Software Quality Assurance Based on Order Statistics

[1]Dr. R. Satya Prasad, [2]D. Haritha
[1]Dept. of Computer Science, Acharya Nagarjuna University, Guntur, AP, India
[2]Dept. of ECM, KL University, Vaddeswaram, AP, India

Abstract
Statistical Process Control is an effective technique to optimize the quality and productivity of large scale software firms. Control charts are extensively used to monitor the process to note the variations in the software process that are results of unpredictable causes that behave in an unintended way results in fixing the bug in a flash by the team lead. For an effective monitoring of failure process the time between every r^{th} failure (r is a natural number >=2) instead of inter-failure times is considered for developing a variable control chart called Time Control Charts. This paper projects a controlling framework based on order statistics of the cumulative quantity between observations of time domain failure data using mean value function of Logarithmic Poisson Execution Time Model (LPETM), which is a Non Homogenous Poisson Process (NHPP). The two unknown parameters of the Logarithmic model are arrived at, using The Maximum Likelihood Estimation (MLE).

Keywords
Software Process Control, Software Quality, Order Statistics, Logarithmic Poisson Execution Time Model, MLE, Control Charts

I. Introduction
Nowadays computers are used in diverse areas and applications. As the computer applications become essential the corresponding and in other words critical software applications increase in size and complexity. There is a growing need for computer software reliability. A software failure is defined as the occurrence of an unwanted output as a result of input that is received with respect to the specification. [IEEE Std. 610.12,1990]. Process monitoring has two major advantages compared to the detailed inspection of completed software units. First, errors may be detected earlier or prevented altogether. With process monitoring, error-prone zones like inadequate standards, insufficient training, incompatible hardware may be detected before defective inputs are made. [3]. A number of statistical models [1-2] has been developed to quantify the reliability of a software system during testing and operational phases based on its failure track record. This paper proposes a statistical process control in monitoring the quality of a software system being tested. Control charts are effective tools in Statistical Process Control (SPC) for monitoring the stability of a process over time. Currently numbers of manufacturing companies are implementing SPC in various applications. The practical applications of control charts now extend far beyond manufacturing in to biology, genetics, medicine, finance and other areas [4]. Let $X_1, X_2, \ldots, X_n$ 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. According to Xie et al (2002) we have taken the coverage probability as 0.9973. The control limits are taken as the equitailed percentiles namely 0.00135 percentile and 0.99865 percentile respectively denoted as TL, TU. The TL and TU represent the two horizontal parallel lines of time control chart. [10].A point above TL shows a larger gap of inter failure times and hence advantage for the production process. A point below TL indicates too many frequent failures that alarms a negative signal. The failure mechanism is tolerable when the points lie between TL and TU. 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 [10]. A control chart for times between every rth failure spotlights on the out of control signals than that of inter failure times. Xie et al. (2002) named such a control chart as t_r-control chart and developed control limits using the sampling distribution of . If the inter failure times are not exponentials, the control limits of tr-chart of Xie et al (2002) cannot be used. To Overcome this disability we would like to recommend the following alternative approach to get control limits of tr-chart for any distribution. If $(X_1, X_2, \ldots, X_r)$; $(X_{r+1}, X_{r+2}, \ldots, X_{2r})$; $(X_{2r+1}, X_{2r+2}, \ldots, X_{3r})$; etc are regarded as independent samples of size r each then $Y_1=X_1$, $Y_2=\sum_{i=1}^{2} X_i$, $Y_3=\sum_{i=1}^{3} X_i$, …, $Y_r=\sum_{i=1}^{r} X_i$ becomes an ordered sample of size r representing the time to first failure, time to second failure,…, time to r^{th} failure respectively. Y_r is the highest order statistics in an ordered sample $Y_1<Y_2<\ldots<Y_r$. 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. We know that $[F(x)]^r$ is the cumulative distribution function of rth order statistic in a sample of size r for the model F(x). Hence, the percentiles of t_r-chart with 0.9973 coverage probability would be the solutions of $[F(x)]^r = 0.99865$ and $[F(x)]^r = 0.00135$. The central line of the tr-chart would be the solution of $[F(x)]^r = 0.5$. [10].

II. Review of LPETM
Logarithmic Poisson Execution Time Model (LPETM) proposed by Musa and Okumoto[1], is a NHPP based Infinite failure category model. The striking feature of NHPP models is to find out an appropriate mean value function to denote the expected number of failures occurring up to a certain point in time. A software reliability model, in general is a random process $\{m(t), t \geq 0\}$ representing the number of failures of a software system during occurring execution time t. A model may be characterized by specifying the distribution of m(t), the mean value function,

$$\mu(t)=E[m(t)]. \quad (1)$$

and the failure intensity function can then be derived as

$$\lambda(t) = \frac{d\mu(t)}{dt} \quad (2)$$

LPETM may be defined in terms of Non Homogeneous poisson process. The mean value function m(t), and the failure intensity λ(t) can be deduced based on the assumption that the failure intensity decreases exponentially as failures encountered [4]. The mean value function for the given model is,

$$m(t) = a.\log(1+bt) \quad (3)$$

The failure intensity function

$$\lambda(t) = \frac{ab}{1+bt} \quad (4)$$

where a is the initial expected total number of faults and b is the failure detection rate.

III. MLE Approach

Least-Squares Estimation (LSE) and Maximum Likelihood Estimation (MLE) are the two well known parameter estimation methods exist in the literature [6]. In this paper MLE approach is used to estimate the model parameters that maximize the probability of the failure data sets. Determining whether a process is in control or out of control requires estimation for the variables and charting over inter failure times.

This paper makes use of failure occurrence times, s_j(observed failures) where $0 \le s_1 \le s_2 \ldots \le s_n$.[pham].

Therefore the time between failures $t_i = s_i - (s_{i-1})$ for i=1,2,…,n.

The Log likelihood function for the time domain data is,

$$LLF = \sum_{i=1}^{n} \log[\lambda(s_i)^r] - m(s_n)^r \qquad (5)$$

Where $\lambda(t) = \frac{\partial}{\partial t} m(t)$

Substitute (3) in (5) we get,

LLF = Log L

$$\sum_{i=1}^{n} \log[\frac{ab}{1+bs_i}]^r - a.\log(1+bs_n)^r \qquad (6)$$

The unknown parameters a & b of the given LPETM can be obtained using order statistics approach as follows:

To get m(t) value for r^{th} Order Statistics, take m(t) to the power 'r'

$$m(t) = a.\log(1+bt)^r$$

Firstly, take the partial derivative of equn(6) w.r.t 'a' and equate it to zero we get,

$$\sum_{i=1}^{n} \frac{\frac{\partial}{\partial a}[\frac{ab}{1+bs_i}]^r}{[\frac{ab}{1+bs_i}]^r} - \frac{\partial}{\partial a}(a.\log(1+bs_n))^r = 0$$

$$a^r = \frac{n}{\log(1+bs_n)^r} \qquad (7)$$

Upon Differentiating Eqn.(6) w.r.t 'b' and equate it to zero, we get,

$$g(b) = \sum_{i=1}^{n} \frac{1}{b} - \frac{r*s_n}{\log(1+bs_n)^r(1+bs_n)} - \frac{s_i}{1+bs_i} \qquad (8)$$

Differentiating g(b) in Eqn.(8) we get ,

$$g/(b) = \frac{s_n^2*r}{(1+bs_n)^2\log(1+bs_n)^r} + \frac{s_n^2*r^2}{[(1+bs_n)\log(1+bs_n)^r]^2} + \frac{s_i^2}{(1+bs_i)^2} - \frac{1}{b^2} \qquad (9)$$

Iterative Newton-Raphson method is used to Solve the equations (7),(8),(9) in order to get the approximated a & b values for the given sets of failure data.

IV. Illustrating Time Control Charts

Software process control requires periodic monitoring of the progress being made towards accomplishing the quality. Three parameters namely CL (Center Line), Lower Control Limit (LCL) and Upper Control Limit (UCL) are used as reference points to detect the reliability of a software system.CL is the centre line that represents mean value. By comparing our failure data with them we can draw a conclusion that whether a process is predictable or not. [11]

Control limits are important decision aids as if all points fall between the control limits, process is in control otherwise it is out-of-control. Assuming the false alarm probability to be 0.27% , and considering the Table 1 data the control limits can be determined.

Table 1: Failure Data Set

Failure .No	Time Between Failures (hrs)	Failure .No	Time Between Failures (hrs)	Failure .No	Time Between Failures (hrs)	Failure .No	Time Between Failures (hrs)	Failure .No	Time Between Failures (hrs)
1	479	18	277	35	1620	52	181	69	1487
2	266	19	596	36	1601	53	1485	70	4322
3	277	20	757	37	298	54	757	71	1418
4	554	21	437	38	874	55	3154	72	1023
5	1034	22	2230	39	618	56	2115	73	5490
6	249	23	437	40	2640	57	884	74	1520
7	693	24	340	41	5	58	2037	75	3281
8	597	25	405	42	149	59	1481	76	2716
9	117	26	535	43	1034	60	559	77	2175
10	170	27	277	44	2441	61	490	78	3505
11	117	28	363	45	460	62	593	79	725
12	1274	29	522	46	565	63	1769	80	1963
13	469	30	613	47	1119	64	85	81	3979
14	1174	31	277	48	437	65	2836	82	1090
15	693	32	1300	49	927	66	213	83	245
16	1908	33	821	50	4462	67	1866	84	1194
17	135	34	213	51	714	68	490	85	994

$$T_L = \frac{1}{b}\left(e^{0.00135} - 1\right)$$

$$T_C = \frac{1}{b}\left(e^{0.5} - 1\right)$$

$$T_U = \frac{1}{b}\left(e^{0.99865} - 1\right)$$

The control limits T_L,T_C ,T_U and the modal parameters a and b are used in determining $m(T_L)$, $m(T_C)$, $m(T_U)$,which are the deciding factors of the software assessing process.

Table 2: Parameter Estimates and Their Control Limits of 4th and 5th Order

order	a	b	$m(t_u)$	$m(t_l)$	$m(t_c)$
4	1.836407	0.000023	0.9446	0.1811	0.7946
5	1.567825	0.000021	0.9257	0.2469	0.8061

Table 3: Successive Differences of 4th Order Mean Values of Table 1

F.No	Cumm. Failure of 4-order	m(t)	SD of m(t)	F. No	Cumm. Failure of 4-order	m(t)	SD of m(t)	F. No	Cumm. Failure of 4-order	m(t)	SD of m(t)
1	1576	0.4105	0.1087	8	19572	0.7378	0.0304	15	53223	0.8935	0.0082
2	4149	0.5192	0.0435	9	23827	0.7683	0.0266	16	56160	0.9018	0.0140
3	5827	0.5627	0.0756	10	28257	0.7949	0.0189	17	61565	0.9158	0.0190
4	10071	0.6384	0.0235	11	31886	0.8138	0.0121	18	69815	0.9348	0.0254
5	11836	0.6619	0.0380	12	34467	0.8260	0.0261	19	82822	0.9603	0.0141
6	15280	0.7000	0.0149	13	40751	0.8521	0.0262	20	91190	0.9744	0.0100
7	16860	0.7150	0.0228	14	48262	0.8784	0.0150	21	97698	0.9844	---

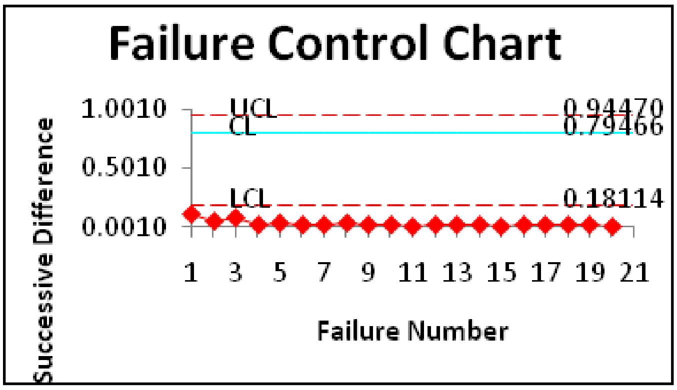

Fig 1: Control Chart for Table 3.

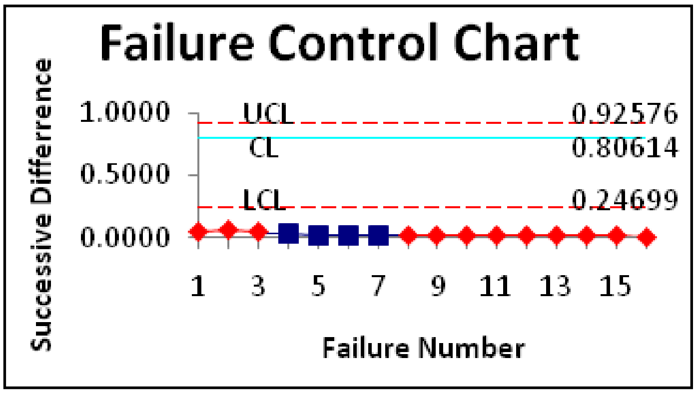

Fig 2: Control Chart for Table 4.

Table 4: Successive Differences of 5th Order Mean Values of Table 1

F.No.	Cumm. Failure of 5-order	m(t)	SD of m(t)	F.No.	Cumm. Failure of 5-order	m(t)	SD Of m(t)	F.No.	Cumm. Failure of 5-order	m(t)	SD of m(t)
1	2610	0.5153	0.0556	7	22226	0.7643	0.0305	13	58996	0.8869	0.0160
2	4436	0.5709	0.0695	8	28257	0.7948	0.0171	14	67374	0.9029	0.0205
3	8163	0.6404	0.0448	9	32346	0.8119	0.0263	15	80106	0.9235	0.0150
4	11836	0.6853	0.0349	10	39856	0.8383	0.0183	16	91190	0.9386	0.0091
5	15685	0.7202	0.0172	11	46147	0.8566	0.0176	17	98692	0.9476	----
6	17995	0.7375	0.0267	12	53223	0.8743	0.0126	-------	-----	-----	-----

A point that falls below the $m(T_L)$ is indicating an Alarm signal and above the $m(T_L)$ is advantageous. The software is stable When the points lie in between the control limits $m(T_L)$ and $m(T_U)$.

V. Conclusion

The results of our study are shown in the control charts in fig. 1., and fig. 2, for the given failure data set using order 4 and order 5 respectively. Our proposed method using order statistics approaches have successfully identified the failures at 1st point itself.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

Hence, we have succeeded in proving our approach as worthy even when the inter failure times are exponential. Three control limits UCL, CL and LCL are exhibited over the failure control charting. It is worthy to point that the majority of failures are within the allowable limits of lower level of control and therefore, the system assures software quality.

References

[1] J. D. Musa, K. Okumoto,"A logarithmic Poisson execution time model for software reliability measurement", in Proc. 7th Int. Conf. Software Eng., 1984, pp. 230-238.

[2] Kazuhira Okumoto,"A Statistical Method for Software Quality Control", in IEEE Transactions on Software Engg., Vol. SE-11, No. 12, Dec. 1985.

[3] M.Xie, T.N. Goh, P.Ranjan,"Some Effective control chart procedures for reliability monitoring", Elsevier, Reliability Engineering & System Safety 77, 2002, pp. 143-150.

[4] N. Boffoli, G. Bruno, D. Caivano, G. Mastelloni, "Statistical Process Control for oftware: A Systematic Approach", ESEM'08, October 9–10, 2008, Kaiserslautern, Germany, pp. 327-329, Copyright 2008 ACM.

[5] Shewhart, Walter A.,"The Economic Control of Quality of Manufactured Product", D. Van Nostrand Company, New York, 1931, reprinted by ASQC Quality Press, Milwaukee, Wisconsin, 1980.

[6] In Jae Myung,"Tutorial on maximum likelihood estimation", Journal of Mathematical Psychology 47, 2003, pp. 90–100.

[7] A text book on,"Software Reliability", By Hoang Pham.

[8] Qin Zhou, Yunzhao Luo, Zhaojun Wang,"A control chart based on likelihood ratio test for detecting patterned mean and variance shifts", Computational Statistics and Data Analysis 54, 2010, pp. 1634-1645.

[9] Sunantha Teyarachakul, Suresh Chand, Jen Tang, "Estimating the limits for statistical process control charts:A direct method improving upon the bootstrap", European Journal of Operational Research 178, 2007, pp. 472–481.

[10] R. R. L. Kantam, M. Ch. Priya,"Time Control Charts Using Order Statistics", Interstat.statjournals.net, 2011.

[11] D. Haritha, R. Satya Prasad,"A Statistical Process Control to Monitor the Software Quality", IJCA, Vol. 62, No. 11, January 2013, pp. 39-43.

Dr. R. Satya Prasad received Ph.D degree in Computer Science in the faculty of Engineering in 2007 from ANU. He received gold medal from ANU for his outstanding performance in Masters Degree. He is currently working as Associate Professor in the department of Computer science , ANU. His current research is focused on Software Engineering.

Smt. D. Haritha Received her B.Tech Degree in Computer Science & Engineering from JNTUH and M.Tech in CSE from ANU, and pursuing Ph.D from ANU. She is working as an Assistant Professor in the department of ECM, KL University. Her research interests include Software Engineering, Computer Networks, Embedded Systems.