Assessing Software Reliability Using SPC – An Order Statistics Approach

K.Ramchand H Rao   Dr. R.Satya Prasad   Dr. R.R.L.Kantham

[1] Department of Computer Science, A.S.N. Degree College, Tenali, India
ramkolasani@gmail.com
[2]Department of Computer Science, Acharya Nagarjuna University, Guntur, India
Profrsp@gmail.com
[3]Dept. of Statistics, Acharya Nagarjuna University, Guntur, INDIA,
kantam_rrl@rediffmail.com

Abstract

There are many software reliability models that are based on the times of occurrences of errors in the debugging of software. It is shown that it is possible to do asymptotic likelihood inference for software reliability models based on order statistics or Non-Homogeneous Poisson Processes (NHPP), with asymptotic confidence levels for interval estimates of parameters. In particular, interval estimates from these models are obtained for the conditional failure rate of the software, given the data from the debugging process. The data can be grouped or ungrouped. For someone making a decision about when to market software, the conditional failure rate is an important parameter. 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. Statistical Process Control (SPC) can monitor the forecasting of software failure and thereby contribute significantly to the improvement of software reliability. Control charts are widely used for software process control in the software industry. In this paper we proposed a control mechanism based on order statistics of cumulative quantity between observations of time domain failure data using mean value function of Half Logistics Distribution (HLD) based on NHPP.

Keywords

Order Statistics, Statistical Process Control (SPC), Half Logistics Distribution (HLD), NHPP

1. INTRODUCTION

The monitoring of Software reliability process is a far from simple activity. 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]. The main thrust of the paper is to formalize and present an array of guidelines in a disciplined process with a view to helping the practitioner in putting SPC to correct use during software process monitoring. Over the years, SPC has come to be widely used among others, in manufacturing industries for the purpose of controlling and improving processes [11]. Our effort is to apply SPC techniques in the software development process so as to improve software reliability and quality [2]. It is reported that SPC can be successfully applied to several processes for software development, 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

DOI : 10.5121/ijcsea.2011.1411

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 also. Viewed in this light, 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. Ordered 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 denote 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 conviently 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 rth ordered statistics in a subgroup of size r, which would be equal to rth 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 know they have to be estimated using a sample data by any admissible, efficient method of distribution. This is essential because the control limits depend on mean value function, which intern depends on the parameters. If software failures are quite frequent keeping track of inter-failure is tedacious. If failures are more frequent order statistics are preferable.

3. Model Description

To calculate the parameter values and control limits using Order Statistics approach, we considered Half Logistic Distribution [8][12].

The mean value function of HLD [8] is $m(t) = \dfrac{a\left(1-e^{-bt}\right)}{\left(1+e^{-bt}\right)}$ 3.1

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

$$[m(t)] = a^r \left(\frac{1-e^{-bt}}{1+e^{-bt}} \right)^r \tag{3.2}$$

$$[m(s_k)] = a^r \left(\frac{1-e^{-bs_k}}{1+e^{-bs_k}} \right)^r \tag{3.3}$$

Derivation with respect to t of equation 4.3.2

$$m'(s_k) = a^r.r\left(\frac{1-e^{-bs_k}}{1+e^{-bs_k}} \right)^{r-1} \left[\frac{be^{-bs_k}\left(1+e^{-bs_k}\right) - \left(1-e^{-bs_k}\right)\left(-be^{-bs_k}\right)}{\left(1+e^{-bs_k}\right)^2} \right] \tag{3.4}$$

$$m'(s_k) = \frac{2b\, a^r.r\, e^{-bs_k}\,(1-e^{-bs_k})^{r-1}}{\left(1+e^{-bs_k}\right)^{r+1}} \tag{3.5}$$

$$L = e^{-m(s_n)} \prod_{k=1}^{n} m'(s_k)$$

$$Log\ L = Log\left[e^{-m(s_n)} \prod_{k=1}^{n} m'(s_k) \right]$$

$$= -m(s_n) + \sum_{k=1}^{n} Log\left[\frac{2b\, a^r.r\, e^{-bs_k}(1-e^{-bs_k})^{r-1}}{\left(1+e^{-bs_k}\right)^{r+1}} \right] \tag{3.6}$$

$$L = -a^r \left(\frac{1-e^{-bs_k}}{1+e^{-bs_k}} \right)^r + \sum_{k=1}^{n} \left[\log 2 + \log b + r\log a + \log r - bs_k + (r-1)\log(1-e^{-bs_k}) - (r+1)\log(1+e^{-bs_k}) \right] \tag{3.7}$$

Derivation with respect to 'a'

$$\frac{1}{L}\frac{\partial L}{\partial a} = -ra^{r-1} \left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}} \right)^r + \sum_{k=1}^{n} \left[0+0+\frac{r}{a}+0-0 = 0-0 \right]$$

$$= \frac{r}{a}\left(-a^{r}\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r}+n\right)$$

$$\frac{1}{L}\frac{\partial L}{\partial a}=0$$

$$\frac{r}{a}\left[-a^{r}\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r}+n\right]=0$$

$$-a^{r}\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r}+n=0$$

$$a^{r}=n\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r}$$

$$\frac{1}{L}\frac{\partial L}{\partial b}=-a^{r}.r\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r-1}\left[\frac{2b\,e^{-bs_n}}{\left(1+e^{-bs_n}\right)^{2}}\right]$$

$$\frac{1}{L}\frac{\partial L}{\partial b}=-a^{r}.r\left(\frac{1-e^{-bs_n}}{1+e^{-bs_n}}\right)^{r-1}\left[\frac{2b\,e^{-bs_n}}{\left(1+e^{-bs_n}\right)^{2}}\right]+\sum_{k=1}^{n}0+\frac{1}{b}+0+0-sk+\frac{r-1}{1-e^{-bs_k}}(be^{-bs_k})-\frac{r+1}{1+e^{-bs_k}}(-be^{-bs_k})$$

$$=\frac{-\,2br\,a^{r}\;e^{-bs_n}\;(1-e^{-bs_n})^{r-1}}{\left(1+e^{-bs_n}\right)^{r+1}}+\sum_{k=1}^{n}\frac{1}{b}-s_k+be^{-bs_k}\left(\frac{r-1}{1-e^{-bs_k}}+\frac{r+1}{1+e^{-bs_k}}\right)$$

$$\frac{1}{L}\frac{\partial L}{\partial b}=\frac{-\,2br\,a^{r}\;e^{-bs_n}\;(1-e^{-bs_n})^{r-1}}{\left(1+e^{-bs_n}\right)^{r+1}}+\frac{n}{b}-ns_k+\sum_{K=1}^{n}be^{-bs_k}\left[\frac{r-1+(r-1)e^{-bs_k}+r+1-(r+1)(e^{-bs_k})}{1-e^{-2bs_k}}\right]$$

$$\frac{1}{L}\frac{\partial L}{\partial b}=\frac{-\,2br\,a^{r}\;e^{-bs_n}\;(1-e^{-bs_n})^{r-1}}{\left(1+e^{-bs_n}\right)^{r+1}}+\frac{n}{b}-ns_k+\sum_{K=1}^{n}be^{-bs_k}\left[\frac{2r-2e^{-bs_k}}{1-e^{-2bs_k}}\right]$$

$$g(b) = \frac{-2br\, a^{r}\, e^{-bs_n}\, (1-e^{-bs_n})^{r-1}}{\left(1+e^{-bs_n}\right)^{r+2}} + \frac{n}{b} - ns_k + \sum_{K=1}^{n} \left[\frac{2be^{-bs_k}\, r - e^{-bs_k}}{1-e^{-2bs_k}} \right]$$

$$g(b) = \frac{2br\, e^{-bs_n}\, (1-e^{-bs_n})^{r-1}}{\left(1+e^{-bs_n}\right)^{r+1}}\, n\left(\frac{1+e^{-bs_n}}{1-e^{-bs_n}}\right)^{r} + \frac{n}{b} - ns_k + \sum_{K=1}^{n} \left[\frac{2be^{-bs_k}\, r - e^{-bs_k}}{1-e^{-2bs_k}} \right]$$

$$= \frac{2nb\, e^{-bs_n}}{\left(1+e^{-bs_n}\right)\left(1-e^{-bs_n}\right)} + \frac{n}{b} - ns_k + \sum_{K=1}^{n} \left[\frac{2be^{-bs_k}\, r - e^{-bs_k}}{1-e^{-2bs_k}} \right]$$

$$= \frac{2nbe^{-bs_n}}{1-e^{-2bs_n}} + \frac{n}{b} - ns_k + \sum_{K=1}^{n} \left[\frac{2be^{-bs_k}\, r - e^{-bs_k}}{1-e^{-2bs_k}} \right] + 2\sum_{K=1}^{n} \frac{\left(bre^{-bs_k} - be^{-2bs_k}\right)}{1-e^{-2bs_k}} + 2r\sum_{K=1}^{n} \frac{be^{-bs_k}}{1-e^{-2bs_k}} - 2\sum_{K=1}^{n} \frac{be^{-2bs_k}}{1-e^{-2bs_k}}$$

$$g(b) = \frac{2nb\, e^{-bs_n}}{1-e^{-2bs_n}} + \frac{n}{b} - ns_k + 2r\sum_{K=1}^{n} \frac{b}{e^{bs_k} - e^{-bs_k}} - 2\sum_{K=1}^{n} \frac{b}{e^{-2bs_k} - 1} \tag{3.8}$$

$$g'(b) = 2n\left[\frac{\left(e^{-bs_n} - bs_n e^{bs_n}\right)\left(1 - e^{-2bs_n}\right) - be^{-bs_n}(2be^{-bs_n})}{\left(1-e^{-2bs_n}\right)^{2}} \right] + n\left(\frac{-1}{b^2}\right) - 0 + 2r\sum_{k=1}^{n} \frac{1.\left(e^{bs_k} - e^{-bs_k}\right) - b\left(s_k e^{bs_k} + s_k e^{-bs_k}\right)}{\left(e^{bs_k} - e^{-bs_k}\right)^{2}}$$

$$-2n\sum_{k=1}^{n} \frac{1.\left(e^{2bs_k} - 1\right) - b2s_k e^{2bs_k}}{\left(e^{2bs_k} - 1\right)^{2}}$$

$$= 2n\, \frac{e^{-bs_n}\left[(1-bs_n)(1-e^{-2bs_n}) - 2b^2 e^{-2bs_n}\right]}{\left(1-e^{-2bs_n}\right)^{2}} + \frac{n}{b^2} + 2r\sum_{k=1}^{n} \frac{e^{bs_k} - e^{-bs_k} - bs_k\left(e^{bs_k} + e^{-bs_k}\right)}{\left(e^{bs_k} - e^{-bs_k}\right)^{2}}$$

$$-2n\sum_{k=1}^{n} \frac{e^{2bs_k} - 1 - 2bs_k e^{2bs_k}}{\left(e^{2bs_k} - 1\right)^{2}} \tag{3.9}$$

4. Monitoring the time between failures using control -chart

The selection of proper SPC charts is essential to effective statistical process control implementation and use. There are many charts which use statistical techniques. It is important to use the best chart for the given data, situation and need [9].There are advances charts that provide more effective statistical analysis. The basic types of advanced charts, depending on the type of data are the variable and attribute charts. 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. However, when dealing with attributes and used properly, especially by incorporating a real time pareto analysis, they can be effective improvement tools. For attribute data there are : p-charts, c-charts, np-charts, and u-charts. We have named the control chart as ***Failures Control Chart*** in this paper. The said control chart helps to assess the software failure phenomena on the basis of the given inter- failure time data

5. Estimation of Parameters and Control Limits

Given the data observations and sample size and using equations (3.1),(3.8),(3.9), the parameters

'a ' and 'b' are computed by using the popular NR method . A program written in C was used for this purpose. The equation for mean value function of Half Logistic Distribution is given by

$$m(t) = a\left[\frac{1-e^{-bt}}{1+e^{-bt}}\right]$$

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 for 't', for half logistic distribution , in order to get the usual Six sigma corresponding control limits, central line.

F(t) = $\frac{1-e^{-bt}}{1+e^{-bt}} = 0.99865$

$\Rightarrow 1 - e^{-bt} = 0.99865\left(1+e^{-bt}\right)$

$\Rightarrow 1 - e^{-bt} = 0.99865 + 0.99865e^{-bt}$

$\Rightarrow 1 - 0.99865 = e^{-bt} + 0.99865.e^{-bt}$

$\Rightarrow 0.00135 = (1 + 0.99865)e^{-bt}$

$\Rightarrow e^{-bt} = \frac{0.00135}{1.99865} = 0.000675456$

$\Rightarrow +bt = \log(0.000675456) = +7.300122639$

It gives

$$t = \frac{7.300122639}{b} = t_U \quad 4.1$$

$$t = \frac{0.002700002}{b} = t_L \quad 4.2$$

$$t = \frac{1.098612289}{b} = t_C \quad 4.3$$

The control limits are such that the point above the $m(t_U)$ (4.1)(UCL) is an alarm signal. A point below the $m(t_L)$(4.3) (LCL) is an indication of better quality of software. A point within the control limits indicates stable process.

5.1 Developing Failures Chart:

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 a failures control chart for failure software process is illustrated with an example here. Table 1 show the time between failures of software product [10].

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

Fault	Time	Fault	Time	Fault	Time	Fault	Time
1	3	35	227	69	529	103	108
2	30	36	65	70	379	104	0
3	113	37	176	71	44	105	3110
4	81	38	58	72	129	106	1247
5	115	39	457	73	810	107	943
6	9	40	300	74	290	108	700
7	2	41	97	75	300	109	875
8	91	42	263	76	529	110	245
9	112	43	452	77	281	111	729
10	15	44	255	78	160	112	1897
11	138	45	197	79	828	113	447
12	50	46	193	80	1011	114	386
13	77	47	6	81	445	115	446
14	24	48	79	82	296	116	122
15	108	49	816	83	1755	117	990
16	88	50	1351	84	1064	118	948
17	670	51	148	85	1783	119	1082
18	120	52	21	86	860	120	22
19	26	53	233	87	983	121	75
20	114	54	134	88	707	122	482
21	325	55	357	89	33	123	5509
22	55	56	193	90	868	124	100
23	242	57	236	91	724	125	10
24	68	58	31	92	2323	126	1071
25	422	59	369	93	2930	127	371
26	180	60	748	94	1461	128	790
27	10	61	0	95	843	129	6150
28	1146	62	232	96	12	130	3321
29	600	63	330	97	261	131	1045
30	15	64	365	98	1800	132	648
31	36	65	1222	99	865	133	5485
32	4	66	543	100	1435	134	1160
33	0	67	10	101	30	135	1864
34	8	68	16	102	143	136	4116

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

Data Set	Order	a	b	$m(t_U)$	$m(t_C)$	$m(t_L)$
Table 1	**4**	2.414736	0.000727	2.411476	1.207368	0.003260
	5	1.933309	0.000114	1.930699	0.966655	0.002610

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

Fault	4-order Cumul-atives	m(t)	Successive Difference's Of m(t)'s	Fault	4-order Cumul-atives	m(t)	Successive Difference's Of m(t)'s
1	227	0.198799753	0.187575193	18	16358	2.414702952	2.49182E-05
2	444	0.386374945	0.263437663	19	18287	2.41472787	6.58063E-06
3	759	0.649812608	0.234105106	20	20567	2.41473445	1.43319E-06
4	1056	0.883917714	0.608610384	21	24127	2.414735884	1.11484E-07
5	1986	1.492528098	0.318281856	22	28460	2.414735995	4.70782E-09
6	2676	1.810809953	0.419004464	23	32408	2.414736	2.76672E-10
7	4434	2.229814417	0.068367032	24	37654	2.414736	5.98011E-12
8	5089	2.298181449	0.022394142	25	42015	2.414736	4.84057E-14
9	5389	2.320575591	0.047885291	26	42296	2.414736	2.10498E-13
10	6380	2.368460882	0.02486038	27	48296	2.414736	0
11	7447	2.393321262	0.006233656	28	52042	2.414736	0
12	7922	2.399554918	0.012395799	29	53443	2.414736	0
13	10258	2.411950717	0.001354859	30	56485	2.414736	0
14	11175	2.413305576	0.000907338	31	62651	2.414736	0
15	12559	2.414212914	0.000256454	32	64893	2.414736	0
16	13486	2.414469368	0.000194113	33	76057	2.414736	0
17	15277	2.414663481	3.94709E-05	34	76057	2.414736	

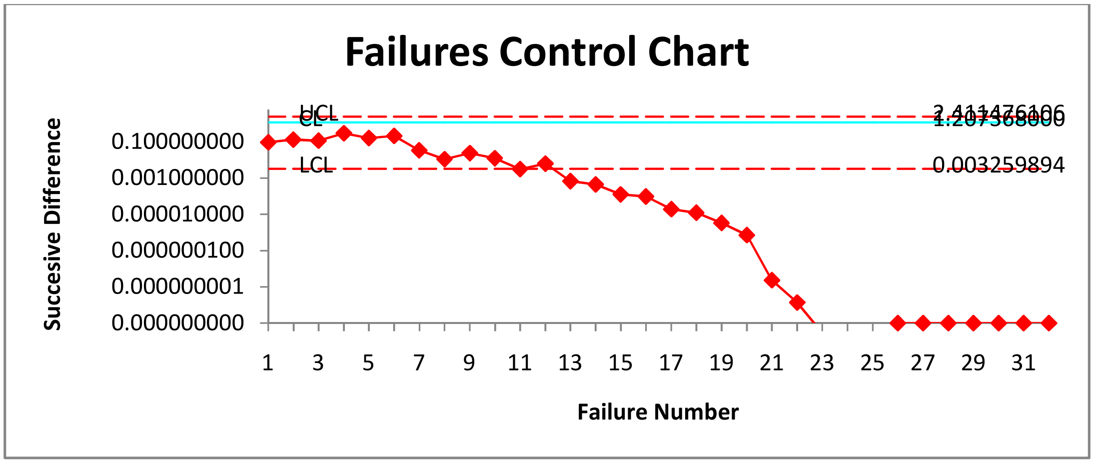

Fig 1: Failures Control Chart of Table 3

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

Fault	5-order Cumul-atives	m(t)	Successive Difference's Of m(t)'s	Fault	5-order Cumula-tives	m(t)	Successive Difference's Of m(t)'s
1	342	0.037683152	0.025218047	15	17758	1.482215615	0.112791696
2	571	0.062901199	0.043662939	16	20567	1.595007310	0.146671923
3	968	0.106564138	0.111360223	17	25910	1.741679234	0.060209283
4	1986	0.217924362	0.119966118	18	29361	1.801888517	0.079206429
5	3098	0.337890480	0.203633678	19	37642	1.881094946	0.020328756
6	5049	0.541524158	0.027802654	20	42015	1.901423702	0.010165625
7	5324	0.569326811	0.104303350	21	45406	1.911589327	0.007940730
8	6380	0.673630162	0.119182170	22	49416	1.919530057	0.004939245
9	7644	0.792812332	0.210720360	23	53321	1.924469302	0.002672472
10	10089	1.003532692	0.069983093	24	56485	1.927141773	0.003114633
11	10982	1.073515785	0.114193668	25	62661	1.930256406	0.002248122
12	12559	1.187709454	0.136486011	26	74364	1.932504528	0.000553008
13	14708	1.324195465	0.081511891	27	84566	1.933057536	
14	16185	1.405707355	0.076508259				

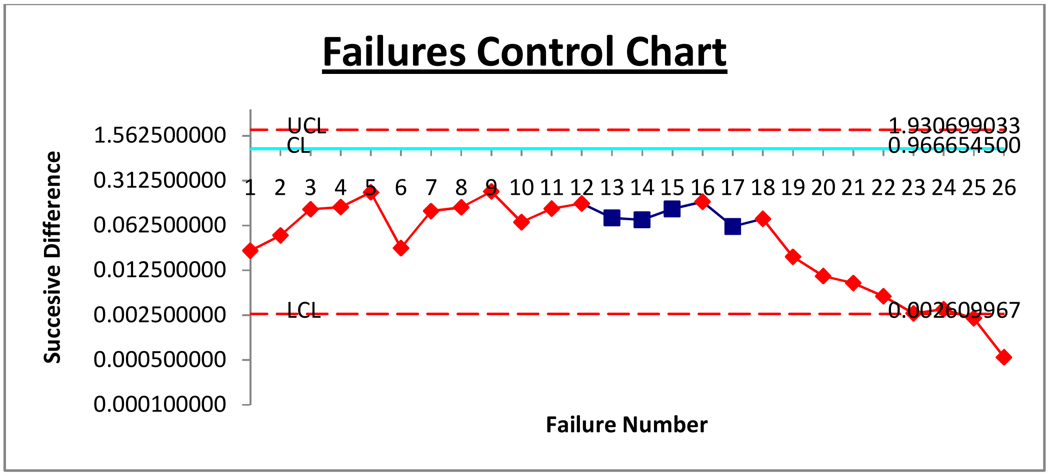

Fig 2 : Failures Control Chart of Table 4

7. CONCLUSION

The Failures Control Charts of Fig 1 to 2 have shown out of control signals i.e. below LCL. By observing Failures Control Charts, we identified that failures situation is detected at an early stages. The early detection of software failure will improve the software reliability. When the control signals are below LCL, it is likely that there are assignable causes leading to significant process deterioration and it should be investigated. Hence, we conclude that our control mechanism proposed in this chapter with order statistics approach giving a positive recommendation for its use to estimate whether the process is in control or out of control.

ACKNOWLEDGMENTS

Our thanks to Department of Computer Science and Engineering; Department of Statistics, Acharya Nagarjuna University; Department of Computer Sceince, Annabathuni Satyanarayya Degree College, Tenali, for providing necessary facilities to carryout the research work.

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] Mutsumi Komuro; Experiences of Applying SPC Techniques to software development processes; 2006 ACM 1-59593-085-x/06/0005.

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

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

[10] Hong Pharm; System Reliability; Springer;2005;Page No.281

[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.satyaprasad, Half Logistic Software Reliability Growth Model,Ph.D. Thesis,2007

Authors

K.Ramchand H Rao, received Master's degree in Technology with Computer Science from Dr. M.G.R University, Chennai, Tamilnadu, India, . He is currently working as Associate Professor and Head of the Department, in the Department of Computer Science, A.S.N. Degree College, Tenali, which is affiliated to Acharya Nagarjuna University. He has 18 years teaching experience and 2 years of Industry experience at Morgan Stanly, USA as Software Analyst. He is currently pursuing Ph.D., at Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. His research area is software Engineering. He has published several papers in National & International Journals.

Dr. R. Satya Prasad received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Andhra Pradesh. He received gold medal from Acharya Nagarjuna University for his out standing performance in Masters Degree. He is currently working as Associate Professor and H.O.D, in the Department of Computer Science & Engineering, Acharya Nagarjuna University. His current research is focused on Software Engineering, Software reliability. He has published several papers in National & International Journals.

R.R.L.Kantam is professor of statistics at Acharya Nagarjuna University,Guntur-India. He has 31 years of teaching experience in statistics at Under Graduate and Post Graduate programs. As researcher in Statistics, he has successfully guided many students for M.Phil and Ph.D. in statistics. He has authored more than 60 research publications appeared various statistics and computer science journals published in India and other countries like US, UK, Germany, Prakistan, Srilanka and Bangladesh. He has been a referee for Journal of Applied Statistics (U.K), METRON (Italy), Pakistan Journal of Statistics (Pakistan), IAPQR – Transactions (India), Assam Statistical Review (India) and Gujarat Statistical Review (India). He has been a special speaker in technical sessions of a number of Seminars and Conferences. His areas of research interest are Statistical Inference, Reliability Studies, Quality Control Methods and Actuarial Statistics. As a teacher his present teaching areas are Probability Theory, Reliability, and Actuarial Statistics. His earlier teaching topics include Statistical Inference, Mathematical Analysis, Operations Research, Econometrics, Statistical Quality control, Measure theory.