International Journal of Soft Computing 11 (3): 165-171, 2016
ISSN: 1816-9503

The Comparative Study for Statistical Process Control of Software Reliability Model Based on Finite and Infinite NHPP Using Rayleigh Distribution

Hee-Cheul Kim
Department of Industrial and Management Engineering, Namseoul University, Republic of Korea

Abstract: Software reliability in the software development process is an important issue. Software process improvement helps in finishing with reliable software product. In this field, SPC (Statistical Process Control) is a method of process management through application of statistical analysis which involves and includes the defining, measuring, controlling and improving of the processes. The proposed process involves evaluation of the parameter of the mean value function and hence the mean value function of infinite and finite failure model in order to develop appropriate mean control chart was considered. In this study, a control mechanism, based on time between failures observations using Rayleigh distribution property was proposed which is based on Non Homogeneous Poisson Process (NHPP).

Key words: Statistical process control, rayleigh distribution, non homogeneous poisson process, function, mean value

INTRODUCTION

Software failures caused by failure of computer systems in our society can lead to huge losses. Thus, software reliability in the software development process is an important issue. These issues of the user requirements meet the cost of testing. Software testing (debugging) to reduce costs in terms of changes in the software reliability and testing costs, need to know in advance is more efficient. Thus, the reliability, cost and consideration of release time for software development process are essential. Eventually, the software to predict the contents of a defect in the product development model is needed. Until now, many software reliability models have been proposed. Non-Homogeneous Poisson Process (NHPP) Models rely on an excellent model (Gokhale and Trivedi, 1999; Goel and Okumoto, 1979) in terms of the error discovery process and if a fault occurs, immediately remove the debugging process and the assumption that no new fault has occurred.

The monitoring of software reliability process is not a simple activity. In recent years, several authors have recommended the use of SPC (Statistical Process Control) for software process monitoring (Boffoli *et al*., 2008; Sargut and Demirors, 2006). Over the years, SPC has come to be widely used among others, in manufacturing industries for the purpose of controlling and improving processes (Xie *et al*., 2002).

In this field, SPC (Statistical Process Control) is a method of process management through application of statistical analysis which involves and includes the defining, controlling and improving of the processes (Komuro, 2002). The control chart in measuring software reliability can be used as efficient and appropriate SPC Tools (Burr and Owen, 2006).

In this study, a control mechanism, based on time between failures observations using Rayleigh distribution property was proposed which is based on Non Homogeneous Poisson Process (NHPP). The proposed process involves evaluation of the parameter of the mean value function and hence the mean value function of infinite and finite failure model in order to develop appropriate mean control chart was considered.

MATERIALS AND METHODS

NHPP Model: The mean value function and the intensity function (Yamada *et al*., 1983; Kuo and Yang, 1996) for Non-Homogeneous Poisson Process (NHPP) Model are given by:

$$m(t) = \int_0^t \lambda(s)ds, \ \frac{dm(t)}{dt} = \lambda(t) \tag{1}$$

Therefore, N(t) is known Poisson Probability Density Function (PDF) with the parameter m(t). In other words:

$$p(N(t) = n) = \frac{[m(t)]^n}{n!} e^{-m(t)}, n = 0, 1, \cdots, \infty \tag{2}$$

These time domain models for the NHPP process can be described by the probability of failure are possible.

This model is the failure intensity function $\lambda(t)$ expressed differently, also mean value the function $m(t)$ will be expressed differently.

These models are classified into categories, the finite failure NHPP Models and infinite failure (Dubey, 1973). Finite failure NHPP Models if they are given sufficient time to test, the expected value of faults has a finite expectation value ($\lim m(t) = \theta < \infty$ $t \to \infty$) while infinite failure NHPP Model assumes that the value is infinite.

Let θ denotes the expected number of faults that would be detected given finite failure NHPP Models. Then, the mean value function of the finite failure NHPP Models can also be written as:

$$m(t) = \theta\ F(t) \tag{3}$$

Note that F(t) is a CDF (cumulative distribution function). From Eq. 3, the (instantaneous) failure intensity $\lambda(t)$ in case of the finite failure NHPP Models is given by:

$$\lambda(t) = \theta\ F'(t) = \theta f(t) \tag{4}$$

In finite model, at the time of each repair, a new defect is assumed not to occur. However, the actual situation at the point of repair new failure may occur. Add to this situation, the RVS (Record Value Statistics) model could be used NHPP Model and mean value function was as follows (Kuo and Yang, 1996):

$$m(t) = -\ln(1 - F(t)) \tag{5}$$

Equation 5 is mean value function of infinite failure NHPP Model. Therefore, from Eq. 5 using the related equations of NHPP in Eq. 1, intensity function can be the hazard function h(t). In other words:

$$\lambda(t) = m'(t) = f(t) / (1 - F(t)) = h(t) \tag{6}$$

Note that f(t) is a PDF (the probability density function). Let $\{t_m, n\ 1, 2, ...\}$ denote the sequence of times between successive software failures. Then $t_n$denote the time between $(n = t)_{st}$ and nth failure. Let $x_n$denotes nth failure time, so that:

$$x_n = \sum_{k=1}^{n} t_k\ (k = 1, 2, \cdots, n;\quad 0 \le x_1 \le x_2 \le \cdots \le x_n) \tag{7}$$

The joint density or the likelihood function of $x_1, x_2, ..., x_n$ can be written as (Gokhale and Trivedi, 1999):

$$\begin{aligned} &f_{x_1, x_2, \cdots, x_n}(x_1, x_2, \ldots, x_n) \\ &= e^{-m(x_n)} \prod_{i=1}^{n} \lambda(x_i) \end{aligned} \tag{8}$$

For a given sequence of software failure times $x_1, x_2, ..., x_n$ that are realizations of the random variables, $x_1, x_2, ..., x_n$ the parameters of the software reliability NHPP Models are estimated using the Maximum Likelihood Method (MLE).

Software reliability model based on finite and infinite NHPP using Rayleigh distribution: In this study, Rayleigh distribution model was applied. The Rayleigh distribution was originally derived in connection with a problem in acoustics and has been used in modeling certain features of electronic waves and as the distance distribution between individuals in a spatial Poisson process. Most frequently, however, it appears as a suitable model in life testing and reliability theory (Tadikamalla, 1980). Rayleigh distribution and the distribution function of the probability density function are known as follows (Shin and Kim, 2014):

$$f(t) = 2\ b\ t \exp\left(-b t^2\right), F(t) = 1 - \exp\left(-b t^2\right) \tag{9}$$

The b>0, $t \in (0, \infty)$. As a result, finite failure NHPP intensity function and the mean value function can be expressed, using Eq. 3 and 4 as follows:

$$\begin{aligned} \lambda(t) &= \theta f(t) = 2\theta\ b\ t \exp\left(-b t^2\right) \\ m(t) &= \theta\ F(t) = \theta\left[1 - \exp\left(-b t^2\right)\right] \end{aligned} \tag{10}$$

Note that b refers to the shape parameter. In addition, using Eq. 9, the hazard function is derived as follows:

$$h(t) = f(t) / (1 - F(t)) = 2bt \tag{11}$$

The likelihood function, using Eq. 8 is as follows:

$$\begin{aligned} L_{NHPP}(\Theta|\underline{x}) = &\left[\prod_{i=1}^{n} 2\theta b x_i e^{-b x_i^2}\right] \\ &\exp\left[-\theta(1 - e^{-b x_n^2})\right] \end{aligned} \tag{12}$$

The $x = (x_1, x_2, x_3, ..., x_n)$, θ is parameter space. Using log-likelihood function from Eq. 12, θ_{MLE} and b_{MLE} can be obtained as the solutions of Eq. 13:

$$\hat{\theta} = \frac{n}{1 - e^{-\hat{b} x_n^2}}, \frac{n}{\hat{b}} = \sum_{i=1}^{n} x_i^2 + \hat{\theta} x_n^2 e^{-\hat{b} x_n^2} \tag{13}$$

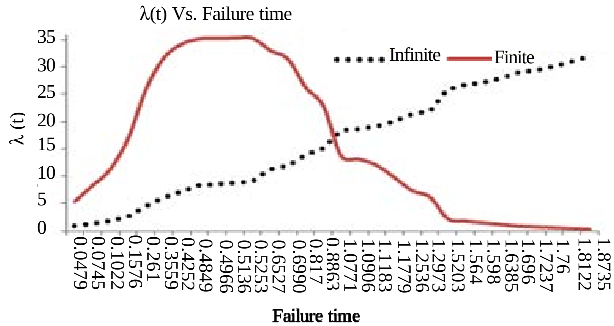

Fig. 1: Types of intensity function infinite and finite NHPP Model

On the other hand, using Eq. 5 and 6, intensity function and mean value function of the infinite NHPP can be expressed as follows:

$$\lambda(t) = m'(t) = f(t)/(1-F(t)) = h(t) = 2bt \quad (14)$$

$$m(t) = -\ln(1-F(t) = bt^2 \quad (15)$$

Note that f (t) is a PDF (the probability density function) F (t) is a CDF (cumulative distribution function) and h (t) is hazard function. The likelihood function, using Eq. 8 is as follows:

$$L_{NHPP}(\Theta|\underline{x}) = \left[\prod_{i=1}^{n} 2bx_i\right]\exp(-bx_n^2) \quad (16)$$

The log-likelihood function to Maximum Likelihood Estimation (MLE) is derived as follows using Eq. 16.

$$\ln L_{NHPP}(\Theta|\underline{x}) = n\ln 2 + n\ln b + \sum_{i=1}^{n}\ln x_i - bx_n^2 = 0 \quad (17)$$

Therefore, the partial differential equation with respect to b using Eq. 17, the maximum likelihood estimation can be calculated as follows:

$$\hat{b}_{MLE} = \frac{n}{x_n^2} \quad (18)$$

Approach for the finite and infinite failure NHPP Model based on software reliability: In this study, for intensity function with firstly increasing and then decreasing (finite NHPP model) and the intensity function with non-decreasing (infinite NHPP Model) want to be compared to the control process (Fig. 1).

The selection of proper SPC charts is essential to effective statistical process control implementation and use. During normal operation, the failures of software system are random events caused by some situation, for example, problem in design or analysis and in some cases insufficient testing of software.

In this study, Rayleigh distribution (Rao *et al*., 2011), for the analysis of time between failures and monitoring of reliability was applied. The upper control limit t_u, lower control limit t_L and centerline t_c, for software reliability models based on 6-sigma probability (0.99865, 0.00135 and 0.5) can be estimated using cumulative distribution function (Prasad *et al*., 2011). The equation for upper control limit, based on finite NHPP of Rayleigh distribution model from Eq. 9 as follows:

$$F(t) = 1 - \exp\left(-bt^2\right) = 0.99865 \quad (19)$$

Using Eq. 19, upper control limit as follows:

$$t = \sqrt{\frac{\ln(0.00135)}{-b}} = t_U \quad (20)$$

Similarly, lower control limit t_L and centerline t_c are derived as follows:

$$t = \sqrt{\frac{\ln(0.99865)}{-b}} = t_L,\; t = \sqrt{\frac{\ln(0.5)}{-b}} = t_C \quad (21)$$

Using successive difference of between m (t)'s (average values for finite NHPP Model) can be estimated the upper control limit (m (t_u)), lower control limit (m (t_u)) and centerline (m (t_c)) were derived as follows: (Prasad *et al*., 2011):

$$m(t_C) = \theta F(t_C) = \theta\left[1-\exp\left(-bt_C^2\right)\right]$$

$$m(t_L) = \theta F(t_L) = \theta\left[1-\exp\left(-bt_L^2\right)\right]$$

$$m(t_U) = \theta F(t_U) = \theta\left[1-\exp\left(-b\, t_U{}^2\right)\right]$$

$$t_U = \sqrt{\frac{\ln(0.00135)}{-b}},\ t_L = \sqrt{\frac{\ln(0.99865)}{-b}},$$

$$t_C = \sqrt{\frac{\ln(0.5)}{-b}}$$

Using similar way, the upper control limit (m (t_u)), lower control limit (m (t_L)) and centerline (m (t_c)), based on successive difference between the average value for infinite NHPP Model were derived as follows:

$$m(t_U) = -\ln(1-F(t_U)) = b\, t_U{}^2$$

$$m(t_C) = -\ln(1-F(t_C)) = b\, t_C{}^2$$

$$m(t_L) = -\ln(1-F(t_L)) = b\, t_L{}^2$$

The control limits are such that the point above the (m (t_u)) (UCL; Upper Control Limit) is an alarm signal. A point below the (LCL; Lower Control Limit) is an indication of bad quality of software because time between failures is more short. A point within the control limits indicates stable process (CL; Control Limit) indicates central line (Ravi1 *et al*., 2011).

RESULTS AND DISCUSSION

Software failure data and process analysis: The procedure of a failures control chart for failure software process was illustrated here. Table 1 shows failure time data of software. In general, the Laplace trend test analysis is used (Kanoun and Laprie, 1996) for reliability property. As a result of this test in this Fig. 2 as indicated in the Laplace factor is between 2 and-2, reliability growth shows the properties. Thus, used to this data it is possible to estimate the reliability (Aneesh, 2011).

In order to facilitate the parameter estimation, in this study, numerical conversion data (Failure time (hours)×0.1) was used. Using the n failure time data, the values of m (t) at t_L,t_c and t_u can be calculated.

Then successive differences of the m(t) can be calculated (n-1) values. The graph with failure times from 1 to n-1 on X-axis, the n-1 values of successive differences m(t)'s on Y-axis and the 3 control lines parallel to X-axis at, (m (t_L)),(m (t_u)), (m (t_c)) respectively were constituted failures control chart to assess the software failure phenomena on the basis of the given failure time data (Prasad *et al*., 2011; Kim, 2013).

Table 1: Failure time data (Hayakawa and Telfar, 2000)

Failure number	Failure time (h)	Failure number	Failure time (h)
1	0.479	16	10.771
2	0.745	17	10.906
3	1.022	18	11.183
4	1.576	19	11.779
5	2.61	20	12.536
6	3.559	21	12.973
7	4.252	22	15.203
8	4.849	23	15.64
9	4.966	24	15.98
10	5.136	25	16.385
11	5.253	26	16.96
12	6.527	27	17.237
13	6.996	28	17.6
14	8.17	29	18.122
15	8.863	30	18.735

Table 2: Parameter estimation and their control limits

Finite model	MLE			Control limits	
	θ	β	m(t_u)	m(t_c)	m(t_L)
NHPP	30.041	1.884	29.989	15.02	0.041
Infinite	-	β	m(t_u)	m(t_c)	m(t_L)
NHPP	-	8.547	6.608	0.693	0.001

MLE: Maximum likelihood estimation

The estimation parameters for each model used to maximum likelihood method. The result of parameter estimation has been summarized in Table 2. These calculations, solving numerically, the initial values given to 0.001 and 15.0 and tolerance value for width of interval given using C-language checking adequate convergent were performed iteration of 100 times. In addition, value of (m (t_L)), (m (t_u)) and (m (t_c)) were calculated.

Table 3, shows the time between failures (cumulative) in hours, corresponding m (t) and successive difference between m (t)'s (infinite and finite NHPP) using Rayleigh distribution (Soni *et al*., 2011) (Table 3).

Using (m (t_L)), (m (t_u)) and (m (t_c)) failure control charts for finite and infinite NHPP Model are showing forms (Fig. 3 and 4). Figure 3 shows the situation of failure control chart using Rayleigh distribution (finite NHPP) is out of control from 26-29th failure number because the corresponding successive difference of m (t) falling below the LCL. It results in out-of-control for the product quality. The assignable cause for this situation is to be investigated and promoted (Prasad *et al*., 2011; Kim, 2013).

On the other hand, using Rayleigh distribution (infinite NHPP), in Fig. 4 for situation of failure control chart, corresponding successive difference of m (t) falling below not the LCL. Therefore, the situation of failure control charts using Rayleigh distribution is in-of-control for the product quality. In addition because infinite NHPP Model than finite NHPP Model appears, the high value (in terms of range) of successive difference between m (t), infinite NHPP Model than finite NHPP Model can be said

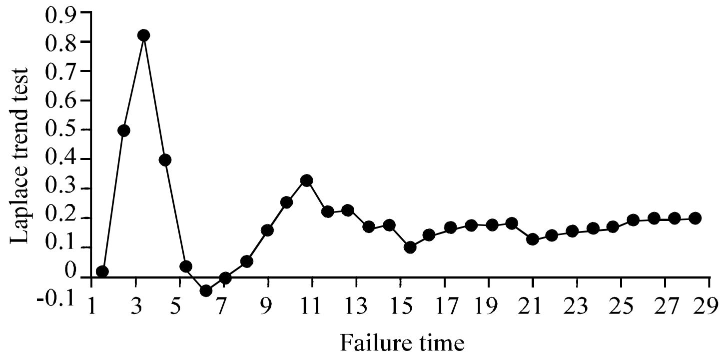

Fig. 2: Results of Laplace trend test

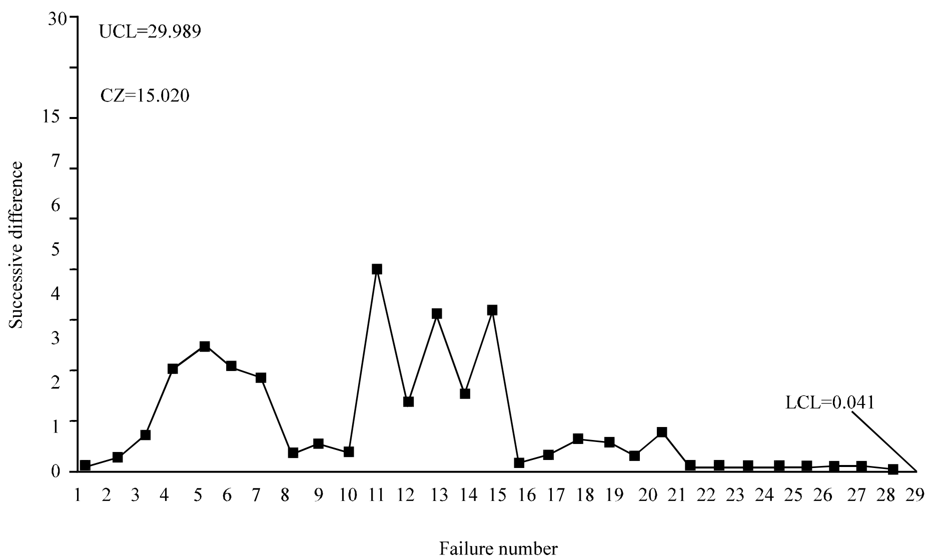

Fig. 3: Result of Mean value chart using finite NHPP

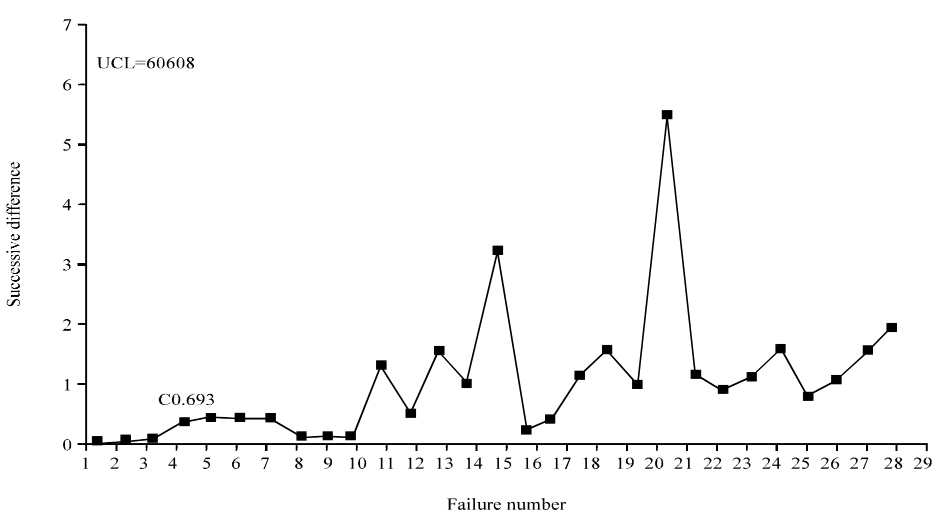

Fig. 4: Result of Mean value chart using infinite NHPP

Table 3: Successive difference of mean value function

Failure number	Failure time (i)	m (i)i = 1, 2, ..., 30 Finite	Infinite	m(I)-m(i-1) successive difference Finite	Infinite
1	0.0479	0.1295	0.0196	0.1829	0.0278
2	0.0745	0.3124	0.0474	0.2728	0.0418
3	0.1022	0.5852	0.0893	0.7878	0.1230
4	0.1576	1.3730	0.2123	2.2443	0.3699
5	0.2610	3.6173	0.5822	2.7587	0.5004
6	0.3559	6.3760	1.0826	2.2937	0.4627
7	0.4252	8.6697	1.5453	2.0788	0.4644
8	0.4849	10.7485	2.0096	0.4128	0.0981
9	0.4966	11.1612	2.1078	0.6009	0.1468
10	0.5136	11.7621	2.2546	0.4137	0.1039
11	0.5253	12.1758	2.3585	4.3987	1.2827
12	0.6527	16.5746	3.6412	1.5162	0.5421
13	0.6996	18.0907	4.1832	3.4045	1.5218
14	0.8170	21.4953	5.705	1.7034	1.0089
15	0.8863	23.1986	6.7139	3.4630	3.2019
16	1.0771	26.6616	9.9158	0.1812	0.2501
17	1.0906	26.8428	10.1659	0.3480	0.5230
18	1.1183	27.1908	10.6888	0.6474	1.1697
19	1.1779	27.8382	11.8585	0.6451	1.5732
20	1.2536	28.4833	13.4317	0.2948	0.9528
21	1.2973	28.7781	14.3845	0.8755	5.3703
22	1.5203	29.6536	19.7548	0.0866	1.1520
23	1.5640	29.7402	20.9068	0.0550	0.9189
24	1.5980	29.7952	21.8257	0.0536	1.1203
25	1.6385	29.8487	22.9460	0.0580	1.6387
26	1.6960	29.9067	24.5847	0.0218	0.8096
27	1.7237	29.9285	25.3944	0.0236	1.0808
28	1.7600	29.9521	26.4752	0.0260	1.5937
29	1.8122	29.9781	28.0689	0.0214	1.9311
30	1.8735	29.9996	30.0000	-	-

that between failure time is long distance. Eventually, infinite NHPP Model more efficient process than finite NHPP model is said to be.

CONCLUSION

There are many charts which use statistical techniques. It is important to use the best chart for the given data, situation and need. There are advances charts that provide more effective statistical analysis.

In this study, graph using given inter failure times was constituted through the estimated mean value function against the failure serial order. The failure control chart has shown out of control signals i.e., below the LCL. Hence, 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. When the time between failures is less than LCL, it is likely that there are assignable causes leading to significant process deterioration and it should be investigated. On the other hand when the time between failures has exceeded the UCL, there are probably reasons that have led to significant improvement.

In this study, in comparison result of infinite and finite NHPP Model (using Rayleigh distribution), infinite NHPP than finite NHPP Model can be said more efficient process because infinite NHPP Model is in-of-control for the product property and finite NHPP model is out-of-control for the product property. As an alternative to this area feel that the content is a valuable research.

ACKNOWLEDGEMENTS

Funding for this study was provided by Namseoul University.

REFERENCES

Aneesh, C., 2011. Efficiency in software development projects. Int. J. Software Eng. Applic., 5: 171-180.

Boffoli, N., G. Bruno, D. Cavivano and G. Mastelloni, 2008. Statistical process control for Software: A sys-tematic approach. Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement. October 09-10, 2008, Kaiserslautern, Germany, pp: 327-329.

Burr, A. and M. Owen, 1996. Statistical Methods for Software Quality: Using Metrics to Control Process and Product Quality. Coriolis Group, UK., ISBN-13: 978-1850321712, Pages: 453.

Dubey, S.D., 1973. Statistical treatment of certain life testing and reliability problems. Final Report, Accession Number: AD0774537, pp: 73-0155.

Goel, A.L. and K. Okumoto, 1979. Time dependent error detection rate model for software reliability and other performance measure. IEEE Trans. Reliab., 28: 206-211.

Gokhale, S.S. and K.S. Trivedi, 1999. A time/structure based software reliability model. Ann. Software Eng., 8: 85-121.

Hayakawa, Y. and G. Telfar, 2000. Mixed poisson-type processes with application in software reliability. Math. Comput. Model., 31: 151-156.

Kanoun, K. and J.C. Laprie, 1996. Trend Analysis. In: Handbook of Software Reliability Engineering, Lyu, M.R. (Ed.). IEEE Computer Society Press, New York, ISBN-13: 9780070394001, pp: 401-437.

Kim, H.C., 2013. Assessing software reliability based on NHPP using SPC. Int. J. Software Eng. Applic., 7: 61-70.

Komuro, M., 2006. Experiences of applying SPC techniques to software development processes. Proceedings of the 28th International Conference on Software Engineering, May 20-28, 2006, Shanghai, China, pp: 577-584.

Kuo, L. and T.Y. Yang, 1996. Bayesian computation for nonhomogeneous poisson process in software reliability. J. Am. Stat. Assoc., 91: 763-773.

Prasad, R.S., K. Ramchand, H. Rao and R.R.L. Kantham, 2011. Software reliability measuring using modified maximum likelihood estimation and SPC. Int. J. Comput. Applic., 21: 1-5.

Rao, R.K.H., R.S. Prasad and R.R.L. Kantham, 2011. Assessing software reliability using SPC: An order statistics approach. Int. J. Comput. Sci. Eng. Applic., 1: 122-131.

Ravi1, D.S.P., N. Supriya and G.K. Mohan, 2011. SPC for software reliability: Imperfect software debugging model. Int. J. Comput. Sci., 8: 219-224.

Sargut, K.U. and O. Demirors, 2006. Utilization of statistical process control (SPC) in emergent software organizations: Pitfalls and suggestions. Software Qual. J., 14: 135-157.

Shin, H. D. and H.C. Kim, 2014. Assessing rayleigh software reliability model based on NHPP using SPC. J. Next Gen. Inform. Technol., 5: 77-84.

Soni, D.R., D.A. Jolly and A. Rana, 2011. Effect of residual defect density on software re-lease management. Int. J. Software Eng. Applic., 5: 151-158.

Tadikamalla, P.R., 1980. A look at the Burr and related distributions. Int. Stat. Rev., 48: 337-344.

Xie, M., T.N. Goh and P. Ranjan, 2002. Some effective control chart procedures for reliability monitoring. Reliab. Eng. Syst. Safe., 77: 143-150.

Yamada, S., M. Ohba and S. Osaki, 1983. S-shaped reliability growth modeling for software error detection. IEEE. Reliab. Trans., R-32: 475-484.