A Statistical Process Control to Monitor the Software Quality

D. Haritha
Asst. Professor
Department of ECM
KL University, Vaddeswaram

R. Satya Prasad
Associate Professor
Department of Computer Science
Acharya Nagarjuna University, Guntur.

ABSTRACT
Software is the only man made Omni Present system contributing immensely providing complex and critical services to mankind. It's increasing popularity and usefulness enforces us to measure the software quality. In this paper Time domain failure data is applied to Statistical Process Control(SPC) method to monitor the quality of a software system. We propose a Statistical Control Method over the cumulative quantity between observations of time domain failure data using mean value function of an Non Homogeneous Poisson Process(NHPP) based Logarithmic Poisson Execution Time Model(LPETM).Maximum Likelihood estimation(MLE) is used to estimate the unknown parameters of proposed model.The SPC method employs LPETM to construct the control limits. Two failure data sets are used.

General Terms
Software quality , Software Reliability, Statistical Quality Control,MLE, NHPP.

Keywords
Software Process Control, Logarithmic Poisson Execution Time Model, Control Charts.

1. INTRODUCTION
Software reliability is defined as the probability of failure-free operation of a computer program in a specified environment for a specified time[1]. A failure is a departure of program operation from program requirements. A software product is viewed in terms of its quality, cost, and schedule. Since reliability directly affects the cost, schedule and performance of a software product, software reliability studies are very important. A large number of analytical models have been proposed and studied over the decades for assessing the quality of a software system. The reliability growth model represents the reliability or failure rate of a system as a function of time or the number of test cases.[pham]. Musa and Okumoto [1] proposed a new execution time model which predicts the reliability of a software process as well or better than other existing models, and is simpler than any of the models that approach in predictive validity. In this paper Time domain approach is used in analyzing reliability that has much higher accuracy in estimating the modal parameters than the interval domain approach.

Statistical Process Control (SPC), has been a valuable tool for decades in the manufacturing industry that helped the management in predicting the quality of the yield. Recently, SPC became a suitable instrument in the software industry too. It helps the team leads to monitor, predict, control and asses the quality of a software system being tested. It uses several "control charts" together with their indicators to establish operational limits for acceptable process variation. By using few data points, it is able to dynamically determine an upper and lower control limits of acceptable process performance variability[4].The main aim of monitoring task is to improve the software process quality by finding the failure causes, eliminate them if they are deleterious otherwise add them as part of your process.

Statistical process control is one of the best approach that is used to determine whether a process is in control or out of control. For a process to be in control, variation in the quality characteristics of the end-item must be predictable. Shewhart [5] substantiates that the variation in a quality characteristic arises because of two types of causes . Common (chance) causes which are always part of the process and Special (assignable) causes, that arise due to special circumstances and are not always part of the process.

Statistical control charts are widely used for failure process monitoring. In this paper we have opted SPC to monitor the reliability of a software system being tested. In doing this MLE approach is used to estimate the model parameters. Failure intensity can be predicted with control limits over the inter failure times using control charting.

2. DESCRIPTION OF LOGARITHMIC POISSON MODEL
Logarithmic Poisson Execution Time Model(LPETM) proposed by Musa and Okumoto[1], is a NHPP based Infinite failure category model. The prime aspect of NHPP models is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain point in time. A software reliability model, in general can be described as a random process $\{m(t),t\geq 0\}$ representing the number of failures of a software system experienced by execution time t. A model may be characterized by specifying the distribution of m(t) , the mean value function,

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

The failure intensity function can then be derived as

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

LPETM uses poisson distribution for the m(t) process. The mean value function and failure intensity are derived based on the assumption that the failure intensity decreases exponentially as failures are experienced.[4].

The mean value function for the proposed 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.

This proposed model is highly accurate and easy when compared with other models that exist in the literature.

3. PARAMETER ESTIMATION

Deliberating whether a process is in control or out of control requires an estimation for the variables and charting over inter failure times. There are two methods of parameter estimation available in the literature. They are least-squares estimation (LSE) and maximum likelihood estimation (MLE) [6]. Maximum-likelihood estimation (MLE) is a well known method of estimating the parameters of the model.

In this paper we have chosen the data to be occurrence times of the failures, sj(observed failures) where $0 \le s1 \le s2 \ldots \le sn$.[pham].

Therefore the time between failures $t_i = s_i - (si-1)$ for $i=1,2,\ldots,n$.

The Loglikelihood function for the time domain data is,

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

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

Substitute (3) in (5) we get,

$$LLF = \text{Log } L = \sum_{i=1}^{n} \log[\frac{ab}{1+bs_i}] - a.\log(1+bs_n) \quad (6)$$

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

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

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

$$\therefore \quad a = \frac{n}{\log(1+bs_n)} \quad (7)$$

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

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

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

$$g'(b) = \sum_{i=1}^{n} \frac{s_n \log(1+bs_n).s_n + (1+bs_n)\frac{s_n}{\log(1+bs_n)}}{[\log(1+bs_n)(1+bs_n)]^2} + \frac{{s_i}^2}{(1+bs_i)^2} - \frac{1}{b^2} \quad (9)$$

By applying Newton-Raphson method over Eqns.(7),(8),(9) we get approximated values of a & b for the given failure data sets.

4. MONITORING FAILURE INTENSITY

Software process control requires periodic monitoring of the progress being made towards attaining the quality. Three parameters viz. ,namely CL(Center Line), Lower Control Limit(LCL) and Upper Control Limit(UCL). These are used as reference points to assess 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.

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 are,

$T_L = \frac{1}{b}(e^{0.00135} - 1) = 5.339572$

$T_C = \frac{1}{b}(e^{0.5} - 1) = 2564.116$

$T_U = \frac{1}{b}(e^{0.99865} - 1) = 6777.133$

Failure data shown in Table-1 . We found out the values for a & b solving eqns. 7,8 and 9 .

a=175.1249

b=0.000253

These control limits and modal parameters are used in finding out $m(T_L)$, $m(T_C)$, $m(T_U)$, that helps in deciding state of the software process.

$m(T_U) = 75.9531$
$m(T_C) = 38.0279$
$m(T_L) = 0.102675$

Table 1 : Data set I

Failure Number	Time b/w failures	Failure Number	Time b/w failures
1	30.02	16	15.53
2	1.44	17	25.72
3	22.47	18	2.79
4	1.36	19	1.92
5	3.43	20	4.13
6	13.2	21	70.47
7	5.15	22	17.07
8	3.83	23	3.99
9	21	24	176.06
10	12.97	25	81.07
11	0.47	26	2.27
12	6.23	27	15.63
13	3.39	28	120.78
14	9.11	29	30.81
15	2.18	30	34.19

Table 2 Successive Differences of m(t) for Data set I.

Failure No.	m(t)	Successive Difference	Failure No.	m(t)	Successive Difference	Failure No.	m(t)	Successive Difference
1	0.575466	0.027494814	11	2.18762	0.11639029	21	4.787665	0.307800042
2	0.602961	0.427750982	12	2.30401	0.063258009	22	5.095465	0.071766972
3	1.030712	0.025812642	13	2.367268	0.169734021	23	5.167232	3.101154028
4	1.056524	0.065062117	14	2.537002	0.040560816	24	8.268386	1.386545204
5	1.121586	0.249866865	15	2.577563	0.28832506	25	9.654931	0.038462471
6	1.371453	0.097263756	16	2.865888	0.475118284	26	9.693394	0.264305168
7	1.468717	0.072253453	17	3.341007	0.051360888	27	9.957699	2.012010017
8	1.54097	0.394952378	18	3.392368	0.035324991	28	11.96971	0.5048389
9	1.935923	0.242909531	19	3.427692	0.075929969	29	12.47455	0.556327211
10	2.178832	0.008787878	20	3.503622	1.284042624	30	13.03088	

The control chart for cumulative failure data is shown in Figure-1 by considering the failure number as X - axis and Successive difference as Y-axis.

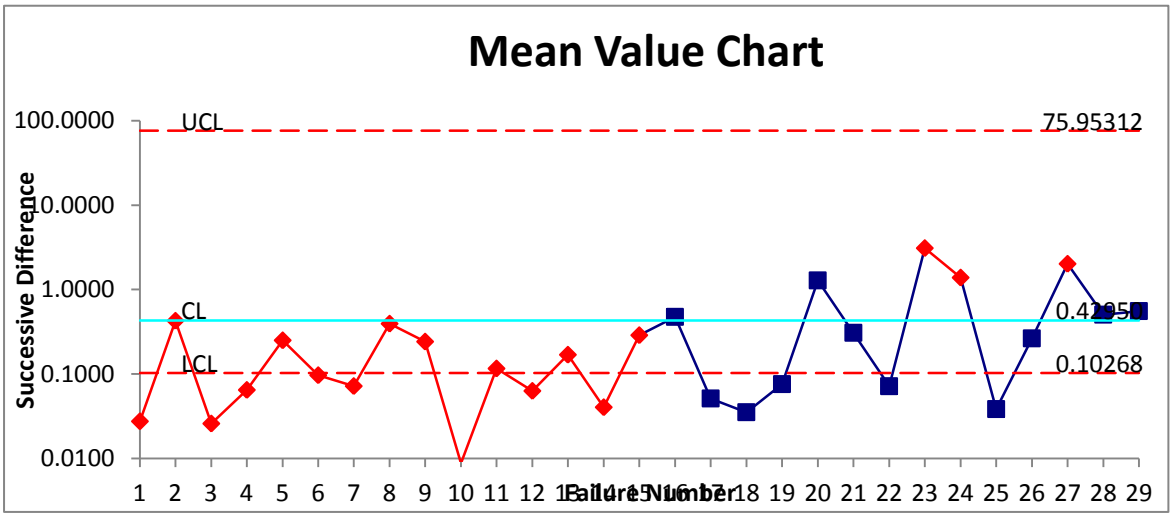

Fig 1. Failure Control Chart for Table 1 (Data Set I)

Table 2 :Data Set II.

Failure Number	Time b/w failures	Failure Number	Time b/w failures	Failure Number	Time b/w failures	Failure Number	Time b/w failures		
1	9	8	58	15	91	22	149		
2	21	9	63	16	92	23	156		
3	32	10	70	17	95	24	247		
4	36	11	71	18	98	25	249		
5	43	12	77	19	104	26	250		
6	45	13	78	20	105				
7	50	14	87	21	116				

Table 3. Times of Successive Failures data

Failure data shown in Table-3 . We found out the values for a & b solving eqns. 7,8 and 9 .

a= 192.0153

b=0.00058

Upon considering the Table 2 data for finding out the control limits ,

$T_L = \frac{1}{b}(e^{0.00135} - 1) = 2.3291$

$T_C = \frac{1}{b}(e^{0.5} - 1) = 1118.485$

$T_U = \frac{1}{b}(e^{0.99865} - 1) = 2956.232$

These control limits and modal parameters are used in finding out $m(T_L)$, $m(T_C)$, $m(T_U)$, that helps in deciding state of the software process.

$m(T_U) = 83.2786$

$m(T_C) = 41.6955$

$m(T_L) = 0.112578$

Table 4. Successive Differences of m(t) for Data Set II.

Failure No.	m(t)	Successive Difference	Failure No.	m(t)	Successive Difference	Failure No.	m(t)	Successive Difference
1	0.43417	0.575399	**10**	3.318758	0.325995	**19**	4.884284	0.274142
2	1.009569	0.523984	**11**	3.365225	0.046467	**20**	4.929887	0.045603
3	1.533553	0.189726	**12**	3.643484	0.278259	**21**	5.42988	0.499993
4	1.723279	0.330986	**13**	3.68977	0.046286	**22**	6.912119	1.482239
5	2.054265	0.094327	**14**	4.105195	0.415425	**23**	7.223175	0.311056
6	2.148591	0.235351	**15**	4.289166	0.183971	**24**	11.16473	3.941556
7	2.383942	0.375185	**16**	4.335095	0.045929	**25**	11.2493	0.084569
8	2.759127	0.233636	**17**	4.472732	0.137637	**26**	11.29155	0.042253
9	2.992763	0.575399	**18**	4.610142	0.13741			

We draw the control chart for cumulative failure data shown in Table-3 by considering the failure number as X – axis and Successive difference as Y-axis.

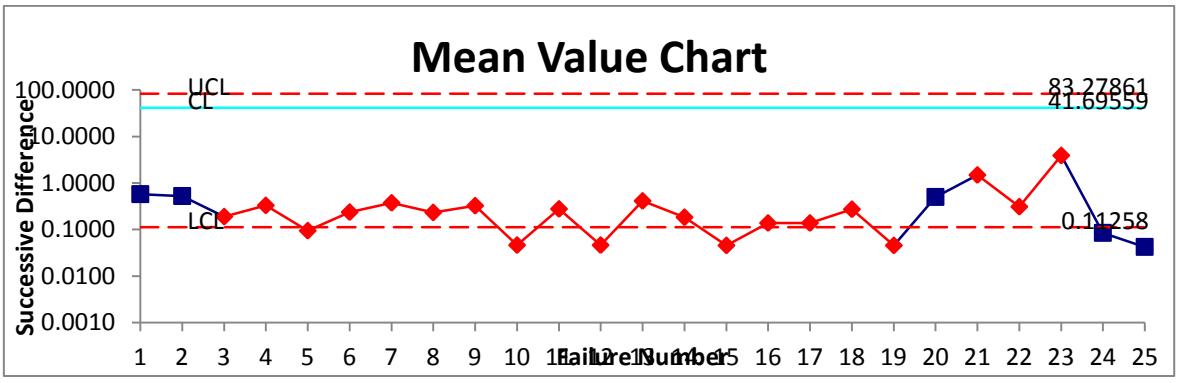

Fig 2. Failure Control Chart for II Data Set Table 4.

A point below the $m(T_L)$ is indicating an Alarm signal and above the $m(T_U)$ indicates better quality. When the points fall with in the control limits ,the software process is in control.

5. CONCLUSION

The results of our study are shown in the control charts in Fig 1., and Fig 2., for data set 1 and 2 respectively. These are Control charts that asseses the confidence limits of software quality. Three reference points UCL, Cl and LCL are also given.. It is heartening to note that the majority of failures are with in the allowable limits of lower level of control and therefore, the system is in control and assure software quality.

6. REFERENCES

[1] J. D. Musa and 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) 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, 327-329, Copyright 2008 ACM 978-1-59593-971-5/08/10.

[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) 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) 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) 472–481.