# Assessing Software Quality with Time Domain Pareto Type II using SPC

K.Sita Kumari
Associate Professor
VRSEC, IT Department
Vijayawada

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

## ABSTRACT
As the usage of software is getting increased day by day, there is a need for software reliability and for this several software reliability growth models exist that are capable of finding the occurrence of errors. It is also possible to find the reliability of time domain models based on order statistics with Non-Homogeneous Poisson Process (NHPP). The conditional failure rate is an important factor for software and order statistics can be used in various applications like characterization of problems, detecting outliers, linear estimation, study of system reliability, life testing, survival analysis, data compression and many others. Using Statistical Process Control (SPC), we can monitor when the software failures occurs, that helps in improving the reliability of software. In this paper, we proposed a control mechanism Pareto Type II model with an on order statistics based on NHPP and the observations are considered for time domain failure data. The unknown parameters are estimated using a well-known estimation method known as Maximum Likelihood Estimation (MLE). The model is analyzed using live data sets.

## Keywords
Order Statistics, Statistical Process Control, Pareto Type II,NHPP, MLE, Control charts.

## 1. INTRODUCTION
Software Reliability plays an important role in software quality. As more and more software is creeping into the embedded system, reliability has become an essential characteristic for the software. The probability that a given program will work as intended by the user without failures in a specified environment and for a specified duration can be termed as software reliability [13]. Software Reliability can prevent major faults that have the possibility of taking human life, money and time. For this a number of models have been developed for better predictions. A common approach for measuring software reliability is by using an analytical model whose parameters are generally estimated from available software failure data. Reliability quantities have been defined with respect to time, although it is possible to define them with respect to other variables. We have taken inter failures time data of Musa 1975) [ 10] and Michael [9] . In reliability study there are two characteristics of a random process: 1) the probability distribution of the random variables, i.e., Poisson and 2) the variation of the process with time. A random process whose probability distribution varies with time is called non homogeneous. The random process for time variation we can define two functions, the mean value function m(t), as the average cumulative failures associated with each time point and the failure intensity function as the rate of change of mean value function.

### 1.1 1.1 Statistical Process Control
Statistical Process Control (SPC) is an important tool with which we can monitor the reliability of developed software. For monitoring the software reliability, software failure data is very much essential that is available in two different types. The time domain data records the time of the failure occurrences and interval domain data records the number of failures in a given time interval. In this paper, we want to monitor the reliability of a software using SPC based on order statistics for time domain data. The main intention of this paper is to give a systematic procedure to show how SPC can be used to monitor the software reliability. In recent years, several authors have recommended the use of SPC for software process monitoring. A few others have highlighted the potential pitfalls in its use [1]. Over the years, SPC has come to be widely used among others, in manufacturing industries for the purpose of controlling and improving the quality of a product [11]. The SPC is applied in the development of software product with the intention to improve the reliability and quality of software [2]. It was proved that SPC can be applied successfully to various processes for development of software, including software reliability process. SPC is traditionally so well adopted in manufacturing industry. In general software development activities are more process centric than product centric which makes it difficult to apply SPC in a straight forward manner. The utilization of SPC for software reliability has been the subject of study of several researchers. A few of these studies are based on reliability process improvement models. They turn the search light on SPC as a means of accomplishing high process maturities. Some of the studies furnish guidelines in the use of SPC by modifying general SPC principles to suit the special requirements of software development [2](Burr and Owen[3]; Flora and Carleton[4]). It is especially noteworthy that Burr and Owen provide seminal guidelines by delineating the techniques currently in vogue for managing and controlling the reliability of software. Significantly, in doing so, their focus is on control charts as efficient and appropriate SPC tools. It is accepted on all hands that statistical process control acts as a powerful tool for bringing about improvement of quality as well as productivity of any manufacturing procedure and is particularly relevant to software development. SPC is a method of process management through application of statistical analysis, which involves and includes the defining, measuring, controlling, and improving of the processes [5].

## 2. ORDER STATISTICS
Order statistics are used in a wide variety of practical situations. Their use in characterization problems, detection of outliers, linear estimation, study of system reliability, life-testing, survival analysis, data compression and many other fields can be seen from the many books [6]. Order statistics deals with properties and applications of ordered random

variables and of functions of these variables. The use of order statistics is significant when failures are frequent or inter failure time is less. Let X denote a continuous random variable with probability density function f(x) and cumulative distribution function F(x), and let $(X_1, X_2, \ldots, X_n)$ denote a random sample of size n drawn on X. The original sample observations may be unordered with respect to magnitude. A transformation is required to produce a corresponding ordered sample. Let $(X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ denote the ordered random sample such that $X_{(1)} < X_{(2)} < \ldots < X_{(n)}$; then $(X_{(1)}, X_{(2)}, \ldots, X_{(n)})$ are collectively known as the order statistics derived from the parent X. The various distributional characteristics can be known from Balakrishnan and Cohen [7]. The inter-failure time data represent the time lapse between every two consecutive failures. On the other hand if a reasonable waiting time for failures is not a serious problem, we can group the inter-failure time data into non overlapping successive sub groups of size 4 or 5 and add the failure times with in each sub group. For instance if a data of 100 inter-failure times are available we can group them into 20 disjoint subgroups of size 5. The sum total in each subgroup would devote the time lapse between every $5^{th}$ order statistics in a sample of size 5. In general for inter-failure data of size 'n', if r (any natural no) less than 'n' and preferably a factor n, we can conveniently divide the data into 'k' disjoint subgroups (k=n/r) and the cumulative total in each subgroup indicate the time between every rth failure. The probability distribution of such a time lapse would be that of the $r^{th}$ ordered statistics in a subgroup of size r, which would be equal to $r^{th}$ power of the distribution function of the original variable (m(t)). The whole process involves the mathematical model of the mean value function and knowledge about its parameters. If the parameters are known they can be taken as they are for the further analysis, if the parameters are not known they have to be estimated using a sample data by any admissible, efficient method of estimation. This is essential because the control limits depend on mean value function, which in turn depends on the parameters. If software failures are quite frequent keeping track of inter-failure is tedious. If failures are more frequent order statistics are preferable [5].

# 3. MODEL DESCRIPTION

For calculating the parameters and the control limits using ordered Statistics approach, we consider here the Pareto Type II distribution [8].

The mean value of Pareto Type II distribution [8] representing the number of failures experienced by the time t is given by

$$m(t) = a\left(1 - \frac{c^b}{(t+c)^b}\right) \qquad 3.1$$

In order to group the Time domain data into non overlapping successive sub groups of size r, we need to take m(t) to the power r

$$m(t) = \left(a\left(1 - \frac{c^b}{(t+c)^b}\right)\right)^r \qquad 3.2$$

Differentiating with respect to t of equation 3.2

$$m'(t) = \frac{d}{dt}\left[a - \frac{ac^b}{(t+c)^b}\right]^r$$

$$= r\left[a - \frac{ac^b}{(t+c)^b}\right]^{r-1} * \left[abc^b * (t+c)^{-(b+1)}\right]$$

$$M'(t) = r\left[a - \frac{ac^b}{(t+c)^b}\right]^{r-1} * \left[\frac{abc^b}{(t+c)^{b+1}}\right] \qquad 3.3$$

The Likelihood function L can be written as

$$L = e^{-m(t)} * \prod_{i=1}^{n} m'(t_i) \qquad 3.4$$

Substituting equations 3.1 and 3.3 in 3.4 we get

$$= e^{-\left[a - \frac{ac^b}{(t+c)^b}\right]^r} * \prod_{i=1}^{n} r\left[a - \frac{ac^b}{(t_i+c)^b}\right]^{r-1}\left[\frac{abc^b}{(t_i+c)^{b+1}}\right]$$

Applying log on both sides:

$$\text{LogL} = -\left[a - \frac{ac^b}{(t+c)^b}\right]^r + \sum_{i=1}^{n}\log\left[r\left(a\frac{ac^b}{(t_i+c)^b}\right)^{r-1}\left(\frac{abc^b}{(t_i+c)^{b+1}}\right)\right]$$

$$\text{logL} = -a^r\left[1 - \frac{c^b}{(t+c)^b}\right]^r + \sum_{i=1}^{n}\log r + \sum_{i=1}^{n}(r-1)\log\left[a - \frac{ac^b}{(t+c)^b}\right] + \sum_{i=1}^{n}[(\log a + \log b + b\log c) - (b+1)\log(t_i+c)] \qquad 3.5$$

Differentiating with respect to 'a'

$$\frac{\partial \text{logL}}{\partial a} = -ra^{r-1}\left[1 - \frac{c^b}{(t+c)^b}\right]^r + 0 + \sum_{i=1}^{n}(r-1)\frac{1}{\left(a - \frac{ac^b}{(t+c)^b}\right)}\left[1 - \frac{c^b}{(t+c)^b}\right] + \frac{n}{a}$$

$$= -ra^{r-1}\left[1 - \frac{c^b}{(t+c)^b}\right]^r + \frac{n(r-1)}{a} + \frac{n}{a}$$

$$\frac{\partial \log L}{\partial a} = 0 \Rightarrow$$

$$a^r = n * \left[\frac{(t+c)^b}{(t+c)^b - c^b}\right]^r \qquad 3.6$$

Differentiating with respect to 'b'

$$\frac{\partial \log L}{\partial b} = a^r r\left(1 - \left(\frac{c}{t+c}\right)^b\right)^{r-1}\left(\frac{c}{t+c}\right)^b \log\left(\frac{c}{t+c}\right) + \sum_{i=1}^{n}[(r-1) * \frac{(t+c)^b}{(t+c)^b - c^b} * -\left(\frac{c}{t+c}\right)^b \log\left(\frac{c}{t+c}\right)] + \sum_{i=1}^{n}[\frac{1}{b} + \log c - \log(t_i+c)]$$

$$\frac{\partial \log L}{\partial b} = nr\left[\frac{(t+c)^b}{(t+c)^b - c^b}\right]\left(\frac{c}{t+c}\right)^b \log\left(\frac{c}{t+c}\right) + \sum_{i=1}^{n}[(r-1) * \frac{(t+c)^b}{(t+c)^b - c^b} * -\left(\frac{c}{t+c}\right)^b \log\left(\frac{c}{t+c}\right)] + \sum_{i=1}^{n}[\frac{1}{b} + \log c - \log(t_i+c)]$$

Taking c=1,

$$= nr\left[\frac{1}{(t+1)^b - 1}\right]\log\left(\frac{1}{t+1}\right) + \sum_{i=1}^{n}(r-1)\log(t_i + 1)\frac{1}{[(t_i+1)^b - 1]} + \frac{n}{b} - \sum_{i=1}^{n}\log(t_i+1)$$

$$g(b) = \frac{nr}{(t+1)^b - 1}\log\left(\frac{1}{t+1}\right) + \sum_{i=1}^{n}(r-1)\log(t_i + 1)\frac{1}{[(t_i+1)^b - 1]} + \frac{n}{b} - \sum_{i=1}^{n}\log(t_i+1) \quad 3.7$$

$$g'(b) = nr\log\left(\frac{1}{t+1}\right)(-1) * \frac{(t+1)^b\log(t+1)}{[(t+1)^b - 1]^2} - \sum_{i=1}^{n}(r-1)\log(t_i+1)(t_i+1)^b\log(t_i+1)\left[(t_i+1)^b - 1\right]^{-2} - \frac{n}{b^2} \quad 3.8$$

$$\log L = -a^r\left[1 - \frac{c^b}{(t+c)^b}\right]^r + \sum_{i=1}^{n}\log r + \sum_{i=1}^{n}(r-1)\log\left[a - \frac{ac^b}{(t+c)^b}\right] + \sum_{i=1}^{n}[(\log a + \log b + b\log c) - (b+1)\log(t_i+c)]$$

Substitute $a^r$ value in this equation and Differentiate with respect to ‘c’,

$$\frac{\partial \log L}{\partial c} = -nr\left[\frac{(t+c)^b}{(t+c)^b - c^b}\right]^r\left[\frac{(t+c)^b - c^b}{(t+c)^b}\right]^{r-1}\left[-b\left(\frac{c}{t+c}\right)^{b-1} * \frac{t}{(t+c)^2}\right] + \sum_{i=1}^{n}(r-1) * \frac{(t+c)^b}{(t+c)^b - c^b}\left[-b\left(\frac{c}{t+c}\right)^{b-1} * \frac{t}{(t+c)^2}\right] + \frac{nb}{c} - \sum_{i=1}^{n}\left(\frac{b+1}{t_i+c}\right)$$

Taking b=1,

$$= \frac{nrct}{(t+c)^1 - c^1} * \frac{1}{c} * \frac{1}{t+c} - \sum_{i=1}^{n} t * \frac{(r-1)c}{(t+c)^1 - c^1} * \frac{1}{c} * \frac{1}{t+c} + \frac{n}{c} - \sum_{i=1}^{n}\frac{2}{t_i+c}$$

$$g(c) = \frac{nr}{t+c} - \sum_{i=1}^{n}\frac{(r-1)}{(t_i+c)} + \frac{n}{c} - \sum_{i=1}^{n}\left(\frac{2}{t_i+c}\right) \quad 3.9$$

$$g'(c) = \frac{-nr}{(t+c)^2} + \sum_{i=1}^{n}\frac{(r-1)}{(t_i+c)^2} - \frac{n}{c^2} + \sum_{i=1}^{n}\frac{2}{(t_i+c)^2} \quad 3.10$$

## 4. MONITORING TIME BETWEEN FAILURES BY DEVELOPING CONTROL CHARTS

Several types of SPC charts are available that uses statistical techniques. It is very much important to select proper charts for implementing statistical process control based on given data, situation and need [9]. Advanced charts are also available that helps us to analyse the statistics very effectively. Variable control charts and Attribute control charts are the basic type of advanced charts that are available and can be used depending on the type of data and nature. Variable control chats are designed to control product or process parameters which are measured on a continuous measurement scale. X-bar, R charts are variable control charts. Attributes are characteristics of a process which are stated in terms of good are bad, accept or reject, etc. Attribute charts are not sensitive to variation in the process as variables charts. Forattribute data there are : p-charts, c-charts, np-charts, and u-charts. The control charts are named as Failure Control Charts that helps to assess the software failure phenomena on the basis of the given inter-failure time data.

## 5. ESTIMATION OF CONTROL LIMITS AND PARAMETERS

Parameter estimation is a statistical method trying to estimate parameters for time domain data based on ordered statistics. Given the data observations and sample size, using the equations 3.6, 3.7, 3.8, 3.9, & 3.10, the parameters ‘a’ ,’b’ and ‘c’ can be computed by using the popular Newton Rapson method . To compute the parameters a program is developed in C language .The equation for mean value function of Pareto Type II Distribution is given by

$$m(t) = a\left(1 - \frac{c^b}{(t+c)^b}\right)$$

The Control limits are obtained as follows: Delete the term ‘a’ from the mean value function. Equate the remaining function successively to 0.99865, 0.00135, 0.5 and solve the value of ‘t’ , for Pareto Type II Distribution , in order to get the usual Six sigma corresponding Upper Control limit, Lower Control limit , Central line.

$$F(t) = 1 - \frac{c^b}{(t+c)^b} = 0.99865$$

$$\Rightarrow \frac{c^b}{(t+c)^b} = 1 - 0.99865$$

$$\Rightarrow \frac{c^b}{(t+c)^b} = 0.00135$$

$$\Rightarrow (t+c)^b = \frac{c^b}{0.00135}$$

Apply log on both sides

$$b\log(t+c) = b\log c - \log 0.00135$$

$$\log(t+c) = \log c - (1/b) * \log 0.00135$$

$$t + c = e^{\{\log c - \left(\frac{1}{b}\right)*\log 0.00135\}}$$

$$t = e^{\{\log c - \left(\frac{1}{b}\right)*\log 0.00135\}} - c = t_U \quad 5.1$$

$$t = e^{\{\log c - \left(\frac{1}{b}\right)*\log 0.99865\}} - c = t_L \quad 5.2$$

$$t = e^{\{\log c - \left(\frac{1}{b}\right)*\log 0.5\}} - c = t_C \quad 5.3$$

The control limits are defined in such a way that the point above the $m(t_U)$ (5.1) Upper Control Limit (UCL) is an alarm signal. A point below $m(t_L)$ (5.2) Lower Control Limit (LCL) is an indication specifying that the quality of software is better. A point within the control limits indicates that the process is stable.

## 5.1 Developing Failure Charts

Given the n inter-failure data the values of m(t) at $t_C$, $t_U$, $t_L$ and at the given n inter-failure times are calculated. Then successive differences of the m(t)'s are taken, which leads to n-1 values. The graph with the said inter-failure times 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 constitutes failures control chart to assess the software failure phenomena on the basis of the given inter-failures time data.

## 6. ILLUSTRATION

The procedure of generating the failures control charts for failure software process is illustrated considering the live data sets [9][10]. The results have given the +ve recommendations that software reliability can be assessed and the failures are detected at an early stage.

**Table 1: Software failure data reported by Musa(1975) [10]**

| Fault | Time | Fault | Time | Fault | Time |
|---|---|---|---|---|---|
| 1 | 3 | 42 | 263 | 83 | 1800 |
| 2 | 30 | 43 | 452 | 84 | 865 |
| 3 | 113 | 44 | 255 | 85 | 1435 |
| 4 | 81 | 45 | 197 | 86 | 30 |
| 5 | 115 | 46 | 193 | 87 | 143 |
| 6 | 9 | 47 | 6 | 88 | 108 |
| 7 | 2 | 48 | 79 | 89 | 0 |
| 8 | 91 | 49 | 816 | 90 | 3110 |
| 9 | 112 | 50 | 1351 | 91 | 1247 |
| 10 | 15 | 51 | 148 | 92 | 943 |
| 11 | 138 | 52 | 21 | 93 | 700 |
| 12 | 50 | 53 | 233 | 94 | 875 |
| 13 | 77 | 54 | 134 | 95 | 245 |
| 14 | 24 | 55 | 357 | 96 | 729 |
| 15 | 108 | 56 | 193 | 97 | 1897 |
| 16 | 88 | 57 | 236 | 98 | 447 |
| 17 | 670 | 58 | 31 | 99 | 386 |
| 18 | 120 | 59 | 369 | 100 | 446 |
| 19 | 26 | 60 | 748 | 101 | 122 |
| 20 | 114 | 61 | 0 | 102 | 990 |
| 21 | 325 | 62 | 232 | 103 | 948 |
| 22 | 55 | 63 | 330 | 104 | 1082 |
| 23 | 242 | 64 | 365 | 105 | 22 |
| 24 | 68 | 65 | 1222 | 106 | 75 |
| 25 | 422 | 66 | 543 | 107 | 482 |
| 26 | 180 | 67 | 10 | 108 | 5509 |
| 27 | 10 | 68 | 16 | 109 | 100 |
| 28 | 1146 | 69 | 529 | 110 | 10 |
| 29 | 600 | 70 | 379 | 111 | 1071 |
| 30 | 15 | 71 | 44 | 112 | 371 |
| 31 | 36 | 72 | 129 | 113 | 790 |
| 32 | 4 | 73 | 810 | 114 | 6150 |
| 33 | 0 | 74 | 290 | 115 | 3321 |
| 34 | 8 | 75 | 300 | 116 | 1045 |
| 35 | 227 | 76 | 529 | 117 | 648 |
| 36 | 65 | 77 | 281 | 118 | 5485 |
| 37 | 176 | 78 | 160 | 119 | 1160 |
| 38 | 58 | 79 | 828 | 120 | 1864 |
| 39 | 457 | 80 | 1011 | 121 | 4116 |
| 40 | 300 | 81 | 445 | -- | -- |
| 41 | 97 | 82 | 296 | -- | -- |

**Table: 2 Parameter estimates and their control limits of 4 and 5 order**

| Data Set | Order | a | b | c | $m(t_U)$ | $m(t_C)$ | $m(t_L)$ |
|---|---|---|---|---|---|---|---|
| Musa | 4 | 26.0268 | 1.00028 | 3.96197 | 25.9916 | 13.0134 | 0.03514 |
| | 5 | 27.0071 | 1.00011 | 4.47055 | 26.9707 | 13.5036 | 0.03646 |

**Table: 3 Successive differences of 4th order m(t)'s of Table 1**

| Fault | 4-order Cumulatives | m(t) | Successive Difference's Of m(t)'s | Fault | 4-order Cumulatives | m(t) | Successive Difference's Of m(t)'s |
|---|---|---|---|---|---|---|---|
| 1 | 227 | 25.57502 | 0.220341 | 18 | 16358 | 26.02049 | 0.000664 |
| 2 | 444 | 25.79536 | 0.095939 | 19 | 18287 | 26.02116 | 0.000624 |
| 3 | 759 | 25.8913 | 0.038078 | 20 | 20567 | 26.02178 | 0.000738 |
| 4 | 1056 | 25.92937 | 0.045599 | 21 | 24127 | 26.02252 | 0.000649 |
| 5 | 1986 | 25.97497 | 0.013356 | 22 | 28460 | 26.02317 | 0.00044 |
| 6 | 2676 | 25.98833 | 0.015245 | 23 | 32408 | 26.02361 | 0.000442 |
| 7 | 4434 | 26.00358 | 0.002987 | 24 | 37654 | 26.02405 | 0.000284 |
| 8 | 5089 | 26.00656 | 0.001126 | 25 | 42015 | 26.02433 | 1.63E-05 |
| 9 | 5389 | 26.00769 | 0.002966 | 26 | 42296 | 26.02435 | 0.000302 |
| 10 | 6380 | 26.01065 | 0.002311 | 27 | 48296 | 26.02465 | 0.000153 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 11 | 7447 | 26.01296 | 0.000829 | 28 | 52042 | 26.0248 | 5.18E-05 |
| 12 | 7922 | 26.01379 | 0.002958 | 29 | 53443 | 26.02486 | 0.000104 |
| 13 | 10258 | 26.01675 | 0.000823 | 30 | 56485 | 26.02496 | 0.000179 |
| | | | | | | | |
| 14 | 11175 | 26.01757 | 0.001015 | 31 | 62651 | 26.02514 | 5.67E-05 |
| 15 | 12559 | 26.01859 | 0.000563 | 32 | 64893 | 26.0252 | 0.000233 |
| 16 | 13486 | 26.01915 | 0.000895 | 33 | 76057 | 26.02543 | 0.000193 |
| 17 | 15277 | 26.02005 | 0.000445 | 34 | 88682 | 26.02562 | |

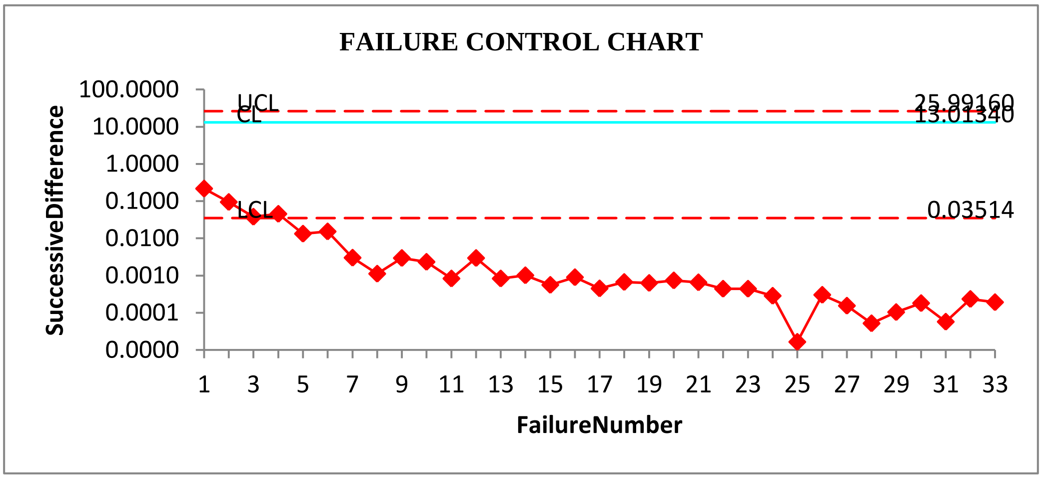


**Fig 1 : Failure Control chart of Table 3**

**Table 4: Successive differences of 5th order m(t)'s of Table 1**

| Fault | 5-order Cumulatives | m(t) | Successive Difference's Of m(t)'s | Fault | 5-order Cumulatives | m(t) | Successive Difference's Of m(t)'s |
|---|---|---|---|---|---|---|---|
| 1 | 342 | 26.65529 | 0.14087 | 18 | 29361 | 27.00302 | 0.000904 |
| 2 | 571 | 26.79616 | 0.086441 | 19 | 37642 | 27.00393 | 0.000334 |
| 3 | 968 | 26.8826 | 0.063804 | 20 | 42015 | 27.00426 | 0.000214 |
| 4 | 1986 | 26.94641 | 0.021791 | 21 | 45406 | 27.00447 | 0.000216 |
| 5 | 3098 | 26.9682 | 0.015042 | 22 | 49416 | 27.00469 | 0.000179 |
| 6 | 5049 | 26.98324 | 0.001234 | 23 | 53321 | 27.00487 | 0.000127 |
| 7 | 5324 | 26.98447 | 0.00375 | 24 | 56485 | 27.00499 | 0.00021 |
| 8 | 6380 | 26.98822 | 0.003126 | 25 | 62661 | 27.00521 | 0.000303 |
| 9 | 7644 | 26.99135 | 0.003824 | 26 | 74364 | 27.00551 | 0.000196 |
| 10 | 10089 | 26.99517 | 0.000972 | 27 | 84566 | 27.0057 | -0.00089 |
| 11 | 10982 | 26.99615 | 0.001379 | 28 | 52042 | 27.00481 | 6.08E-05 |
| 12 | 12559 | 26.99753 | 0.001403 | 29 | 53443 | 27.00487 | 0.000122 |
| 13 | 14708 | 26.99893 | 0.000748 | 30 | 56485 | 27.00499 | 0.00021 |
| 14 | 16185 | 26.99968 | 0.00066 | 31 | 62651 | 27.0052 | 6.65E-05 |
| 15 | 17758 | 27.00034 | 0.000928 | 32 | 64893 | 27.00527 | 0.000273 |
| 16 | 20567 | 27.00127 | 0.001209 | 33 | 76057 | 27.00554 | 0.000226 |
| 17 | 25910 | 27.00247 | 0.000547 | 34 | 88682 | 27.00577 | |

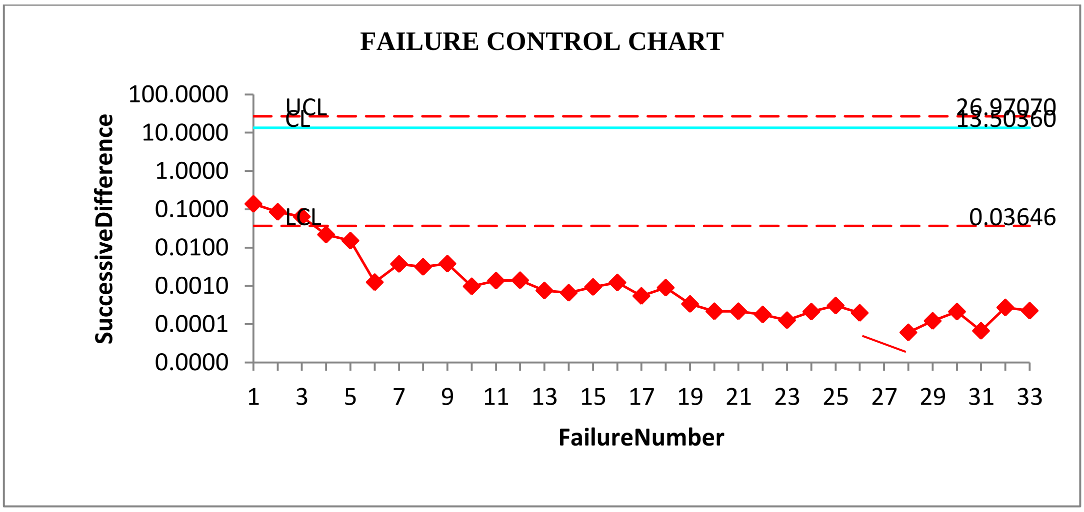


**Fig 2 : Failure Control chart of Table 4**

## 7. CONCLUSION

The Failure control charts of Figure 1 and Figure 2 have shown out of control signals i.e., below LCL. . By observing Failure Control Charts, it is identified that failures are detected at early stages. The early detection of software failure will considerably improve the software reliability. Hence our method of estimation with order statistics and the control charts are giving a +ve recommendation for their use in monitoring the reliability of a software.

## 8. REFERENCES

[1]. N. Boffoli, G. Bruno, D. Cavivano, G. Mastelloni; Statistical process control for Software: a systematic approach; 2008 ACM 978-1-595933-971-5/08/10.

[2]. K. U. Sargut, O. Demirors; Utilization of statistical process control (SPC) in emergent software organizations: Pitfallsand suggestions; Springer Science + Business media Inc. 2006

[3] Burr,A. and Owen ,M.1996. Statistical Methods for Software quality . Thomson publishing Company. ISBN 1-85032-171-X.

[4] Carleton, A.D. and Florac, A.W. 1999. Statistically controlling the Software process. The 99 SEI Software Engineering Symposimn, Software Engineering Institute, Carnegie Mellon University

[5]. K.Ramchand H Rao, R.Satya Prasad, R.R.L.Kantham; Assessing Software Reliability Using SPC – An Order Statistics Approach; IJCSEA Vol.1, No.4, August 2011

[6] Arak M. Mathai ;Order Statistics from a Logistic Dstribution and Applications to Survival and Reliability Analysis;IEEE Transactions on Reliability, vol.52, No.2; 2003

[7] Balakrishnan.N., Clifford Cohen; Order Statistics and Inference; Academic Press inc.;1991

[8] R.Satya Prasad, K.Sita kumari, G.Sridevi; Assessing pareto software reliability using SPC, IJCSI, Vol 10, Issue 1, January 2013

[9] Ronald P.Anjard;SPC CHART selection process;Pergaman 0026-27(1995)00119-0Elsevier science ltd.

[10] R.SatyaPrasad, " Software Reliability with SPC"; International Journal of Computer Science and Emerging Technologies; Vol 2, issue 2, April 2011. 233-237

[11] M.Xie, T.N. Goh, P. Rajan; Some effective control chart procedures for reliability monitoring; Elsevier science Ltd, Reliability Engineering and system safety 77(2002) 143- 150

[12]. R Satya Prasad, N Geetha Rani,"Pareto type II software reliability growth model", International Journal of Software Engineering, Vol. 2, Issue 4, 2011. pp. 81-86

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