Software Reliability Growth Model Based on Linear Failure Rate Distribution

B.Vara Prasad Rao
Associate Professor of Computer Science,
Department of Computer Science & Engineering,
R.V.R & J.C College of Engineering,
Andhra Pradesh, India.
e-mail : boyapativaraprasad@gmail.com

K.Gangadhara Rao
Professor Of Computer Science ,
Department of Computer Science & Engineering,
Acharya Nagarjuna University
Andhra Pradesh, India.
e-mail: kancharla123@gmail.com

Abstract—A Non Homogenous Poisson Process (NHPP) with its mean value function generated by the cumulative distribution function of linear failure rate distribution is considered. It is modeled to assess the failure phenomenon of a developed software. When the failure data is in the form of number of failures in a given interval of time the model parameters are estimated by moment type estimation method and the performance of the model using three data sets is discussed in comparison with similar other models.

Keywords- *LFRD,MLE,MSE,NHPP,SRGM*

*****

I. INTRODUCTION

It is well-known that computers are used in diverse areas for various applications. The growing importance of software dictates that a reliable software is by all means essential. A software itself does not fail unless the faults within the software result in its failure. Generally, software faults are more difficult to handle. All design faults are present from the time the software is installed in the computer. A software fault inherent in a program is not dangerous unless and until it results in a failure of software. Accordingly, the concept of software reliability is rather dependent on the failure of a software and its frequency rather than the unknown number of faults latent in the software. Therefore, the term software reliability may be defined as the probability of failure free functioning of a software rather than the faults contained in it. However we cannot rule out the fact that software reliability depends on the number of faults also. In this regard, theory of probability and hence statistical analysis have become essential in the development of a model that can be used to evaluate the reliability of real world software systems. Quantifying the software quality in terms of reliability is attempted through the study of software reliability growth models.

Software reliability models are statistical models which can be used to make predictions about a software system's failure rate,given the failure history of the system. The models make assumptions about a fault discovery and removal process. These assumptions determine the form of the model and the meaning of the model's parameters. Some recent works in this regard are by Akaike(1974) [1], Yamada et al (1986) [16], Huang et al (1999) [11], Pham et al (1999) [14], Huang et al (2000) [12], Kapur et al (2002) [3], Haung and Kuo(2002) [6], Pham and Zhang(2003) [20], Yamada et al (2003) [22], Yamada and Inoue(2004) [24], Huang(2005) [7], Huang and Lyu(2005) [8], Kapur et al (2005)[2], Pham(2005)[5], Quadri et al (2006)[17], Huang et al (2007)[25], Lan and Leemis(2007)[13]. With this backdrop, we study the modeling of software reliability as a Non Homogenous Poisson Process (NHPP) with mean value function based on linear failure rate distribution. Similar attempt based on Pareto distribution is made by Kantam and Subbarao(2009)[9] and that based on half logistic distribution is given by Srinivasa Rao et al (2011) [21] and that based on Inverse Rayleigh distribution is given by Prasad et al (2013) [19].

The genesis and the development of the model with the necessary input about a Non Homogenous Poisson Process are presented in section II Moment type method of estimation of the parameters of the LFRD and its application to the SRGM are discussed in section III. The proposed SRGM is then compared with other software reliability growth models generated by half logistic, gamma ane exponential distributions in section IV. The concept of cost aspect in developing a software,associated randomness and the optimum release time of a developed software with respect to cost aspect are given in section V. Summary and Conclusions are given in section VI.

ISSN: 2454-4248
92 – 101

II. SRGM AS A NON HOMOGENOUS POISSON PROCESS

Suppose that we are interested in observing the occurrences of a repeatable event over a period of time. The situation relevant here can be the number of times a developed software fails in a given period of testing/operational time. As failures do not occur in a predictable way such a failure process can be identified with a random counting process, generally defined as a count of number of events that have occurred in a specified interval of time. Let it be denoted by N(t), where t is any non negative real number. N(t) indicates the number of random occurrences in the interval [0,t]. A counting process is said to be a Poisson process if the failure has stationary independent increments and the number of failures in any time interval of length s has a Poisson distribution with mean λs given by

$$P(\mathrm{N(t+s)}-N(t)=y)=\frac{e^{-\lambda s}(\lambda s)^{y}}{y!}\quad y=0,1,2,3\ldots..(2.1)$$

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)] = λt, ∀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

$$P(\mathrm{N(t)}=y)=\frac{e^{-\mathrm{m(t)}}(\mathrm{m(t)})^{y}}{y!},\quad y=0,1,2,3\ldots.. \qquad (2.2)$$

In this equation m(t) is a positive valued, non decreasing, continuous function of t, generally tending to a finite limit 'a' as t → ∞, m(t) is called the mean value function and its derivative with respect to t is the intensity function λ(t). Equation (2.2) is called a Non Homogenous Poisson Process. If a software system when put to use fails with probability F(t) before time t, if 'a' 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 aF(t). Hence aF(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. The first and foremost of such models is due to Goel and Okumoto(1979) [4] which is based on the well-known exponential distribution. Later many such models have been suggested and studied by various researchers that can be found in Wood(1996) [23], Pham(2000) [18] and Huang et al (2007) [25] and references therein.

III. 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. All these models are either constant failure rate (CFR) or absolutely instantatenous failure rate (IFR) . 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 . We consider the pdf

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

$$f(x)=(a+bx)e^{-\left(ax+\frac{b}{2}x^{2}\right)},x>0,a>0,b>0 \qquad (3.1)$$

Its cumulative distribution function (cdf) is

$$F(x)=1-e^{-\left(ax+\frac{b}{2}x^{2}\right)},x>0,a>0,b>0 \qquad (3.2)$$

The NHPP with F(θ,x) as the mean value function is prepared by us as the SRGM for our present study.

$$F(\theta,x)=\theta[1-e^{-\left(ax+\frac{b}{2}x^{2}\right)}],x>0,a>0,b>0 \qquad (3.3)$$

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

$$R(x/t)=\mathrm{P}\{\,(\mathrm{N(t+s)}-\mathrm{N(t)}=\}=\;e^{-[m(t+x)-m(t)]} \qquad (3.4)$$

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 M.L.method. The loglihood equations to get the MLEs of the parameter after simplification for LFRD generated SRGM are:

$$\sum_{i=1}^{n} \frac{t_i\, e^{-at_i-\frac{b}{2}t_i} - t_{i-1}\, e^{-at_{i-1}-\frac{b}{2}t_{i-1}^2}}{e^{-at_{i-1}-\frac{b}{2}t_{i-1}} - e^{-at_i-\frac{b}{2}t_i^2}} (y_i - y_{i-1}) - \theta t_n\, e^{-at_n-\frac{b}{2}t_n^2} = 0 \quad (3.5)$$

$$\sum_{i=1}^{n} \frac{t_i^2 e^{-at_i-\frac{b}{2}t_i} - t_{i-1}^2 e^{-at_{i-1}-\frac{b}{2}t_{i-1}^2}}{e^{-at_{i-1}-\frac{b}{2}t_{i-1}} - e^{-at_i-\frac{b}{2}t_i^2}} (y_i - y_{i-1}) - \theta\, t_n^2\, e^{-at_n-\frac{b}{2}t_n^2} = 0 \quad (3.6)$$

$$\theta = \frac{y_n}{1 - e^{-at_n-\frac{b}{2}t_n^2}} \quad (3.7)$$

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) [10]. For a ready reference this method is presented below briefly: The Mean, Variance and coefficient of variation(CV) of a reparameterised LFRD are respectively

$$\mu = \sqrt{\frac{2\pi}{b}}\; e^{\left(\frac{a^2}{2b}\right)}\left(1-\phi\left(\frac{a}{\sqrt{b}}\right)\right) \quad (3.8)$$

$$\sigma^2 = \frac{2}{b}(1 - a\mu) - \mu^2 \quad (3.9)$$

$$CV = \left(\frac{\frac{2}{b}\left(1 - \sqrt{2\pi}\,\theta e^{\frac{\theta^2}{2}}\,(1-\phi(\theta))\right) - \pi\left(e^{\frac{\theta^2}{2}}\right)^2 (1-\phi(\theta))^2}{\frac{2\pi}{b}\left(e^{\frac{\theta^2}{2}}\right)^2 (1-\phi(\theta))^2} \right)^2 \quad (3.10)$$

where $\Phi(\theta)$ is cumulative distribution function of standard normal distribution. It can be seen that from equation(3.10) 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(3.10). In fact the RHS of equation (3.10) is evaluated for various values of θ=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 the values corresponding to θ =0(0.001)0.5 is listed in the table 1. The remaining values are available with the authors.

IV. Comparative Study

The present model can be compared with other models also w.r.t some criteria of preference. The standard models we considered here are those based on the

(i) Exponential cumulative distribution function.

(ii) Half logistic cumulative distribution function.

(iii) Gamma cumulative distribution function with shape parameter 2.

in succession. The first NHPP is called Goel -Okumoto model (1979) [4]. The second NHPP is software reliability growth model based on half logistic model(2011) [21]. The third NHPP is called

Yamada S-shaped software reliability growth model (1983) [15]. For a ready reference we give below the associated results of differentiation useful to get the ML estimates of the parameters in the three competitive models.

1.Exponential Distribution: (Goel-Okumoto(1979)[4] Model):

$$\frac{y_n t_n e^{-bt_n}}{1-e^{-bt_n}} = \sum_{k=1}^{n} \frac{t_k e^{-bt_k} - t_{k-1} e^{-bt_{k-1}}}{e^{-bt_{k-1}} - e^{-bt_k}} (y_k - y_{k-1})$$

$$a = \frac{y_n}{1-e^{-bt_n}}$$

ISSN: 2454-4248
92 – 101

Table 1. 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	0.596541	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

ISSN: 2454-4248
92 – 101

θ	0.000	0.001	0.002	0.003	0.004	0.005	0.006	0.007	0.008	0.009
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	0.655078
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	0.66645	0.666709	0.666962	0.667214	0.667466	0.667718	0.667969
0.45	0.668221	0.668472	0.668722	0.66897	0.669223	0.669473	0.669723	0.669972	0.670222	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

 ISSN: 2454-4248
92 – 101

Table 3. Estimated values of parameters,MSE and AIC

Linear failure rate distribution

	θ	a	b	MSE	AIC
DS1	0.494	0.163436×10^{-3}	0.1095×10^{-6}	18.6130	36.6582
DS2	0.106	0.314203×10^{-4}	0.879×10^{-7}	2.5956	52.2226
DS3	0.106	0.268885×10^{-4}	0.643×10^{-6}	4.5474	64.9840

Exponential distribution

	θ	a	b	MSE	AIC
DS1		63.8248	0.617×10^{-3}	65.3991	38.7225
DS2		26.0300	0.905×10^{-3}	136.7666	85.6727
DS3		43.0927	0.703×10^{-3}	314.4548	108.1386

Half logistic distribution

	θ	a	b	MSE	AIC
DS1		63.4795	0.775×10^{-3}	23.8514	35.2038
DS2		26.9414	0.539×10^{-3}	41.1789	57.3215
DS3		44.5194	0.464×10^{-3}	98.0323	72.2615

Gamma distribution

	θ	a	b	MSE	AIC
DS1		294.6153	0.167×10^{-3}	224.5987	61.9170
DS2		194.4502	0.85×10^{-4}	15.9128	53.1344
DS3		45.3157	0.540×10^{-3}	44.5361	68.8644

Table 5. Parametric estimates of testing times

Test time T	$\hat{\theta}$	$\hat{a}$	$\hat{b}$
9	0.439	0.0931897610	0.0450616740
10	0.525	0.1321511269	0.0633611605
11	0.620	0.0880194232	0.0201545730
12	0.695	0.0832765228	0.0143572543
13	0.908	0.0835792944	0.0084727760
14	0.879	0.0741290078	0.0071121133
15	0.906	0.0899778008	0.0098631205
16	0.831	0.0799411312	0.0092541929
17	0.752	0.0526556596	0.0049029160
18	0.678	0.0463150330	0.0046664281
19	0.596	0.0407489650	0.0046745585
20	0.516	0.0353531204	0.0046941414
21	0.462	0.0310891587	0.0045282873

Table 6. Expected total cost at testing times

Test time t	E(t)
9.5	250.0831
10.5	252.9958
11.5	258.2581
12.5	260.7160
13.5	266.0768
*14.5	264.3584
15.5	266.4871
16.5	263.9885
17.5	257.8971
18.5	255.1880
19.5	252.6191
20.5	250.1922

ISSN: 2454-4248
92 – 101

2.Half Logistic Distribution:(Srinivasa Rao et al (2011) [21] Model):

$$\sum_{i=1}^{n}\frac{\dfrac{t_i e^{-bt_i}}{(1+e^{-bt_i})^2}-\dfrac{t_{i-1}e^{-bt_{i-1}}}{(1+e^{-bt_{i-1}})^2}}{\hat{a}\left[\dfrac{-2e^{-bt_i}}{(1+e^{-bt_i})(1-e^{-bt_{i-1}})}\right]}(y_i-y_{i-1})=\frac{t_n e^{-bt_n}}{(1+e^{-bt_n})^2}$$

3.Gamma Distribution (Yamada(1983) [15] Model):

$$at_n{}^2e^{-bt_n}=\sum_{k=1}^{n}\frac{t_k{}^2e^{-bt_k}-t_{k-1}{}^2e^{-bt_{k-1}}}{\hat{a}\left[(1+bt_{k-1})e^{-bt_{k-1}}(1-bt_k)e^{-bt_k})\right]}(y_k-y_{k-1})$$

$$y_n=a[1-(1+bt_n)e^{-bt_n}]$$

Now, we adopt calculation of mean square error(MSE) and Akaike's information criterion (AIC) for model comparison. The formulae are respectively defined as

$$MSE=\frac{\sum_{i=1}^{n}[y_i-\hat{m}(t_i)]^2}{n-N}$$

where $\hat{m}(t)$ stands for MLE of m(t).

AIC=−2log(likelihoodfunctionatitsmaximumvalue)+2N(4.2)where N represents the number of parameters in the model. For the three data sets given in Table 2, we have computed the parameters by moment type estimation for LFRD and by M.L estimation for Exponential, Half logistic and Gamma models and the estimators of the mean value functions thereby the values of MSE and AIC for various models. The results are given in the Table 3. The MSEand AIC values obtained indicate the closeness measures in both cases is minimum for LFRD based SRGM compared with other models. We therefore say that LFRD based SRGM is the best fit to the three data sets considered in relation to the three competing models.

V. Optimal Release Policy

The cost of developing software leads to considerable expenses in a software system development. The quality of a software system usually depends upon the length of testing time. The more the testing time the more reliable the software is. However, the total cost of software development is also expected to increase. On the other hand, if the testing time is too short, though the cost of software development would be reduced we cannot avoid the customer's risk of receiving unreliable software which in turn leads to increase in cost during the operational phase. Testing is an efficient way to remove faults in software products but testing of all possible executable paths in a general program is impractical. To determine when to stop testing or when to release the software to customers keeping the expected total software cost at a minimum subject to warranty and risk is considered as an optimal release policy.

A cost model is essential to define important software cost factors. It should help software developers in scheduling of resources for prompt delivery. Moreover with a reasonably sufficient reliability the model should contribute to decide an appropriate release time of the software. With these objectives several software cost models are suggested (Pham (2000) [18],Chapter 6). In this section we adopt a software cost model with risk factor as discussed in Pham (2000) [18] . The adapted model is presented in the following lines for a ready reference. A software cost generally consists of the following components.

(i) cost to perform testing

(ii) cost incurred in removing errors during testing phase

(iii) risk cost due to software failure.

Testing cost is denoted by C_1t, where t is the total test time. C_1 is software test cost per unit time. If N(t) stands for number of errors detected by time t, expected time to remove all these errors is given by

$$E\left[\sum_{i=1}^{N(T)}Y_i\right]=E[N(t)]E[Yi]=m(t)\,\mu_y \qquad (5.1)$$

where Y_i is time to remove the i^{th} error during testing phase, m(t) is expected number of errors detected by time t given by LFRD mean value function. μ_y is expected time to remove an error during testing phase also called E(Y). Therefore the expected cost to remove all errors is given by $C_2m(t)\mu_y$ where C_2 is cost of removing each error per unit time during testing. The risk cost due to software failure, after releasing the software is

$$E_3(t)=C_3[1-R(x/t)] \qquad (5.2)$$

where C3 is cost due to software failure and R(x/t) is survival probability of the software by x units of time given it is tested

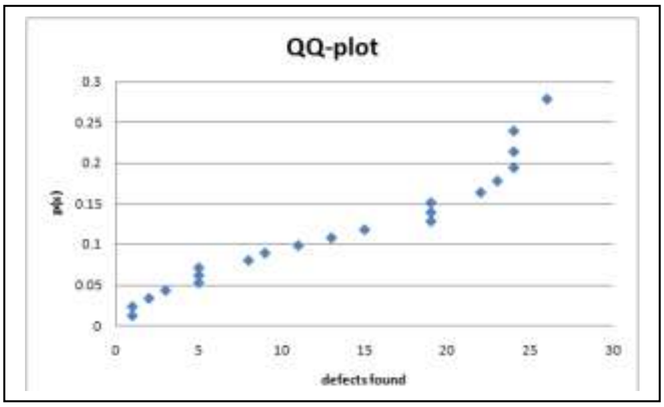
 ISSN: 2454-4248
92 – 101

for t units of time to be obtained by LFRD model. Therefore the total expected cost of software is given by

$$E(t)=C_1t+C_2m(t)\ \mu_y+C_3[1-R(x/t)] \qquad (5.3)$$

We have to find the value of t that minimizes the expected total cost in Equation (5.3). Such an optimal value of t is called optimal release time. In the expression for m(t) in Equation (5.3) we take the mean value function as given by LFRD and t has to be solved. The formula for such a t has to be compared with the value of t for a similar NHPP model say Goel & Okumoto(1979) [4], half logistic model (2011) [21] ,Yamada(1983) [15] etc. The expected cost function given in equation(5:3) will show an increasing trend and falls down at a certain time and then increases from there. The time instant at which the change in the trend is observed is taken as the optimal time at which the testing is to be stopped and the product is ready for release . This methodology of locating optimum release time is explained with the data set given in Table 4.

For the above data, the parametric values of LFRD are estimated as θ=0.462, â=0.0310891587 and b =0.0045282873. After estimating these values, the goodness of fit for 21 observations for LFRD based SRGM is assessed by QQ-plot technique where in the correlation between ordered data observations and its sample quantiles obtained through inversion of LFRD cumulative distribution function at i/n+1. The calculated correlation coefficient between the sample order statistics and population quantile is found to be 0.9761 indicating a very strong relation between the data and the model, suggesting that LFRD is a good fit to the data.



For the above data containing count of cumulative failures, let us start at an arbitrary choice of cumulative failures, say, let us note down the time by which 10 cumulative failures are experienced. In the present example it is.. "10 cumulative failures are observed within 10 weeks". From that time onwards using the data on time t_i,cumulative number of failures y_i, we get the estimate of mean value function with the help of MLEs of the parameters which are given in Table 5. For the sake of explanation let us take the specified costs C_1, C_2, C_3 as $C_1 = 25$, $C_2 = 200$, $C_3 = 7000$,the choice μ_y be kept at $\mu_y = 0.1$ (as considered by Pham (2000) [18]). These specifications would help us to get the values of expected total software cost as given by Equation (5.3). For various times and cumulative failures of the data set, our chosen time is "9th week onwards". Therefore from 9th week onwards in the data set at each time point we can calculate E(t). These are given in Table 6, which searches for a trend in E(t) from a rise to a fall and a rise after 9th week onwards say 9.5 etc. It shows that E(t) gives the desired trend of rise-fall-rise at 14.5. We therefore suggest to release the software after 14th week before 15th week based on LFRD. The same data based on Goel-Okumoto model suggest to release after 20th week as worked out in Pham(2000) [18].Based on half logistic model it is suggested to release after 17th week as worked out in Srinivas et al (2011) [21] This example also indicates that LFRD based NHPP suggests an earlier release than Goel-Okumoto and half logistic models at an optimal expected cost.

VI. SUMMARY & CONCLUSIONS

We have considered the well known linear failure rate distribution of the statistical science to develop a SRGM through NHPP. Its suitability and preferability over three reliability growth models are exemplified with the help of three live data sets . The trend in the time points of the release time data shows an increasing tendency upto some stage and then falling down and again increasing. The turning point from decresing to increasing is considered as optimal release time because beyond that the test time is again increasing. As a matter of check the software reliability at denoted release tome is calculated with the estimated parameters using equation (3.4). It is found to be for this data as 0.9887. This is a very satisfactory reliability level doubly indicating that at the release time the product has reliability is more than 98% strong.

ISSN: 2454-4248
92 – 101

VII. ACKNOWLEDGMENTS

The authors thank Prof.R.R.L.Kantam for his helpful suggestions, comments and encouragement, which helped in improving the final version of the paper.

REFERENCES

[1] Akaike.H. A new look at statistical model identification. IEEE Transactions on Automatic Control, 19:716-723, 1974.

[2] Anu Gupta. Kapur.P.K and Omphal Singh. On discrete software reliability growth model and categorization of faults. Operational Research Society of India(OPSEARCH), 42(4):340-354, Dec 2005.

[3] Bardhan.A.K. Kapur.P.K. and Shatnawi.O. Why software reliability growth modelling should define errors of different severity. Journal of Indian Statistical Association, 40(2):119-142, 2002.

[4] Goel.A.L. and Okumoto.K. A time dependent error detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability, 28(3):206-211, 1979.

[5] Hoang Pham. A generalized logistic software reliability growth model. Operational Research Society of India (OPSEARCH), 42(4):322-331, Dec 2005.

[6] Huang and Kuo.S.Y. Analysis of incorporating logistic testing effort function into software

[7] 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.

[8] Huang.C.Y. and Lyu.M.R. Optimal release time for software systems considering cost,testing effort and test efficiency. IEEE transactions on Reliability, 54, Dec 2005.

[9] 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.

[10] 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.

[11] Kuo.S.Y. Huang.C.Y. and Lyu.M.R. Optimal software release policy based on cost, reliability and testing efficiency. In Proceedings of 23rd IEEE Annual International, 1999.

[12] Kuo.S.Y. Huang.C.Y. and Lyu.M.R. Effort-index based software reliability growth models and performance assesment. In Proceedings of 24th IEEE Annual International Computer Software and Applications Conference, pages 454-459, 2000.

[13] Lan.Y. and Leemis.L. The logistic exponential survival distribution. Naval Research Logistics,55(3):252-264, 2007.

[14] Nordmann.L. Pham.H. and Zhang.X. A general imperfect software debugging model with s-shaped fault detection rate. IEEE Transaction on Reliability, (48):169-175, 1999.

[15] Ohba.M. Yamada.S. and Osaki.S. S-shaped reliability growth modeling for software error detection. IEEE Transactions on Reliability, 32(5):475-484, 1983.

[16] Ohtera.H. Yamada.S. and Narihisa.R. Software reliability growth models with testing effort.IEEE Transactions on Reliability, R-35:19-23, 1986.

[17] Peer.M.A. Quadri.S.M.K., Ahmad.N. and Kumar.M. Nonhomogeneous poisson process software reliability growth model with generalized exponential testing effort function. RAU Journal of Research, 16(1-2):159-163, 2006.

[18] Pham.H. Software Reliability. Springer-Verlag, 2000.

[19] Vara Prasad Rao.B., Gangadhar Rao.k., Srinivasa Rao.B., Inverse Rayleigh Software reliability growth model. International Journal of Computer Applications, 75(6),2013.

[20] 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.

[21] Srinivasa Rao.B., Vara Prasad Rao.B., Kantam.R.R.L., Software reliability growth model based on half logistic distribution. Journal of Testing and Evalaution, 39(6),:1152-1157, 2011.

[22] Tamura.Y. Yamada.S. and Kimra.M. A software reliability growth model for a distributed development environment. Electronic and Communications in Japan, 83(3):1446-1453, 2003.

[23] Wood.A.P. Predicting software reliability. IEEE Computer, 11:69-77, 1996.

[24] Yamada.S. and Inoue.S. Testing coverage dependent software reliability growth modeling. In-ternational Journal of Quality , Reliability and Safety Engineering, 11(4):303-312, 2004.

[25] 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.