Software Process Control on Time Domain Data: Rayleigh

Mrs P.Padmaja, Dr G.Krishna Mohan, Dr. R.Satya Prasad

Abstract— **Software reliability process can be monitored efficiently by using Statistical Process Control (SPC). It assists the software development team to identify failures and actions to be taken during software failure process and hence, assures better software reliability. In this paper we propose a control mechanism based on the cumulative quantity between observations of time domain failure data using mean value function of Rayleigh distribution, which is based on Non Homogenous Poisson Process (NHPP). The Maximum Likelihood Estimation (MLE) method and Regression methods is used to derive the point estimators of a two-parameter Rayleigh distribution.**

Index Terms— **Software Engineering, Statistical Reliability. Statistical Process Control, Software reliability, Rayleigh Distribution, Mean Value function, Probability limits, Control Charts**

I. INTRODUCTION

Software reliability assessment is important to evaluate and predict the reliability and performance of software system, since it is the main attribute of software. To identify and eliminate human errors in software development process and also to improve software reliability, the Statistical Process Control concepts and methods are the best choice. SPC concepts and methods are used to monitor the performance of a software process over time in order to verify that the process remains in the state of statistical control. It helps in finding assignable causes, long term improvements in the software process. Software quality and reliability can be achieved by eliminating the causes or improving the software process or its operating procedures (Kimura *et al.,* 1995).

The most popular technique for maintaining process control is control charting. The control chart is one of the seven tools for quality control. Software process control is used to secure the quality of the final product which will conform to predefined standards. In any process, regardless of how carefully it is maintained, a certain amount of natural variability will always exist. A process is said to be statistically "in-control" when it operates with only chance causes of variation. On the other hand, when assignable causes are present, then we say that the process is statistically "out-of-control."

The control charts can be classified into several categories, as per several distinct criteria. Depending on the number of quality characteristics under investigation, charts can be divided into univariate control charts and multivariate control charts. Furthermore, the quality characteristic of interest may be a continuous random variable or alternatively a discrete attribute. Control charts should be capable to create an alarm when a shift in the level of one or more parameters of the underlying distribution or a non-random behavior occurs. Normally, such a situation will be reflected in the control chart by points plotted outside the control limits or by the presence of specific patterns. The most common non-random patterns are cycles, trends, mixtures and stratification (Koutras *et al.,* 2007). For a process to be in control the control chart should not have any trend or nonrandom pattern. SPC is a powerful tool to optimize the amount of information needed for use in making management decisions. Statistical techniques provide an understanding of the business baselines, insights for process improvements, communication of value and results of processes, and active and visible involvement. SPC provides real time analysis to establish controllable process baselines; learn, set, and dynamically improves process capabilities; and focus business areas which need improvement. The early detection of software failures will improve the software reliability. The selection of proper SPC charts is essential to effective statistical process control implementation and use. The SPC chart selection is based on data, situation and need (MacGregor, 1995). Many factors influence the process, resulting in variability. The causes of process variability can be broadly classified into two categories, viz., assignable causes and chance causes.

The control limits can then be utilized to monitor the failure times of components. After each failure, the time can be plotted on the chart. If the plotted point falls between the calculated control limits, it indicates that the process is in the state of statistical control and no action is warranted. If the point falls above the UCL, it indicates that the process average, or the failure occurrence rate, may have decreased which results in an increase in the time between failures. This is an important indication of possible process improvement. If this happens, the management should look for possible causes for this improvement and if the causes are discovered then action should be taken to maintain them. If the plotted point falls below the LCL, It indicates that the process average, or the failure occurrence rate, may have increased which results in a decrease in the failure time. This means that process may have deteriorated and thus actions should be taken to identify and the causes may be removed. It can be noted here that the parameter a, b should normally be estimated with the data from the failure process.

The control limits for the chart are defined in such a manner that the process is considered to be out of control when the time to observe exactly one failure is less than LCL or greater than UCL. Our aim is to monitor the failure process and detect any change of the intensity parameter. When the process is normal, there is a chance for this to happen and it is commonly

Manuscript received Aug 01, 2015

Mrs P.Padmaja, Asso.Professor, A.S.R.I.T College,West Godavari
Dr G.Krishna Mohan, Assoc. Professor, Dept. of CSE, KL University Vaddeswaram, Guntur
Dr. R.Satya Prasad, Assoc. Professor, Dept. of CSE Acharya Nagrjuna University

known as false alarm. The traditional false alarm probability is to set to be 0.27% although any other false alarm probability can be used. The actual acceptable false alarm probability should in fact depend on the actual product or process (Swapna and Trivedi, 1998).

II. LITERATURE SURVEY

This section presents the theory that underlies Rayleigh distribution and maximum likelihood estimation for complete data. If 't' is a continuous random variable with pdf: $f(t;\theta_1,\theta_2, \quad ,\theta_k)$. Where $\theta_1,\theta_2, \quad ,\theta_k$ are k unknown constant parameters which need to be estimated, and cdf: $F(t)$. Where, The mathematical relationship between the pdf and cdf is given by: $f(t)=\frac{d(F(t))}{dt}$. Let 'a' denote the expected number of faults that would be detected given infinite testing time in case of finite failure NHPP models. Then, the mean value function of the finite failure NHPP models can be written as: $m(t)=aF(t)$. where, F(t) is a cumulative distribution function. The failure intensity function $\lambda(t)$ in case of the finite failure NHPP models is given by: $\lambda(t)=aF'(t)$ (Pham, 2006).

2.1 NHPP model

The Non-Homogenous Poisson Process (NHPP) based software reliability growth models (SRGMs) are proved to be quite successful in practical software reliability engineering (Musa et al., 1987). The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Model parameters can be estimated by using Maximum Likelihood Estimate (MLE). Various NHPP SRGMs have been built upon various assumptions. Many of the SRGMs assume that each time a failure occurs, the fault that caused it can be immediately removed and no new faults are introduced. Which is usually called perfect debugging. Imperfect debugging models have proposed a relaxation of the above assumption (Ohba, 1984; Pham 1993).

In software reliability, the initial number of faults and the fault detection rate are always unknown. Let $\{N(t),t\geq 0\}$ be the cumulative number of software failures by time 't'. m(t) is the mean value function, representing the expected number of software failures by time 't'. $\lambda(t)$ is the failure intensity function, which is proportional to the residual fault content. Thus $m(t)=a(1-e^{-bt^2})$. where 'a' denotes the initial number of faults contained in a program and 'b' represents the fault detection rate. The maximum likelihood technique can be used to evaluate the unknown parameters. In NHPP SRGM $\lambda(t)$ can be expressed in a more general way as $\lambda(t)=\frac{dm(t)}{dt}$. where $a(t)$ is the time-dependent fault content function which includes the initial and introduced faults in the program and $b(t)$ is the time-dependent fault detection rate. A constant $a(t)$ implies the perfect debugging assumption, i.e no new faults are introduced during the debugging process. A constant $b(t)$ implies the imperfect debugging assumption, i.e when the faults are removed, then there is a possibility to introduce new faults.

2.2 Rayleigh distribution

In recent years the Weibull distribution (Weibull, 1951) has become more popular as a reliability function. It is named after the Swedish scientist Waloddi Weibull. The Weibull distribution has a position of importance in the field of reliability and life testing because of its versatility in fitting time-to-failure distributions. Many researchers considered the distribution and worked on it. Some of them are Kao (1958), Dubey (1963), Menon (1963). The three parameters of the Weibull distribution are θ , β and γ . Where θ and β are known as the scale, shape parameters and γ is known as the location parameter. These parameters are always positive. It is probably the most widely used family of failure distributions, mainly because by proper choice of its shape parameter β, it can be used as an Increasing Failure Rate for $\beta > 1$, Decreasing Failure Rate for $\beta < 1$, or Constant Failure Rate for $\beta = 1$. The Weibull distribution is called Rayleigh distribution at $\beta = 2$, $\gamma = 0$, and Exponential distribution at $\beta = 1$, $\gamma = 0$. The cumulative distribution function is: $F(t)=1-e^{-(bt)^2}$. The mean value function $m(t)=a\left[1-e^{-(bt)^2}\right]$. The failure intensity function is given as: $\lambda(t)=2ab^2te^{-(bt)^2}$.

2.3 MLE (Maximum Likelihood) Parameter Estimation

The idea behind maximum likelihood parameter estimation is to determine the parameters that maximize the probability (likelihood) of the sample data. The method of maximum likelihood is considered to be more robust (with some exceptions) and yields estimators with good statistical properties. In other words, MLE methods are versatile and apply to many models and to different types of data. Although the methodology for maximum likelihood estimation is simple, the implementation is mathematically intense. Using today's computer power, however, mathematical complexity is not a big obstacle. If we conduct an experiment and obtain N independent observations, $t_1,t_2, \quad ,t_N$. The likelihood function [7] may be given by the following product:

$L(t_1,t_2, \quad ,t_N \mid \theta_1,\theta_2, \quad ,\theta_k)=\prod_{i=1}^{N} f(t_i;\theta_1,\theta_2, \quad ,\theta_k)$ Likeli hood function by using λ(t) is: $L=e^{-m(t)}\prod_{i=1}^{n}\lambda(t_i)$

The logarithmic likelihood function is given by:

$$\log L=\log\left(e^{-m(t)}\prod_{i=1}^{n}\lambda(t_i)\right)$$

$$=\sum_{i=1}^{n}\log[\lambda(t_i)]-m(t_n)$$

The maximum likelihood estimators (MLE) of $\theta_1,\theta_2, \quad ,\theta_k$ are obtained by maximizing L or Λ , where Λ is ln L . By maximizing Λ, which is much easier to work with than L, the maximum likelihood estimators (MLE)

of $\theta_1, \theta_2, \ , \theta_k$ are the simultaneous solutions of k equations such as: $\frac{\partial(\Lambda)}{\partial\theta_j} = 0$, j=1,2,…,k

The parameters 'a' and 'b' are estimated using iterative Newton Raphson Method, which is given as

$$x_{n+1} = x_n - \frac{g(x_n)}{g'(x_n)}$$

III. TWO STEP APPROACH FOR PARAMETER ESTIMATION

The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Method of least squares (LSE) or maximum likelihood (MLE) has been suggested and widely used for estimation of parameters of mathematical models (Kapur *et al.,* 2008). Non-linear regression is a method of finding a nonlinear model of the relationship between the dependent variable and a set of independent variables. Unlike traditional linear regression, which is restricted to estimating linear models, nonlinear regression can estimate models with arbitrary relationships between independent and dependent variables. The model proposed in this paper is a non-linear and it is difficult to find solution for nonlinear models using simple Least Square method. Therefore, the model has been transformed from non linear to linear. MLE and LSE techniques are used to estimate the model parameters. Sometimes, the likelihood equations are difficult to solve explicitly. In such cases, the parameters are estimated with some numerical iterative methods (Newton Raphson method). On the other hand, LSE, like MLE, applied for small sample sizes and may provide better estimates (Huang and Kuo, 2002).

1. Algorithm for a 2-step approach of parameter estimation and data as best fit.

- Consider the Cumulative distribution function $F(t)$ and equate to p_i, i.e $F(t) = p_i$, where $p_i = \frac{i}{n+1}$
- Express the equated equation $F(t) = p_i$ as a linear form, $y = mx + b$.
- Find model parameters of mean value function $m(t)$. Where $m(t) = aF(t)$
  - The initial number of faults $\hat{a}$ is estimated through MLE method. Since, it forms a closed solution.
  - The remaining parameters are estimated through LSE regression approach.
- Find the failure intensity function $\lambda(t) = aF'(t)$
- Find likelihood function L
- Find the Log likelihood function log L. (Which comes to be –ve value.)
- The distribution model with the highest –ve value is the best fit.

2. LS (Least Square) estimation

LSE is a popular technique and widely used in many fields for function fit and parameter estimation (Liu, 2011). The least squares method finds values of the parameters such that the sum of the squares of the difference between the fitting function and the experimental data is minimized. Least squares linear regression is a method for predicting the value of a dependent variable *Y*, based on the value of an independent variable *X*.

- *The Least Squares Regression Line*

Linear regression finds the straight line, called the least squares regression line that best represents observations in a bivariate data set. Given a random sample of observations, the population regression line is estimated by: $\hat{y} = bx + a$. where ,'a' is a constant, 'b' is the regression coefficient and 'x' is the value of the independent variable, and ' $\hat{y}$ ' is the predicted value of the dependent variable. The least square method defines the estimate of these parameters as the values which minimize the sum of the squares between the measurements and the model. Which amounts to minimizing the expression: $E = \sum_i \left(Y_i - \hat{Y}_i\right)^2$.

Taking the derivative of E with respect to 'a' and 'b' and equating them to zero gives the following set of equations (called the normal equations):

$$\frac{\partial E}{\partial a} = 2Na + 2b\sum X_i - 2\sum Y_i = 0, \text{ and}$$

$$\frac{\partial E}{\partial b} = 2b\sum X_i^2 + 2a\sum X_i - 2\sum Y_i X_i = 0$$

The solutions of 'a' and 'b' are obtained by solving the above equations. Where, $a = \bar{Y} - b\bar{X}$ and

$$b = \frac{\sum\left(Y_i - \bar{Y}\right)\left(X_i - \bar{X}\right)}{\sum\left(X_i - \bar{X}\right)^2}$$

IV. ILLUSTRATING THE MLE METHOD USING THE RAYLEIGH DISTRIBUTION

4.1 parameter estimation

To estimate 'a' and 'b' , for a sample of n units (all tested to failure), first obtain the likelihood function:

$$L = e^{-a\left(1-e^{-(bt_n)^2}\right)} \prod_{i=1}^{N} 2ab^2 t.e^{-(bt)^2}$$

Taking the natural logarithm on both sides, The Log Likelihood function is given as:

$$\log L = \sum_{i=1}^{n} \log(2ab^2 t_i e^{-(bt)^2}) - a[1 - e^{-(bt_n)^2}] \ .$$

Taking the Partial derivative with respect to 'a' and equating to '0'.

$$a = \frac{n}{\left[1 - e^{-(bt_n)^2}\right]}$$

Taking the Partial derivative with respect to 'b' and equating to'0'.

$$g(b) = \frac{2n}{b} - 2b\sum_{i=1}^{n} t_i^2 - \frac{2.n.b.t_n^2.e^{-(bt_n)^2}}{\left(1 - e^{-(bt_n)^2}\right)} = 0$$

Taking the partial derivative again with respect to 'b' and equating to '0'.

$$g'(b) = 2n\left(\frac{-1}{b^2}\right) - 2\sum_{i=1}^{n} t_i^2 - 2nt_n^2\left\{\frac{e^{-(bt_n)^2}}{\left(1 - e^{-(bt_n)^2}\right)} - \frac{2b^2t_n^2.e^{-(bt_n)^2}}{\left(1 - e^{-(bt_n)^2}\right)^2}\right\}$$

The parameter 'b' is estimated by iterative Newton Raphson Method using $b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)}$, which is substituted in finding 'a'.

4.2 LS Estimation

Procedure to find parameter 'b' using regression approach.

- The cumulative distribution function of Rayleigh is,

$$F(t) = 1 - e^{-\left(\frac{x_i}{\sigma}\right)^2}.$$

- The c.d.f is equated to p_i. Where, $p_i = \frac{i}{n+1}$.
- The equation $F(t) = p_i$ is expressed as a linear form,

$Y_i = C.X_i + D$. Where,

$Y_i = \log\left(-\log\left(1 - p_i\right)\right)$;

$X_i = 2.\log\left(x_i\right)$ and $D = -2\log\sigma$

$\hat{C} = \frac{\sum X_i Y_i - n\overline{Y}\,\overline{X}}{\sum X_i^2 - n\overline{X}^2}$; $\hat{D} = \overline{Y} - \hat{C}\,\overline{X}$;

$\sigma = e^{-\hat{D}/\hat{C}}$

- Where, '$1/\sigma$' is nothing but the parameter 'b' estimated through regression approach.

Table 1: Parameters estimated through MLE and Regression

Data Set (no of observations)	Parameters	
	MLE $\hat{a}$	Regression $\hat{b}$
Xie (30)	30.051592	0.008178
NTDS (26)	28.851930	0.031848
AT&T (22)	23.71965	0.003074
	6	
SONATA (30)	31.961497	0.000345
IBM (15)	19.164356	0.003402
LYU (24)	24.086392	0.127698

4.3 Distribution of Time between failures

Based on the inter failure data given in Table 2 and 3, we compute the software failures process through Mean Value Control chart. We used cumulative time between failures data for software reliability monitoring using Rayleigh distribution. The use of cumulative quality is a different and new approach, which is of particular advantage in reliability.

'$\hat{a}$' and '$\hat{b}$' are Maximum Likely hood Estimates of parameters and the values can be computed using iterative method for the given cumulative time between failures data. Using 'a' and 'b' values we can compute $m(t)$.

Table 2. Time between failures of a software, NTDS

Failure Number	Time between failure(h)	Failure Number	Time between failure(h)
1	9	14	9
2	12	15	4
3	11	16	1
4	4	17	3
5	7	18	3
6	2	19	6
7	5	20	1
8	8	21	11
9	5	22	33
10	7	23	7
11	1	24	91
12	6	25	2
13	1	26	1

Table 3. Time between failures of a software, LYU

Failure Number	Time between failure(h)	Failure Number	Time between failure(h)
1	0.5	13	0.9
2	1.2	14	1.7
3	2.8	15	1.4
4	2.7	16	2.7
5	2.8	17	3.2
6	3.0	18	2.5
7	1.8	19	2.0
8	0.9	20	4.5
9	1.4	21	3.5
10	3.5	22	5.2
11	3.4	23	7.2
12	1.2	24	10.7

Assuming an acceptable probability of false alarm of 0.27%, the control limits can be obtained as (Xie, 2002):

$$T_U = 1 - e^{-(bt)^2} = 0.99865$$

$$T_C = 1 - e^{-(bt)^2} = 0.5$$

$$T_L = 1 - e^{-(bt)^2} = 0.00135$$

These limits are converted to $m(t_U)$, $m(t_C)$ and $m(t_L)$ form. They are used to find whether the software process is in control or not by placing the points in Mean value chart shown in figure 1 and 2 . A point below the control limit $m(t_L)$ indicates an alarming signal. A point above the control limit $m(t_U)$ indicates better quality. If the points are falling within the control limits, it indicates the software process is in stable condition. The values of control limits are as follows.

Table 4.Control Limits

Data set	$m(t_U)$	$m(t_C)$	$m(t_L)$
NTDS	28.812980	14.425965	0.038950
Lyu	24.053875	12.043196	0.032517

Table 5. Successive differences of mean values, NTDS

Cumulative time	m(t)	successive differences	Cumulative time	m(t)	successive differences
9	2.275653	8.129812	87	28.838564	0.006874
21	10.405465	8.234718	91	28.845438	0.001100
32	18.640183	2.462075	92	28.846538	0.002340
36	21.102259	3.326994	95	28.848878	0.001356
43	24.429253	0.723060	98	28.850233	0.001200
45	25.152313	1.414450	104	28.851434	0.000095
50	26.566763	1.333854	105	28.851529	0.000367
58	27.900617	0.436299	116	28.851896	0.000034
63	28.336916	0.314700	149	28.851930	0.000000
70	28.651616	0.026694	156	28.851930	0.000000
71	28.678310	0.103081	247	28.851930	0.000000
77	28.781390	0.010262	249	28.851930	0.000000
78	28.791652	0.046911	250	28.851930	

Table 6. Successive differences of mean values, Lyu

Cumulative time	m(t)	successive differences	Cumulative time	m(t)	successive differences
0.5	0.097994	1.010797	26.1	24.086031	0.000280
1.7	1.108791	5.665069	27.8	24.086311	0.000059
4.5	6.773860	6.969707	29.2	24.086370	0.000021
7.2	13.743567	5.626883	31.9	24.086391	0.000001
10.0	19.370450	3.185178	35.1	24.086392	0.000000
13.0	22.555628	0.853866	37.6	24.086392	0.000000
14.8	23.409494	0.244263	39.6	24.086392	0.000000
15.7	23.653757	0.228031	44.1	24.086392	0.000000
17.1	23.881788	0.180812	47.6	24.086392	0.000000
20.6	24.062599	0.021786	52.8	24.086392	0.000000
24.0	24.084385	0.001241	60.0	24.086392	0.000000
25.2	24.085626	0.000405	70.7	24.086392	

Figure 1 and 2 are obtained by placing the time between failures cumulative data shown in table 5 and 6 on y axis and failure number on x axis and the values of control limits are placed on Mean Value chart. The Mean Value chart shows that the ,10th failure data has fallen below $m(t_L)$ and almost continues to fall below it for both data sets. It indicates the failure process. It is significantly early detection of failures using Mean Value Chart. The software quality is determined by detecting failures at an early stage. No failure data fall outside the $m(t_U)$. It does not indicate any alarm signal

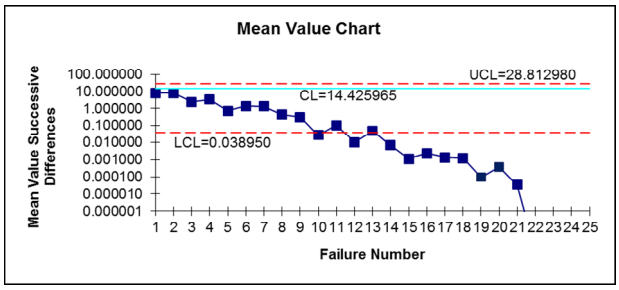

Figure: 1 Mean Value Chart – NTDS

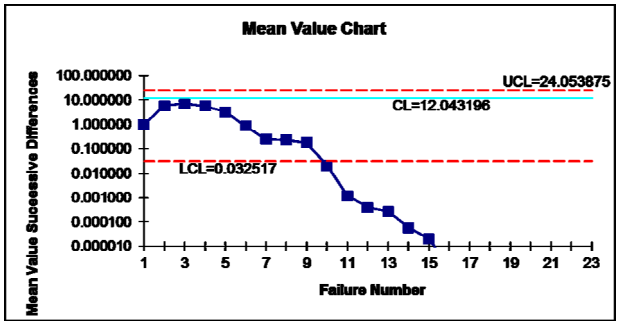

Figure: 2 Mean Value Chart – Lyu

CONCLUSION

The given inter failure times are plotted through the estimated mean value function against the failure serial order. The parameter estimation is carried out by Two step approach for Rayleigh model. The graphs have shown out of control signals i.e below the LCL. Hence we conclude that our method of estimation and the control chart are giving a +ve recommendation for their use in finding out preferable control process or desirable out of control signal. By observing the Mean value Control chart we identified that the failure situation is detected at 10th point of Figure 1 and 2 for the corresponding $m(t)$, which is below $m(t_L)$. It indicates that the failure process is detected at an early stage. Hence our proposed Mean Value Chart detects out of control situation at an earlier. 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 lead to significant improvement.

REFERENCES

[1] Dubey, Satya D., (1963). "On some statistical inferences for Weibull laws", J. Amer. Statist. Assoc., 58, 549.

[2] Huang, C.Y and Kuo, S.Y., (2002). "Analysis of incorporating logistic testing effort function into software reliability modelling", IEEE Transactions on Reliability, Vol.51, No. 3, pp. 261-270.

[3] Kao, J. H. K. (1958). "Computer Methods for estimating Weibull parameters in Reliability Studies", Transactions of IRE-Reliability and Quality Control, 13, 15-22.

[4] Kapur, P.K., Gupta, D., Gupta, A. And Jha, P.C., (2008). "Effect of Introduction of Fault and Imperfect Debugging on Release Time", Ratio Mathematica, 18, pp. 62-90.

[5] Kimura, M., Yamada, S., Osaki, S., 1995. "Statistical Software reliability prediction and its applicability based on mean time between failures". Mathematical and Computer Modeling Volume 22, Issues 10-12, Pages 149-155.

[6] Koutras, M.V., Bersimis, S., Maravelakis,P.E., 2007. "Statistical process control using shewart control charts with supplementary Runs rules" Springer Science + Business media 9:207-224.

[7] Liu, J., (2011). "Function based Nonlinear Least Squares and application to Jelinski-Moranda Software Reliability Model", stat. ME, 25th August.

[8] MacGregor, J.F., Kourti, T., 1995. "Statistical process control of multivariate processes". Control Engineering Practice Volume 3, Issue 3, March 1995, Pages 403-414 .

[9] Menon, M.V., (1963). "Estimation of the shape and scale parameters of the Weibull distribution", Technometrics, 5, 175-182.

[10] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw-Hill, New York.

[11] Ohba, M., 1984. "Software reliability analysis model". IBM J. Res. Develop. 28, 428-443.

[12] Pham. H., 1993. "Software reliability assessment: Imperfect debugging and multiple failure types in software development". EG&G-RAAM-10737; Idaho National Engineering Laboratory.

[13] Pham. H., 2003. "Handbook Of Reliability Engineering", Springer.

[14] Pham. H., 2006. "System software reliability", Springer.

[15] Swapna S. Gokhale and Kishore S.Trivedi, 1998. "Log-Logistic Software Reliability Growth Model". The 3rd IEEE International Symposium on High-Assurance Systems Engineering. IEEE Computer Society.

[16] Weibull, W. (1951). "A Statistical distribution function of wide Applicability", Journal of Applied Mechanics, 18, 293-297.

[17] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 -150¸ 2002.