

Software Quality Control Truncated Time with Linear Failure Rate Distribution based on NHPP

B. Vara Prasad Rao

Associate Professor of Computer Science,
Department of Computer Science & Engineering,
R.V.R & J.C College of Engineering , Chowdavaram,
Guntur- 522 019, Andhra Pradesh, India

K. Gangadhara Rao

Associate Professor of Computer Science,
Department of Computer Science & Engineering,
Acharya Nagarjuna University,
Guntur- 522 010, Andhra Pradesh, India

ABSTRACT

Statistical process control is a method of monitoring product in its development process using statistical techniques with the presumption that the products produced under identical process condition shall not always be alike with respect to some quality characteristic(s). However, if the observed variations are within the tolerable limits statistical process control (SPC) methods would pass them for acceptance. This philosophy is adopted to decide the reliability and quality of a developed software by defining some quality measures and proposing a probability model for the quality measurements. The well known linear failure rate distribution (LFRD) is considered to propose a software reliability based on non-homogenous Poisson process (NHPP). Its mean value function is taken as a quality characteristic and SPC limits for it are developed. These control limits are exemplified to a live failure data to detect the out of control signals for the quality of the software based on the software failure data.

General Terms

NHPP- non homogenous poisson process
SRGM- software reliability growth model
MLE- maximum likelihood estimation
MSE- mean square error
IRD- inverse Rayleigh distribution

Keywords

IRD, MLE, MSE, NHPP, SRGM

1. INTRODUCTION

Like any manufactured product a developed software is also prone to failures for known or unknown reasons. A failed software can be debugged to bring it back to functioning through a testing process. In this procedure the data of observed software failures would throw some light on the quality of the software. There are various methods of measuring the software quality and the most popular among them is software reliability. Non homogenous Poisson processes are suitable models to compute software reliability in the statistical science. The earliest works in this direction can be attributed to those of Yamada *et al*(1986) [9], Wood(1996) [16],

Pham *et al*(1999) [8], Pham(2000) [11], Haung and Kuo(2002) [2], Pham and Zhang(2003) [13], Yamada *et al*(2003) [15], Yamada and Inoue(2004) [17], Huang(2005) [3], Pham(2005) [1], Quadri *et al*(2006) [10], Huang *et al*(2007) [18], Lan and Leemis(2007) [7]. Similar attempt relates to Kantam and Subbarao(2009) [4] - Pareto distribution, Srinivasa Rao *et al*(2011) [14] - Half logistic distribution, Prasad *et al*(2013) [12] - Inverse Rayleigh distribution. All these attempts are focussed on the mathematical model of the type

$$P(N(t+s) - N(t) = y) = \frac{e^{-\lambda s} (\lambda s)^y}{y!}, y = 0, 1, 2, \dots \quad (1)$$

where $N(t)$ indicates the random number of occurrences of an event in the interval $[0, t]$. This mathematical model indicates that the changes in $N(t)$ from one time period to another time period say $[t, t+s]$ depend only on the length of the interval s but not on the extremities $t, t+s$ of the interval. λ is called the failure intensity. In the above equation $E[N(t)] = \lambda t, \forall t$. If we think of a Poisson process whose mean depends on the starting t and also the length of the interval s such a Poisson process can be explained by an equation as

$$P(N(t) = y) = \frac{e^{-m(t)} (m(t))^y}{y!}, y = 0, 1, 2, \dots \quad (2)$$

In this equation $m(t)$ is a positive valued, non decreasing, continuous function and is called the mean value function. Equation (2) is called a Non Homogenous Poisson Process. If a software system when put to use fails with probability $F(t)$ before time t , if θ' stands for the unknown eventual number of failures that it is likely to experience, then the average number of failures expected to be experienced before time t is $\theta F(t)$. Hence $\theta F(t)$ can be taken as the mean value function of an NHPP. In the theory of probability, $F(t)$ is called the cumulative distribution function (CDF) of a continuous non negative valued random variable. Thus an NHPP designed to study the failure process of a software can be constructed as a Poisson process with mean value function based on the cumulative distribution function of a continuous positive valued random variable.

With this backdrop, we consider the well known linear failure rate distribution (LFRD) as $F(t)$ to generate software reliability growth model (SRGM) based Non Homogenous Poisson Process (NHPP). For such a model we developed the statistically admissible control limits for the mean value function and demonstrate the same how a

graphical procedure called a statistical process control (SPC) chart for the mean value function would help in detecting out of control signals for the software quality. The rest of the paper is organized as follows:

The basic distribution characteristics of linear failure rate distribution (LFRD) and moment type method of estimation of the parameters are presented in Section 2. The NHPP based on LFRD and its mean value function, the statistically tolerable limits for the failure time random variable and hence the mean value function are discussed along with an illustration for a live software failure data in Section 3. Summary and Conclusions are given in Section 4.

2. MOMENT TYPE METHOD OF ESTIMATION

In the present paper we consider the CDF of LFRD as the genesis of mean value function of our SRGM. In the theory of distributions a combination of exponential distribution which is CFR model and Rayleigh which is IFR model is used through hazard function to get a model called LFRD whose hazard function is a perfectly increasing straight line of the form $y=a+bx$. Such a distribution is proved to be having a number of important applications in survival analysis, a proxy concept to reliability theory with a view to model software failure data with LFRD.

The probability density function (pdf) of Linear Failure Rate Distribution is given by

$$f(x) = (a + bx)e^{-(ax + \frac{b}{2}x^2)}, x > 0, a > 0, b > 0 \quad (3)$$

Its cumulative distribution function (cdf) is

$$F(x) = 1 - e^{-(ax + \frac{b}{2}x^2)}, x > 0, a > 0, b > 0 \quad (4)$$

The NHPP with $F(\theta, x)$ as the mean value function is the SRGM for our present study and is given by

$$m(x) = \theta[1 - e^{-(ax + \frac{b}{2}x^2)}], x > 0, a > 0, b > 0 \quad (5)$$

Thus our proposed SRGM contains 3 parameters namely θ , a , b where θ stands for the unknown number of faults present in the software. It is also the limiting value of the mean value function as $t \rightarrow \infty$. For any general NHPP representing as SRGM the software reliability is given by

$$R(x/t) = P\{N(t+x) - N(t) = 0\} = e^{-[m(t+x) - m(t)]} \quad (6)$$

which is the probability of zero failures between the time t to $t+x$ where t is the execution time of the software during which testing was done and x is additional time period upto which the user wants the software to function failure free. The quality of the software is based on the magnitude of the software reliability. We can know it only if the parameters of SRGM are known and t, x are specified. But generally, the parameters remain unknown and need to be estimated with the help of software failure data. Usually, the parameters will be estimated using the classical maximum likelihood estimation (MLE) method. The loglikelihood equations to get the MLEs of the parameter after simplification for LFRD generated SRGM are:

$$\sum_{i=1}^n \frac{t_i e^{-at_i - \frac{b}{2}t_i} - t_{i-1} e^{-at_{i-1} - \frac{b}{2}t_{i-1}^2}}{e^{-at_{i-1} - \frac{b}{2}t_{i-1}} - e^{-at_i - \frac{b}{2}t_i^2}} (y_i - y_{i-1}) - \theta t_n e^{-at_n - \frac{b}{2}t_n^2} = 0 \quad (7)$$

$$\sum_{i=1}^n \frac{t_i^2 e^{-at_i - \frac{b}{2}t_i} - t_{i-1}^2 e^{-at_{i-1} - \frac{b}{2}t_{i-1}^2}}{e^{-at_{i-1} - \frac{b}{2}t_{i-1}} - e^{-at_i - \frac{b}{2}t_i^2}} (y_i - y_{i-1}) - \theta t_n^2 e^{-at_n - \frac{b}{2}t_n^2} = 0 \quad (8)$$

$$\theta = \frac{y_n}{1 - e^{-at_n - \frac{b}{2}t_n^2}} \quad (9)$$

In view of the complicated nature to get the solutions of loglikelihood equations, we resort to moment type of estimation of the parameters as provided in Kantam *et al* (2014) [5]. For a ready reference this method is presented below briefly:

The Mean, Variance and coefficient of variation (CV) of a reparameterised LFRD are respectively

$$\mu = \sqrt{\frac{2\pi}{b}} e^{\left(\frac{a^2}{2b}\right)\{1 - \Phi\left(\frac{a}{\sqrt{b}}\right)\}} \quad (10)$$

$$\sigma^2 = \frac{2}{b}(1 - a\mu) - \mu^2 \quad (11)$$

$$CV = \left(\frac{\left[\frac{2}{b} [1 - \sqrt{2\pi}\theta e^{\frac{\theta^2}{2}} (1 - \Phi(\theta)) - \pi(e^{\frac{\theta^2}{2}})^2 (1 - \Phi(\theta))^2] \right]^2}{\frac{2\pi}{b} (e^{\frac{\theta^2}{2}})^2 (1 - \Phi(\theta))^2} \right)^2 \quad (12)$$

where $\Phi(\theta)$ is cumulative distribution function of standard normal distribution. It can be seen that from equation (2.10) that there is a one-one correspondence between the population CV and θ of reparameterised LFRD. This motivates us to develop an auxiliary table between various hypothetical values of θ and CV expressed by equation (2.10). In fact the RHS of equation (2.10) is evaluated for various values of $\theta=0(0.001)0.5$, so that for any live value of coefficient of variation (CV) one can get back the corresponding θ , with interpolation if necessary. A part of these values corresponding to $\theta=0(0.001)0.5$ is listed in Table 1. The remaining values are available with the authors.

Tabl1. Auxiliary Table of CV for a given θ

θ	0.000	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
0.00	0.522723	0.523139	0.523556	0.523971	0.524387	0.524801	0.525215	0.525629	0.526042	0.526454
0.01	0.526866	0.527277	0.527688	0.528098	0.528508	0.528917	0.529326	0.529734	0.530142	0.530549
0.02	0.530955	0.531361	0.531767	0.532172	0.532576	0.532980	0.533384	0.533787	0.534189	0.534591
0.03	0.534992	0.535393	0.535793	0.536193	0.536592	0.536991	0.537389	0.537788	0.538184	0.538581
0.04	0.538977	0.539373	0.539768	0.540163	0.540557	0.540951	0.541344	0.541737	0.542129	0.542521
0.05	0.542912	0.543303	0.543693	0.544083	0.544472	0.544861	0.545249	0.545637	0.546024	0.546411
0.06	0.546797	0.547183	0.547569	0.547953	0.548338	0.548722	0.549105	0.549488	0.549871	0.550253
0.07	0.550634	0.551016	0.551396	0.551776	0.552156	0.552535	0.552914	0.553292	0.553670	0.554047
0.08	0.554424	0.554801	0.555177	0.555552	0.555927	0.556302	0.556676	0.557050	0.557423	0.557796
0.09	0.558168	0.558540	0.558911	0.559282	0.559653	0.560023	0.560392	0.560762	0.561130	0.561498
0.10	0.561866	0.562234	0.562601	0.562967	0.563333	0.563699	0.564064	0.564429	0.564793	0.565157
0.11	0.565520	0.565883	0.566246	0.566608	0.566969	0.567331	0.567692	0.568052	0.568412	0.568771
0.12	0.569130	0.569489	0.569847	0.570205	0.570563	0.570920	0.571276	0.571632	0.571988	0.572343
0.13	0.572698	0.573053	0.573407	0.573760	0.574113	0.574466	0.574818	0.575170	0.575522	0.575873
0.14	0.576224	0.576574	0.576924	0.577273	0.577623	0.577971	0.578319	0.578667	0.579015	0.579362
0.15	0.579708	0.580055	0.580400	0.580746	0.581091	0.581436	0.581780	0.582124	0.582467	0.582810
0.16	0.583153	0.583495	0.583837	0.584178	0.584519	0.584860	0.585200	0.585540	0.585879	0.586219
0.17	0.586557	0.586896	0.587234	0.587571	0.587908	0.588245	0.588581	0.588917	0.589253	0.589588
0.18	0.589923	0.590258	0.590592	0.590925	0.591259	0.591592	0.591924	0.592256	0.592588	0.592920
0.19	0.593251	0.593581	0.593912	0.594242	0.594571	0.594900	0.595229	0.595558	0.595886	0.596218
0.20	0.596544	0.596868	0.597194	0.597520	0.597846	0.598172	0.598497	0.598822	0.599146	0.599470
0.21	0.599794	0.600117	0.600440	0.600763	0.601085	0.601407	0.601728	0.602049	0.602370	0.602691
0.22	0.603011	0.603330	0.603650	0.603969	0.604287	0.604606	0.604924	0.605241	0.605558	0.605875
0.23	0.606192	0.606508	0.606824	0.607139	0.607455	0.607769	0.608084	0.608398	0.608712	0.609025
0.24	0.609338	0.609651	0.609963	0.610275	0.610587	0.610898	0.611209	0.611520	0.611830	0.612140
0.25	0.612450	0.612759	0.613068	0.613377	0.613685	0.613993	0.614301	0.614608	0.614915	0.615222
0.26	0.615528	0.615834	0.616139	0.616445	0.616750	0.617054	0.617359	0.617662	0.617966	0.618269
0.27	0.618572	0.618875	0.619177	0.619479	0.619781	0.620082	0.620383	0.620684	0.620984	0.621284
0.28	0.621584	0.621884	0.622183	0.622481	0.622780	0.623078	0.623376	0.623673	0.623970	0.624267
0.29	0.624564	0.624860	0.625156	0.625451	0.625746	0.626041	0.626336	0.626630	0.626924	0.627218
0.30	0.627511	0.627804	0.628097	0.628389	0.628682	0.628973	0.629265	0.629556	0.629847	0.630137
0.31	0.630428	0.630718	0.631007	0.631297	0.631586	0.631874	0.632163	0.632451	0.632739	0.633026
0.32	0.633313	0.633600	0.633887	0.634173	0.634459	0.634745	0.635030	0.635315	0.635600	0.635884
0.33	0.636168	0.636452	0.636736	0.637019	0.637302	0.637585	0.637867	0.638149	0.638431	0.638713
0.34	0.638994	0.639275	0.639555	0.639836	0.640116	0.640395	0.640675	0.640954	0.641233	0.641511
0.35	0.641790	0.642068	0.642345	0.642623	0.642900	0.643177	0.643453	0.643730	0.644006	0.644281
0.36	0.644557	0.644832	0.645107	0.64538	0.645655	0.645929	0.646203	0.646476	0.646750	0.647022
0.37	0.647295	0.647567	0.647839	0.64811	0.648382	0.648654	0.648924	0.649195	0.649465	0.649735
0.38	0.650005	0.650275	0.650544	0.65081	0.651081	0.651350	0.651618	0.651886	0.652153	0.652421
0.39	0.652688	0.652954	0.653221	0.65348	0.653753	0.654018	0.654284	0.654549	0.654814	0.655079
0.40	0.655343	0.655607	0.655870	0.65613	0.656397	0.656660	0.656923	0.657185	0.657447	0.657709
0.41	0.657971	0.658232	0.658493	0.65875	0.659014	0.659275	0.659535	0.659794	0.660054	0.660313
0.42	0.660572	0.660831	0.661089	0.66134	0.661605	0.661863	0.662120	0.662378	0.662634	0.662891
0.43	0.663147	0.663403	0.663659	0.66391	0.664170	0.664425	0.664680	0.664935	0.665189	0.665443
0.44	0.665697	0.665950	0.666204	0.666458	0.666709	0.666962	0.667214	0.667466	0.667718	0.667969
0.45	0.668221	0.668472	0.668722	0.668977	0.669223	0.669473	0.669723	0.669972	0.670220	0.670471
0.46	0.670719	0.670968	0.671216	0.67146	0.671712	0.671959	0.672207	0.672454	0.672700	0.672947
0.47	0.673193	0.673439	0.673685	0.67393	0.674176	0.674421	0.674666	0.674910	0.675155	0.675399
0.48	0.675643	0.675886	0.676130	0.67637	0.676616	0.676858	0.677101	0.677343	0.677585	0.677826
0.49	0.678068	0.678309	0.678550	0.67879	0.679031	0.679272	0.679512	0.679751	0.679991	0.680230
0.50	0.680469	0.680708	0.680947	0.68118	0.681423	0.681661	0.681899	0.682136	0.682373	0.682610

3. MONITORING THE TIME BETWEEN FAILURES FOR MEAN VALUE FUNCTION WITH CONTROL CHART

Let $F(x)$ be the cumulative distribution function of a continuous positive valued random variable, $f(x)$ be its probability density function. If the random variable is taken as representing inter failure time of a device, a control chart of such data would be based on 0.9973 probability limits of the times between failure random variable say t analogous to the Shewhart's theory of variable control charts. These limits and the central line are respectively the solutions of the following equations taking equi-tailed probabilities.

$$F(t) = 0.00135 \quad (13)$$

$$F(t) = 0.5 \quad (14)$$

$$F(t) = 0.99865 \quad (15)$$

Let t_U , t_C and t_L be respectively the solutions of equations (3.1), (3.2) and (3.3) in the standard form

$$t_L = F^{-1}(0.00135) \quad (16)$$

$$t_C = F^{-1}(0.5) \quad (17)$$

$$t_U = F^{-1}(0.99865) \quad (18)$$

The NHPP based on $F(t)$ as the mean value function is the SRGM for our present study and is given by

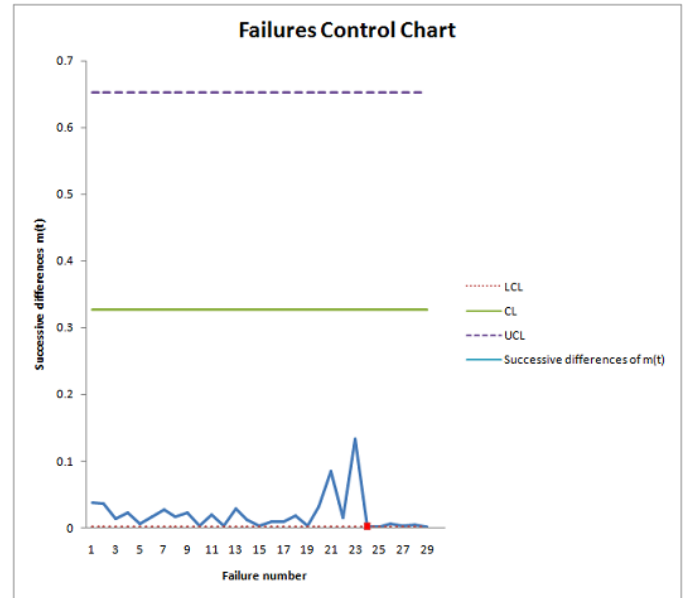
$$m(t_i) = \theta[1 - e^{-(at_i + \frac{b}{2}t_i^2)}], t > 0, a > 0, b > 0 \quad (19)$$

The time control chart based on the mean value function corresponding to inter failure time together with three parallel lines to the horizontal axis at t_L , t_C and t_U for the data of Kim(2013) [6] is given below.

Failure number	Failure time (hours)	Failure number	Failure time (hours)	Failure number	Failure time (hours)
1	9	11	71	21	116
2	21	12	77	22	149
3	32	13	78	23	156
4	36	14	87	24	247
5	43	15	91	25	249
6	45	16	92	26	250
7	50	17	95	27	337
8	58	18	98	28	384
9	63	19	104	29	396
10	70	20	105	30	405

Linear Failure Rate model					
\hat{a}	\hat{b}	$\hat{\theta}$	$m(t_L)$	$m(t_C)$	$m(t_U)$
0.4432612×10^{-2}	0.459371×10^{-4}	0.654	0.000883	0.327	0.653117

Estimated values of $m(t)$ at the given failure times t_1, t_2, \dots, t_n along with the successive differences of these estimates are given in Table 4. The successive differences would indicate the estimated number of failures between consecutive failure times. The graph through $[t_i, \Delta \hat{m}(t_i)]$ $i=1, 2, \dots, n-1$ along with three parallel horizontal lines at $m(t_L), m(t_C), m(t_U)$ would be the required failure control chart and is given in figure below.



Failure number	Failure time (hours)	$m(t)$	Successive differences of $m(t)$ $\Delta m(t)$	Failure number	Failure time (hours)	$m(t)$	Successive differences of $m(t)$ $\Delta m(t)$
1	9	0.026745	0.037390	16	92	0.295868	0.009255
2	21	0.064135	0.035544	17	95	0.305123	0.009157
3	32	0.099679	0.013133	18	98	0.314280	0.017998
4	36	0.112812	0.023156	19	104	0.332278	0.002957
5	43	0.135968	0.006644	20	105	0.335235	0.031657
6	45	0.142612	0.016637	21	116	0.366892	0.084206
7	50	0.159249	0.026620	22	149	0.451098	0.015612
8	58	0.185868	0.016580	23	156	0.466710	0.133399
9	63	0.202448	0.023057	24	247	0.600109	0.001679
10	70	0.225505	0.003274	25	249	0.601788	0.000823
11	71	0.228780	0.019520	26	250	0.602611	0.005470
12	77	0.248299	0.003230	27	337	0.608081	0.002192
13	78	0.251529	0.028708	28	384	0.610274	0.004125
14	87	0.280237	0.012525	29	396	0.614398	0.001301
15	91	0.292762	0.003106	30	405	0.615699	

The control limits are such that the point above the $m(t_U)$ (UCL; upper control limit) is an alarm signal. A point below the $m(t_L)$ (LCL; lower control limit) is an indication of better quality of software. A point within the control limits indicates stable process. In the figure, the first out of control situation is noticed at the 25th failure with the corresponding successive difference of $m(t)$ falling below LCL and hence a preferable out-of-control signal for the product. The assignable cause for this is to be investigated and needs to be promoted.

4. SUMMARY & CONCLUSIONS

There are many charts which use statistical techniques. It is important to use the best chart for the given data, situation and need. In this paper, the control chart for estimated number of failures in consecutive failure time intervals against the serial order of the failure interval is developed with the associated control lines and central line at same serial point on that of Kim(2013) [6]. However the basic model for our chart is a more generalized one allowing IFR as well as CFR failure behavior, whereas that of Kim(2013) [6] relates to only CFR model. Hence our conclusion is in a broader domain of models and hence can be generalized version than that of Kim(2013) [6].

5. REFERENCES

- [1] Hoang Pham. A generalized logistic software reliability growth model. *Operational Research Society of India(OPSEARCH)*, 42(4):322–331, Dec 2005.
- [2] Huang and Kuo.S.Y. Analysis of incorporating logistic testing effort function into software reliability modeling. *IEEE Transactions on Reliability*, 51(3):261–270, Sept 2002.
- [3] Huang.C.Y. Performance analysis of software reliability growth models with testing-effort and change-point. *Journal of Systems and Software*, 76(2), May 2005.
- [4] Kantam.R.R.L. and Subba Rao.R. Pareto distribution - A software reliability growth model. *International Journal of Performability Engineering*, 5(3):275–281, 2009.
- [5] Kantam.R.R.L., Priya.M.Ch., and Ravikumar.M.S. Moment type estimation in linear failure rate distribution. *IAPQR Transactions*, 39(1):87–97, 2014.
- [6] Kim.H.C. Assessing software reliability based on NHPP using SPC. *International Journal of Software Engineering and its Applications*, 7(6):61–70, 2013.
- [7] Lan.Y. and Leemis.L. The logistic exponential survival distribution. *Naval Research Logistics*, 55(3):252–264, 2007.
- [8] Pham.H., Nordmann.L. and Zhang.X. A general imperfect software debugging model with s-shaped fault detection rate. *IEEE Transaction on Reliability*, (48):169–175, 1999.
- [9] Yamada.S., Ohtera.H. and Narihisa.R. Software reliability growth models with testing effort. *IEEE Transactions on Reliability*, R-35:19–23, 1986.
- [10] Quadri.S.M.K.,Peer.M.A., Ahmad.N. and Kumar.M. Non homogeneous poisson process software reliability growth model with generalized exponential testing effort function. *RAU Journal of Research*, 16(1-2):159–163, 2006.
- [11] Pham.H. *Software Reliability*. Springer-Verlag, 2000.
- [12] Vara Prasad Rao.B., Gangadhar Rao.K., and Srinivasa Rao.B. Inverse Rayleigh software reliability growth model. *International Journal of Computer Applications*, 75(6),2013.
- [13] Pham.H. and Zhang.X. Non homogeneous poisson process software reliability and cost models with testing coverage. *European Journal of Operational Research*, (145):443–454, 2003.
- [14] Srinivasa Rao.B., Vara Prasad Rao.B., and Kantam.R.R.L. Software reliability growth model based on half logistic distribution. *Journal of Testing and Evaluation*, 39(6):1152–1157, 2011.
- [15] Yamada.S., Tamura.Y. and Kimura.M. A software reliability growth model for a distributed development environment. *Electronic and Communications in Japan*, 83(3):1446–1453, 2003.
- [16] Wood.A.P. Predicting software reliability. *IEEE Computer*, 11:69–77, 1996.
- [17] Yamada.S. and Inoue.S. Testing coverage dependent software reliability growth modeling. *International Journal of Quality, Reliability and Safety Engineering*, 11(4):303–312, 2004.
- [18] Yen.S.K., Huang.C.Y. and Michael.R.Lyu. An assesment of testing effort dependent software reliability growth models. *IEEE Transactions on Reliability*, 56(2):198–211, 2007.