.298	4.185	2.567	1.141

Table 7: Comparison Criteria Results of SRGM1

SRGM-2	Release 1	Release 2	Release 3	Release 4
M.S.E.	12.049	8.608	4.361	1.005
R^2	.985	.993	.990	.995
Bias	-0.402	-.407	-.458	-.057
Variation	3.537	2.985	2.128	1.028
RMSPE	3.560	3.013	2.177	1.030

Table 8: Comparison Criteria Results of SRGM2

SRGM-3	Release 1	Release 2	Release 3	Release 4
M.S.E.	50.600	29.797	5.372	1.717
R^2	.938	.977	.987	.991
Bias	-2.610	-1.940	-0.623	-.310
Variation	7.239	5.458	2.338	1.308
RMSPE	7.695	5.793	2.420	1.344

Table 9: Comparison Criteria Results of SRGM3

SRGM-4	Release 1	Release 2	Release 3	Release 4
M.S.E.	8.124	7.852	5.835	1.270
R^2	.990	.994	0.986	.993
Bias	-0.410	-0.056	-0.137	-0.066
Variation	2.894	2.878	2.519	1.156
RMSPE	2.923	2.879	2.523	1.156

Table 10: Comparison Criteria Results of SRGM4

SRGM-5	Release	Release	Release	Release
M.S.E.	6.292	6.565	4.410	.984
R^2	.992	.995	.990	.995
Bias	-.162	-.460	.453	-.040
Variation	2.568	2.590	2.142	1.019
RMSPE	2.573	2.630	2.189	1.019

Table 11: Comparison Criteria Results of SRGM4

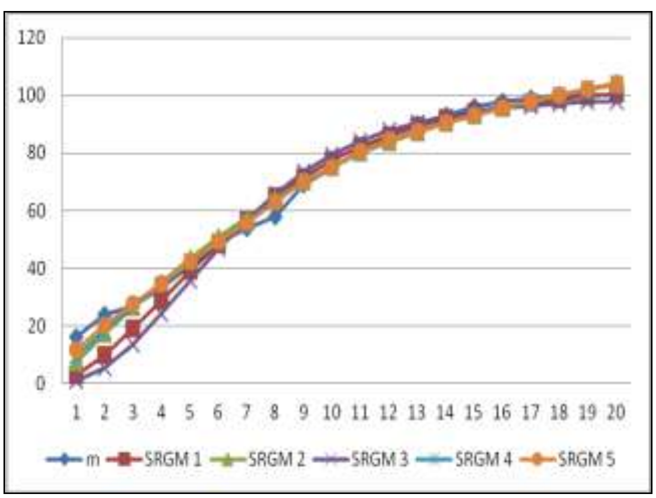

Figure 1: Goodness of fit curve for Release 1

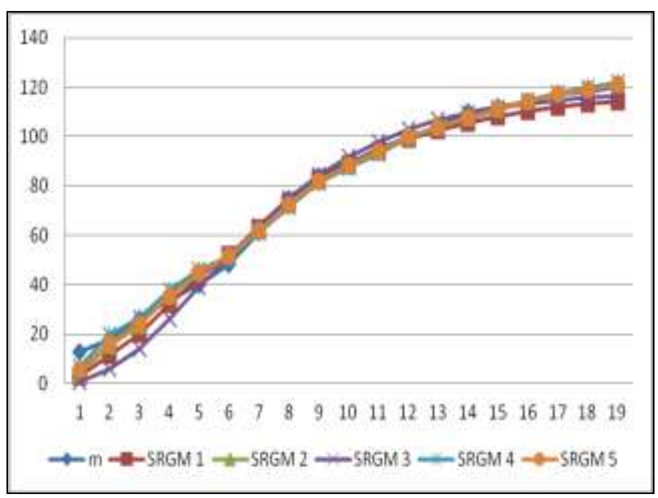

Figure 2: Goodness of fit curve for Release 2

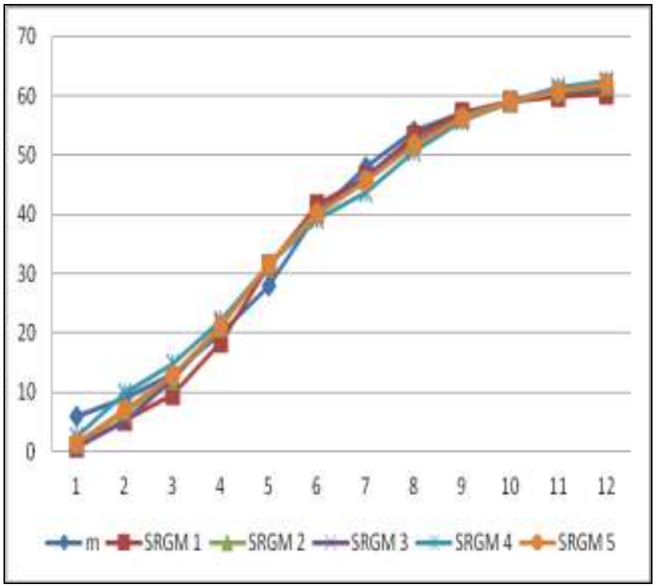

Figure 3: Goodness of fit curve for Release 3

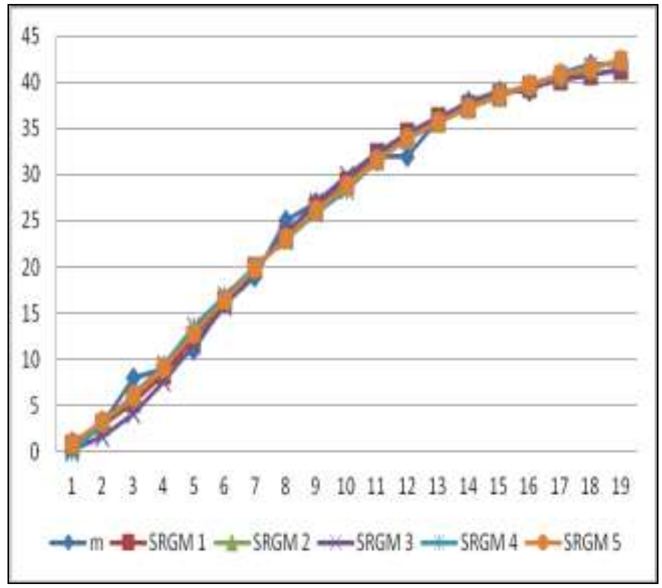

Figure 4: Goodness of fit curve for Release 4

Looking at the comparison criteria results given in table VII to XI we see that SRGM 5 fits best on the data set. This is due to the flexible nature of gamma distribution combined with exponential distribution. Second best SRGM for the data set is SRGM 4 which is obtained by combining exponential and Normal distributions. On the other hand SRGM 3 has poor fitting to the data set due to its S-Shaped nature.

4. Conclusion

In this paper we have proposed a generalized framework for modeling multi release of two dimensional SRGMs under the effect of FDP and FCP. The proposed approach suggests a framework for interaction between various dimensions of software reliability metrics. Testing time and testing coverage is used to designa a two dimensional framework in our study. To capture the joint effect of testing effort and testing time, we have used the Cobb-Douglas function in our proposed approach. Comparison criteria results are given for different set of mean value function obtained on combining different set of distribution functions.

Our work considers that the fault removal takes place with certainty, and no new faults are added during this process i.e. our work does not incorporates the phenomenon of imperfect debugging or error generation during testing. Also we have considered two dimesnions for modeling which may be extended for multiple dimensions. In future we can extend our model to incorportae the phenomenon of change-point and imperfect debugging in our model to make it a more realistic and generalized framework.

5. References

i. A. Wood; “Predicting Software Reliability”, IEEE Computers, Vol. 29(11), pp. 69-77,1996.
ii. Amir H. S. Garmabaki, Anu. G. Aggarwal, P. K. Kapur, V. S. S. Yadavali. (2012) Modeling Two-Dimensional Software Multi-Upgradation And Related Release Problem (A Multi-Attribute Utility Approach). International Journal Of Reliability, Quality And Safety Engineering 19:03.
iii. H.Pham,2006,System Software Reliability,Springer-Verlag.
iv. J.G. Shanthikumar, A General Software Reliability Model For Performance Prediction, Microelectronics Reliability, Vol. 21, pp. 671–682, 1981.
v. Lo Jung-Hua and Huang Chin-Yu,(2006) “An integration of fault detection and correction processes in software reliability analysis”, The Journal of Systems and Software, vol. 79, pp.1312-1323.
vi. M. Xie, Q.P. Hu, Y.P. Wu and S.H. Ng,(2006) “A study of the Modeling and Analysis of Software Fault-detection and Fault-correction Processes”, Quality and Reliability Engineering International, vol. 23, pp.459-470.
vii. Y. P. Wu, Q. P. Hu, M. Xie and S. H. Ng, “Modeling and Analysis of Software Fault Detection and Correction Process by Considering Time Dependency”, IEEE Transactions on Reliability, Vol. 56, No. 4, pp. 629-642, 2007.
viii. N.F. Schneidewind, Analysis Of Error Processes In Computer Software, Sigplan Notices, Vol. 10, pp. 337–346, 1975.
ix. Ompal Singh, Jyotish N.P. Singh, Jagvinder Singh and P.K. Kapur “Two-Dimensional Software Reliability Growth Model under the Effect of Imperfect Debugging and Error Generation: A Unified Scheme “ Published in International Journal of Mathematical Modeling, Simulation and Applications Vol. 5, No. 2, 2012 pp. 252-266
x. P.K. Kapur; Anu G. Aggarwal; Abhishek Tandon A unified approach for developing two-dimensional software reliability modelInt. J. of Operational Research ,2012 Vol.13, No.3,pp.318 – 337.
xi. P.K.kapur, H.Pham,.A. Gupta., P.C.Jha., (2011) Software Reliability Assessment with OR Applications. UK: Springer.

xii. P.K.Kapur, H.Pham.,S. Anand and K.Yadav "A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation", ieee transactions on reliability, vol. 60, no. 1, march 2011,331-340

xiii. P.K.Kapur,H. Pham, A.G. Aggarwal,G. Kaur, "Two Dimensional Multi-Release Software Reliability Modeling and Optimal Release Planning", Reliability, IEEE Transactions on, On page(s): 758 - 768 Volume: 61, Issue: 3, Sept. 2012.

xiv. R.Peng,Y.F.Li,W.J.Zhang,Q.P.Hu"Testingdependentsoftwarereliabilitymodelforimperfect debugging process considering both detection and correction" published in ReliabilityEngineeringandSystemSafety126(2014)37–43.

xv. S. Inoue. and S. Yamada, "Two-Dimensional Software Reliability Assessment with Testing-Coverage.", Second International Conference on Secure System Integration and Reliability Improvement. July 14-July 17, 2008.

xvi. Shinji Inoue and Shigeru Yamada "Two-Dimensional Software Reliability Measurement Technologies Published in IEEM 2009.

xvii. Tomotaka Ishii and Tadashi Dohi "Two-Dimensional Software Reliability Models and Their Application" Published in 12th Pacific Rim International Symposium on Dependable Computing (PRDC'06).

xviii. Tomotaka Ishii and Tadashi Dohi Two-Dimensional Software DefectModels with Test Execution History Published in ISRN Software Engineering 2012.