International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 10, October – 2017, PP – 11-19

Software Quality Control with Linear Failure Rate Distribution Based on Nhpp - An Order Statistics Approach

B.Vara Prasad Rao,

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

K.Gangadhara Rao,

Professor Of Computer Science,
Department of Computer Science & Engineering,
Acharya Nagarjuna University, Guntur
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 characteristics. However, if the observed variations are with in the tolerable limits statistical process control (SPC) methods would pass them for acceptance. This technique 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 control 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. The new concept of the theory of order statistics is also made use of to trace out the off control signals in the context of every r^{th} observed failure rather than every single failure of the software.

Keywords: LFRD, SPC, NHPP, SRGM

1 Introduction

Control charts are based on the principal of monitoring quality variations in the product of a manufacturing process. Though the variations are not completely eliminated, the control chart assures a tolerable zone for the acceptability of the production process with variations. In the classical literature, various types of control charts have become popular both in normal and non-normal situations. The concept of applying control chart for monitoring the failure phenomenon is of recent origin. The failure data represented in terms of inter failure times of a product can be used to assess the quality of the product measured in terms of inter failure times. It is natural to believe that the more the inter failure time, the better the quality of the product. Similarly the less the inter failure time the poorer the quality of the product. However, inter failure time is a positive valued continuous random variable with an induced probability model for it. Hence, percentiles of the probability model with a specified coverage probability can be explored to design a control chart to monitor the failure mechanism there by assessing the quality of the product under consideration as tolerable, superior than tolerable and inferior than tolerable. 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 in statistical science to compute software reliability. The earliest works in this direction can be attributed to those of Yamada *et al*(1986) [20], Wood(1996) [18], Pham *et al*(1999) [9], Pham(2000) [10], Haung and Kuo(2002) [1], Pham and Zhang(2003) [11], Yamada *et al*(2003) [21], Yamada and Inoue(2004) [22], Huang(2005) [2], Pham(2005) [12], Quadri *et al*(2006) [14], Huang *et al*(2007) [3], Lan and Leemis(2007) [7]. Similar attempt relates to Kantam and Subbarao(2009) [4] - Pareto distribution, Srinivasa Rao *et al*(2011) [17] - Half logistic distribution, Prasad *et al*(2013) [13] - Inverse Rayleigh distribution. All these attempts are focussed on the mathematical model of the type

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

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

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

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 θ *F(t)*. Hence θ *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.

Process monitoring of reliability related characteristics has attracted some attention recently. Some related research in this direction is attributed to Xie *et al*(2002) [19], Liu *et al*(2006) [8], Ramchand *et al*(2011) [15], Satya Prasad *et al*(2011) [16] and Kim(2011) [6].

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 through order statistics approach and demonstrate how a graphical procedure called a statistical process control (SPC) 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) are presented in section 2. Instead of developing the control chart failure time that is noticed, sometimes it would be desirable to wait for a considerable number of failures and then develop a control chart for the failure time data after waiting. Incorporating this notion in the theory of order statistics, control charts for the mean value function can be developed after waiting for a fixed number of failures each time. This principle along with the resulting control limits and their applications to a live data are described in section 3. Comparison of the present study with other statistical models and the robustness of our investigation is discussed in section 4. Summary and Conclusions are given in section 5.

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. All these models are either constant failure rate (CFR) or absolutely increasing 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

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

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(θ ,x) as the mean value function as the SRGM for our present study is

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

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

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

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

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

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

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

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

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

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

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

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(2.10) 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 these values corresponding to θ =0(0.001)0.5 is listed in the Table 1.

The remaining values are available with the authors.

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
.00	0.522723	0.523139	0.523556	0.523971	0.524387	0.524801	0.525215	0.525629	0.526042	0.526454
.01	0.526866	0.527277	0.527688	0.528098	0.528508	0.528917	0.529326	0.529734	0.530142	0.530549
.02	0.530955	0.531361	0.531767	0.532172	0.532576	0.532980	0.533384	0.533787	0.534189	0.534591
.03	0.534992	0.535393	0.535793	0.536193	0.536592	0.536991	0.537389	0.537788	0.538184	0.538581
.04	0.538977	0.539373	0.539768	0.540163	0.540557	0.540951	0.541344	0.541737	0.542129	0.542521
.05	0.542912	0.543303	0.543693	0.544083	0.544472	0.544861	0.545249	0.545637	0.546024	0.546411
.06	0.546797	0.547183	0.547569	0.547953	0.548338	0.548722	0.549105	0.549488	0.549871	0.550253
.07	0.550634	0.551016	0.551396	0.551776	0.552156	0.552535	0.552914	0.553292	0.553670	0.554047
.08	0.554424	0.554801	0.555177	0.555552	0.555927	0.556302	0.556676	0.557050	0.557423	0.557796
.09	0.558168	0.558540	0.558911	0.559282	0.559653	0.560023	0.560392	0.560762	0.561130	0.561498
.10	0.561866	0.562234	0.562601	0.562967	0.563333	0.563699	0.564064	0.564429	0.564793	0.565157
.11	0.565520	0.565883	0.566246	0.566608	0.566969	0.567331	0.567692	0.568052	0.568412	0.568771
.12	0.569130	0.569489	0.569847	0.570205	0.570563	0.570920	0.571276	0.571632	0.571988	0.572343
.13	0.572698	0.573053	0.573407	0.573760	0.574113	0.574466	0.574818	0.575170	0.575522	0.575873
.14	0.576224	0.576574	0.576924	0.577273	0.577623	0.577971	0.578319	0.578667	0.579015	0.579362
.15	0.579708	0.580055	0.580400	0.580746	0.581091	0.581436	0.581780	0.582124	0.582467	0.582810
.16	0.583153	0.583495	0.583837	0.584178	0.584519	0.584860	0.585200	0.585540	0.585879	0.586219
.17	0.586557	0.586896	0.587234	0.587571	0.587908	0.588245	0.588581	0.588917	0.589253	0.589588
.18	0.589923	0.590258	0.590592	0.590925	0.591259	0.591592	0.591924	0.592256	0.592588	0.592920
.19	0.593251	0.593581	0.593912	0.594242	0.594571	0.594900	0.595229	0.595558	0.595886	0.596218
.20	0.596541	0.596868	0.597194	0.597520	0.597846	0.598172	0.598497	0.598822	0.599146	0.599470
.21	0.599794	0.600117	0.600440	0.600763	0.601085	0.601407	0.601728	0.602049	0.602370	0.602691
.22	0.603011	0.603330	0.603650	0.603969	0.604287	0.604606	0.604924	0.605241	0.605558	0.605875
.23	0.606192	0.606508	0.606824	0.607139	0.607455	0.607769	0.608084	0.608398	0.608712	0.609025
.24	0.609338	0.609651	0.609963	0.610275	0.610587	0.610898	0.611209	0.611520	0.611830	0.612140
.25	0.612450	0.612759	0.613068	0.613377	0.613685	0.613993	0.614301	0.614608	0.614915	0.615222
.26	0.615528	0.615834	0.616139	0.616445	0.616750	0.617054	0.617359	0.617662	0.617966	0.618269
.27	0.618572	0.618875	0.619177	0.619479	0.619781	0.620082	0.620383	0.620684	0.620984	0.621284
.28	0.621584	0.621884	0.622183	0.622481	0.622780	0.623078	0.623376	0.623673	0.623970	0.624267
.29	0.624564	0.624860	0.625156	0.625451	0.625746	0.626041	0.626336	0.626630	0.626924	0.627218
.30	0.627511	0.627804	0.628097	0.628389	0.628682	0.628973	0.629265	0.629556	0.629847	0.630137
.31	0.630428	0.630718	0.631007	0.631297	0.631586	0.631874	0.632163	0.632451	0.632739	0.633026
.32	0.633313	0.633600	0.633887	0.634173	0.634459	0.634745	0.635030	0.635315	0.635600	0.635884
.33	0.636168	0.636452	0.636736	0.637019	0.637302	0.637585	0.637867	0.638149	0.638431	0.638713
.34	0.638994	0.639275	0.639555	0.639836	0.640116	0.640395	0.640675	0.640954	0.641233	0.641511
.35	0.641790	0.642068	0.642345	0.642623	0.642900	0.643177	0.643453	0.643730	0.644006	0.644281
.36	0.644557	0.644832	0.645107	0.64538	0.645655	0.645929	0.646203	0.646476	0.646750	0.647022
.37	0.647295	0.647567	0.647839	0.64811	0.648382	0.648654	0.648924	0.649195	0.649465	0.649735
.38	0.650005	0.650275	0.650544	0.65081	0.651081	0.651350	0.651618	0.651886	0.652153	0.652421
.39	0.652688	0.652954	0.653221	0.65348	0.653753	0.654018	0.654284	0.654549	0.654814	0.655078
.40	0.655343	0.655607	0.655870	0.65613	0.656397	0.656660	0.656923	0.657185	0.657447	0.657709
.41	0.657971	0.658232	0.658493	0.65875	0.659014	0.659275	0.659535	0.659794	0.660054	0.660313
.42	0.660572	0.660831	0.661089	0.66134	0.661605	0.661863	0.662120	0.662378	0.662634	0.662891
.43	0.663147	0.663403	0.663659	0.66391	0.664170	0.664425	0.664680	0.664935	0.665189	0.665443
.44	0.665697	0.665950	0.666204	0.66645	0.666709	0.666962	0.667214	0.667466	0.667718	0.667969
.45	0.668221	0.668472	0.668722	0.66897	0.669223	0.669473	0.669723	0.669972	0.670222	0.670471
.46	0.670719	0.670968	0.671216	0.67146	0.671712	0.671959	0.672207	0.672454	0.672700	0.672947
.47	0.673193	0.673439	0.673685	0.67393	0.674176	0.674421	0.674666	0.674910	0.675155	0.675399
.48	0.675643	0.675886	0.676130	0.67637	0.676616	0.676858	0.677101	0.677343	0.677585	0.677826
.49	0.678068	0.678309	0.678550	0.67879	0.679031	0.679272	0.679512	0.679751	0.679991	0.680230
.50	0.680469	0.680708	0.680947	0.68118	0.681423	0.681661	0.681899	0.682136	0.682373	0.682610

3 Monitoring the failures based on mean value function using order statistics with control chart

Let $x_1, x_2, .., x_n$ be a random sample of size n representing n inter failure times of a product governed by the probability model of a continuous random variable X. Let F(x) be the cumulative distribution function of X. These inter failure times can be used for assessing the failure phenomenon with respect to two limits of reference called control limits with a pre specified coverage probability. Thus the time control chart plotted for inter failure times would indicate alarms, advantages and stable failure process. If r is a natural number ($<$ n), the summations $\sum_{i=1}^{r} X_i$, $\sum_{i=r+1}^{2r} X_i$, $\sum_{i=2r+1}^{3r} X_i$ etc represent the lapse of time consecutively between every r^{th} failure. A control chart for times between every r^{th} failure would throw light on the out of control signals than that of inter failure times. Xie et al(2002) [19] named such a control chart as t_r -control chart and developed control limits using the sampling distribution of $\sum_{i=1}^{r} X_i$. They have taken the example of exponential distribution and used the theory that the sum of exponential variates is a gamma variate to get the percentiles of t_r -control chart with the help of cumulative summations. If the inter failure times are not exponentials, the control limits of t_r -chart of Xie et al(2002) [19] can not be used. Overcoming this drawback we suggest the following alternative approach to get control limits of t_r -chart for any distribution. If ($X_1, X_2, .., X_r$); ($X_{r+1}, X_{r+2}, .., X_{2r}$); ($X_{2r+1}, X_{2r+2}, .., X_{3r}$); etc are regarded as independent samples of size r each, i.i.d random variables having F(x) as their common model. $Y_1 = X_1$, $Y_2 = \sum_{i=1}^{2} X_i$, $Y_3 = \sum_{i=1}^{3} X_i$, ..., $Y_r = \sum_{i=1}^{r} X_i$ becomes an ordered sample of size r representing the time to first failure, time to second failure, time to third failure,....., time to r^{th} failure respectively. Thus, the t_r -chart is the control chart with Y_r as the points on it representing the time to every r^{th} failure. Therefore, when r is fixed, the percentiles of highest order statistics in a sample of size r would serve the purpose of control limits for the t_r -chart.

Let F(x) be the cumulative distribution function of a continuous positive valued random variable. If the random variable is taken as representing inter failure time of a device, a control chart of such data with order statistics would be based on 0.9973 probability limits of the times between failure random variable say *t*. These limits and the central line are respectively the solutions of the following equations taking equi-tailed probabilities.

$$[F(t)]^r = 0.00135 \tag{13}$$

$$[F(t)]^r = 0.5 \tag{14}$$

$$[F(t)]^r = 0.99865 \tag{15}$$

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

$$t_L = F^{-1}(0.00135^{\frac{1}{r}}) \tag{16}$$

$$t_C = F^{-1}(0.5^{\frac{1}{r}}) \tag{17}$$

$$t_U = F^{-1}(0.99865^{\frac{1}{r}}) \tag{18}$$

The NHPP of LFRD with F(θ ,x) as the mean value function as the SRGM for our present study is

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

The above model is illustrated for the example of 60 failure times considered by Xie *et al*(2002) [19]. For a ready reference the data is produced in table 2.

Table 2: Failure time data of the components

Failure number	Time	Failure number	Time	Failure number	Time	Failure number	Time
1	1065.55	16	2932.96	31	35.85	46	239.66
2	535.8	17	987.67	32	362.8	47	93.78
3	540.53	18	1816.18	33	357.85	48	680.45
4	716.2	19	117.21	34	334.48	49	4.83
5	2525.43	20	190.65	35	80.13	50	102.91
6	1264.18	21	943.99	36	1939.0	51	479.05
7	479.44	22	1084.48	37	77.88	52	156.67
8	1783.22	23	2306.54	38	4.03	53	1286.24
9	473.67	24	6.56	39	98.67	54	443.97
10	2265.42	25	3111.51	40	17.19	55	360.03
11	2191.75	26	283.86	41	289.79	56	414.66
12	1097.26	27	659.39	42	63.99	57	128.9
13	597.59	28	683.48	43	2.46	58	36.1
14	971.16	29	36.14	44	697.68	59	197.31
15	3157.29	30	754.16	45	1167.33	60	418.12

Table 3: Accumulation failure time for every three failures

Observation	Accumulation of 3 failures	Observation	Accumulation of 3 failures
1	2141.88	11	756.5
2	4505.81	12	2353.61
3	2736.33	13	180.58
4	5554.43	14	370.97
5	4726.04	15	1867.47
6	5736.81	16	1013.89
7	1251.85	17	586.79
8	3397.58	18	1886.88
9	4054.76	19	903.59
10	1473.78	20	651.53

The parametric estimates and the time control limits based on the mean value function of LFRD corresponding to inter failure time together with three parallel lines to the horizontal axis at $m(t_L)$, $m(t_C)$ and $m(t_U)$ for the data in table 3 are given below as table 4.

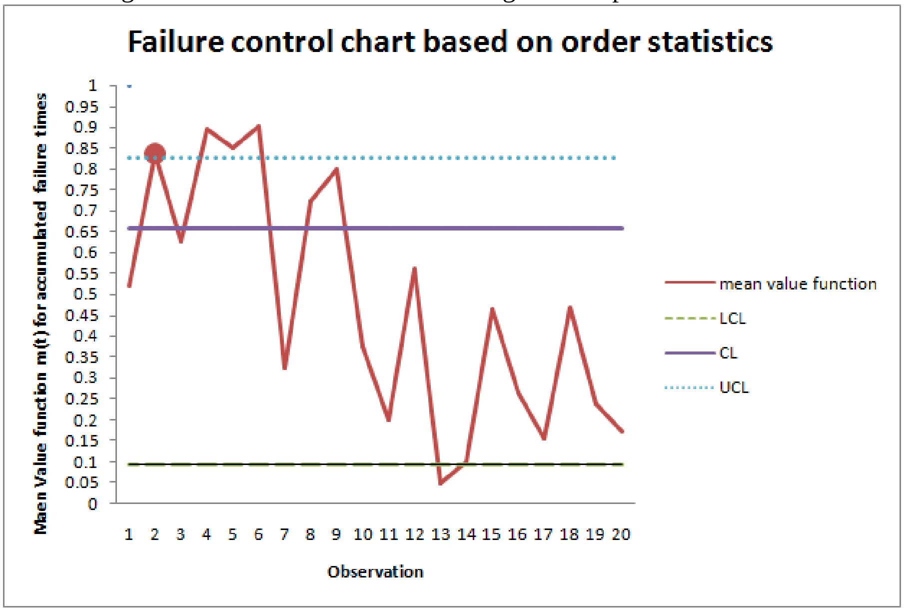
Table 4: Parameter estimates of LFRD for accumulated failure times and their control limits

Linear Failure Rate model					
$\hat{a}$	$\hat{b}$	$\hat{\theta}$	$m(t_L)$	$m(t_C)$	$m(t_U)$
$.2782133 \times 10^{-3}$	0.859×10^{-6}	0.949	0.091511	0.657184	0.827625

Table 5: Mean Value Function for accumulated failure times

Observation	Accumulation of 3 failures	m(t)	Observation	Accumulation of 3 failures	m(t)
1	2141.88	0.519565	11	756.5	0.198785
2	4505.81	0.835722	12	2353.61	0.560349
3	2736.33	0.627648	13	180.58	0.047763
4	5554.43	0.895214	14	370.97	0.098105
5	4726.04	0.851363	15	1867.47	0.463068
6	5736.81	0.902203	16	1013.89	0.264164
7	1251.85	0.322694	17	586.79	0.154775
8	3397.58	0.724391	18	1886.88	0.467196
9	4054.76	0.797410	19	903.59	0.236378
10	1473.78	0.375307	20	651.53	0.171633

These 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 given below the first out of control situation is noticed at the 2^{nd} failure falling above UCL. It results in an earlier alarm signal and hence out-of-control for the software failure. The assignable cause for this is to be investigated and promoted.



4 Comparative Study

The above model is compared with Exponential distribution. Its CDF is given by

$$F(t) = 1 - e^{-bt} \tag{20}$$

The ML estimates of the parameters in exponential model and are given by.

$$\sum_{i=1}^{n} \frac{t_i e^{-bt_i} - t_{i-1} e^{-bt_{i-1}}}{e^{-bt_{i-1}} - e^{-bt_i}} (y_i - y_{i-1}) = \frac{y_n t_n e^{-bt_n}}{1 - e^{-bt_n}} \tag{21}$$

$$a = \frac{y_n}{1 - e^{-bt_n}} \tag{22}$$

The mean value function of Exponential distribution based on NHPP is given by

$$m(t) = a \times (1 - e^{-bt_n}) \tag{23}$$

The parametric estimates and the time control limits based on the mean value function of Exponential distribution corresponding to accumulated inter failure times together with three parallel lines to the horizontal axis at $m(t_L)$, $m(t_C)$ and $m(t_U)$ for the data in table 3 are given below as table 6.

Table 6: Parameter estimates of Exponential model for accumulated failure times and their control limits

Exponential model				
$\hat{a}$	$\hat{b}$	$m(t_L)$	$m(t_C)$	$m(t_U)$
20	0.026	2.210424	15.874009	19.990995

Table 7: Exponential model Mean Value Function for accumulated failure times

Observation	Accumulation of 3 failures	m(t)	Observation	Accumulation of 3 failures	m(t)
1	2141.88	8.5402	11	756.5	3.5711
2	4505.81	13.8020	12	2353.61	9.1540
3	2736.33	10.1813	13	180.58	0.9173
4	5554.43	15.2811	14	370.97	1.8389
5	4726.04	14.1470	15	1867.47	7.6927
6	5736.81	15.4996	16	1013.89	4.6346
7	1251.85	5.5562	17	586.79	2.8299
8	3397.58	11.7323	18	1886.88	7.7547
9	4054.76	13.0308	19	903.59	4.1875
10	1473.78	6.3662	20	651.53	3.1165

Clearly we can see that all the values of m(t) in the above table lie within the control limits $m(t_L)$ and $m(t_U)$ indicating a stable failure process. The same data when applied to the LFRD model has indicated an earlier alarm signal and hence out-of-control for the software failure. The assignable cause for this is to be investigated and promoted. But whereas in the case of Exponential model because of stable failure process there is no indication to detect the earlier failure and one has to wait till the last failure occurs.

5 Summary & Conclusions

There are many control charts which use statistical techniques. It is important to use the best chart for the given data, situation and need. There are advanced charts that provide more effective statistical analysis. In this paper, the accumulated failure times are plotted through the estimated mean value function against the failure serial order. Hence, we conclude that our method of estimation and the control chart are giving a recommendation for their use in finding out preferable control process or desirable out of control signal. Hence our proposed mean value chart detects out of control situation at an earlier instant than the situation in time control chart. The early detection of software failure will improve the software reliability. On the other hand, mean value function based on the accumulated failure times has exceeded the UCL, these are probably reasons that have led to significant improvement. Further evaluation of control limits in our approach is simpler involving inversion of r^{th} power of the distribution function of the parent population of statistical model under consideration.

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.

Compliance with Ethical Standards

The authors declare that there is no conflict of interest. The research is not supported by any source of funding.

References

[1]. 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.

[2]. 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.

[3]. 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

[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]. Liu.J.Y.,Xie.M., Goh.T.N., and Sharma.P.R. A comparative study of exponential time between events charts. *Quality Technology and Quantitative Management*, 3(3):347–359,2006.

[9]. 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.

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

[11]. 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.

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

[13]. 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.

[14]. Quadri.S.M.K.,Peer.M.A., 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.

[15]. Ram Chand.K., Rao.H., Satya Prasad.K. and Kantam.R.R.L. Assesing software reliability using SPC -An order statistics approach. *International Journal of Computer Science Engineering and Applications*, 1(4):140–150, 2011.

[16]. Satya Prasad.K., Rao.H. and Kantam.R.R.L. Software reliability with SPC *International Journal of Computer Science and Emerging Technologies*, 2(2):233–237, 2011.

[17]. 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 Evalaution*, 39(6),:1152–1157, 2011.

[18]. Wood.A.P. Predicting software reliability. *IEEE Computer*, 11:69–77, 1996.

[19]. Xie.M., Goh.T.N., and Ranjan.P. Some effective control chart procedurs for reliability monitoring. *Reliability Engineering and System Safety*, 77:143–150, 2002.

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

[21]. 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.

[22]. 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.

