A Comparative Study of Software Reliability Growth Models using SPC: Order Statistics approach

M. Anuradha[1] Dr. R. Satya Prasad[2] Dr. G. Sridevi[3]
[1]Research Scholar [2]Associate Professor [3]Professor
[1,2,3]Department of Computer Science & Engineering
[1,2]Acharya Nagarjuna University, Andhra Pradesh, India [3]Malla Reddy Institute of Technology, Hyderabad India

Abstract— This paper presents a comparative study of Burr type distribution with three parameters using order statistics. We propose a control mechanism based on the cumulative quantity between observations of time domain failure data using the mean value function of two Burr type versions i.e., Burr type III and Burr type XII which are based on NHPP.
Key words: Burr Type III Model, Burr Type XII Model, Statistical Process Control (SPC), NHPP, ML Estimation

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. Software Reliability is the probability of failure free operation of software in a specified environment for a specified period of time [1][2]. SRGM is a mathematical model of how the software reliability improves as faults are detected and required [3]. Among all SRGMs developed so far a large family of stochastic reliability models based on a Non-Homogeneous Poisson Process known as NHPP reliability model has been widely used. The main objective is to develop a reliability growth model that can be used to provide quantitative measure for software performance assessment. There is several software reliability growth models exist, one can predict the reliability of software and the number of errors in the software systems. During the past three decades research on software reliability engineering has been conducted and developed numerous statistical models for estimating software reliability. Most existing models for predicting software reliability are based purely on the observation of software product failures where they require a considerable amount of failure data to obtain an accurate reliability prediction.

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.” 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 [4]. 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. Since a, b are the parameters in the proposed distributions, any traditional estimator can be used.

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 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 [5].

II. Related work

Burr type distributions were first introduced in 1942 by Irving W. Burr [6]. Since the corresponding density functions have a wide variety of shapes, this system is useful for approximating histograms. The Burr XII (BXII) distribution, having logistic and Weibull as special sub models, is a very popular distribution for modelling lifetime data and for modelling phenomenon with monotone failure rates. It has been applied in the field of reliability studies and failure time modelling. This section presents the theory that underlies the proposed distributions and maximum likelihood estimation for complete data. If 't' is a continuous random variable with pdf: $f(t;\theta_1,\theta_2,...,\theta_k)$. Where $\theta_1,\theta_2,...,\theta_k$ 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 number of expected 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 distributive function. The failure intensity function $\lambda(t)$ in case of the finite failure NHPP models is given by: $\lambda(t)=aF'(t)$ [7].

A. NHPP Model

There are several software reliability growth models available for use according to probabilistic assumptions. The first one is the Markovian model which is the failure process represented by Markov. The second one is the fault counting model which describes the failure phenomenon by stochastic process like Homogeneous Poisson Process (HPP), Non Homogeneous Poisson Process (NHPP) and Compound Poisson Process. The Non Homogenous Poisson Process (NHPP) based software reliability growth models are proved to be quite successful in practical software reliability engineering [8]. Model parameters can be estimated by using maximum Likelihood Estimation (MLE). The formulation of NHPP model is described in the following lines.

A software system is subject to failures at random times caused by errors present in the system. Let $\{N(t), t \geq 0\}$ be the cumulative number of software failures by time't', where t is the failure intensity function, which is proportional to the residual fault content. As there will be no errors at t=0 we have

$$F(t) = 0$$

Let m(t) represent the expected number of software failures by time 't'. As the expected number of errors remaining in the system is finite, the mean value function m(t) is finite.

$$m(t) = \begin{cases} 0, & t = 0 \\ a, & t \to \infty \end{cases}$$

Where 'a' is the expected number of software errors to be eventually detected.

Suppose m(t) is known to have a Poisson probability mass function with parameters m(t) i.e.,

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

Where N(t) is the cumulative number of failures observed by time 't', N(t) can be modeled as a Poisson Process with a time dependent failure rate. Thus the stochastic behavior of software failure phenomena can be described through the N(t) process. Various time domain models have developed in the literature that describes the stochastic failure process by an NHPP which differ in the mean value function m(t).

III. Descriptions of Burr Type Models

In this section, we propose two variations of Burr type distribution models. The Burr distribution has a flexible shape and controllable scale and location which makes it appealing to fit to data. It is frequently used to model insurance claim sizes [9][10].

A. Burr Type III Model Development

The mean value function of Burr type III model is given by [11]

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

The performance given by the Burr Type III software reliability growth model based on order statistics and whose mean value function is given by

$$m(t) = \left(a\left(1+(t_i)^{-c}\right)^{-b}\right)^r \quad (2)$$

The constants a, b and c in the mean value function are called parameters of the proposed model. To assess the software reliability, it is necessary to compute the expressions for finding the values of a, b and c. For doing this, Maximum Likelihood estimation is used whose Log Likelihood function is given by

$$LLF = \sum_{i=1}^{n} Log\left[\lambda(t_i)^r - m(t_n)^r\right] \quad (3)$$

Differentiating m(t) with respect to 't' we get $\lambda(t)$

$$\lambda(t) = \frac{rabc}{(t_i)^{(c+1)}.\left[1+(t_i)^{-c}\right]^{br+1}} \quad (4)$$

The log likelihood equation to estimate the unknown parameters a, b, c after substituting (2) in (3) is given by

$$LogL = -\left[a\left(1+(t_n)^{-c}\right)^{-b}\right]^r + \sum_{i=1}^{n}\left[\log r + \log a + \log b + \log c\right] + \sum_{i=1}^{n}\left[-(br+1)\log(1+(t_i)^{-c}) - (c+1)\log(t_i)\right] \quad (5)$$

Differentiating Log L with respect to 'a' and equating to 0.

$$a^r = \frac{n(1+(t_n)^{-c})^{br}}{r} \quad (6)$$

Differentiating Log L with respect to 'b' and equating to 0.

$$g(b) = \frac{n}{b} + \sum_{i=1}^{n} r\log(1+(t_i)^{-1}) + \frac{n^2(1+(t_n)^{-1})^{br}}{r}\log(1+(t_n)^{-1}) \quad (7)$$

Again differentiating Log L with respect to 'b', we get

$$g'(b) = \frac{-n}{b^2} + n^2\left[1+(t_n)^{-1}\right]^{br}.\log^2\left[1+(t_n)^{-1}\right] \quad (8)$$

$$g(c) = \frac{n}{c} + \sum_{i=1}^{n}\left[\frac{(r+1)(t_i)-c}{1+(t_i)-c} - 1\right]\log t_i - \frac{n(t_n)-c\log t_n}{(1+(t_n)^{-c})} \quad (9)$$

$$g'(c) = \frac{n}{c^2} + \sum_{i=1}^{n}\left[\frac{(r+1)(\log t_i)^2(t_i)^{-c}}{(1+(t_i)^{-c})^2} + \frac{n\log(t_n)^2(t_n)^{-c}}{(1+(t_n)^{-c})^2}\right] \quad (10)$$

The parameters 'b' and 'c' are estimated by iterative Newton-Raphson method.

$$b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)} \quad \text{and} \quad c_{n+1} = c_n - \frac{g(c_n)}{g'(c_n)}$$

B. Burr Type XII Model Development

The mean value function of Burr type XII model is given by [12]

$$m(t) = a\left[1-\left(1+t^{c}\right)^{-b}\right], \quad t \geq 0 \qquad (1)$$

The parameters a, b, c are estimated with Maximum Likelihood (ML) estimation. 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.

$$m(t) = \left[a\left(1-\left(1+t^{c}\right)^{-b}\right)\right]^{r} \qquad (2)$$

To get the estimates of 'a', 'b' and 'c' for a sample of *n* units, the likelihood function must be obtained first.

$$L = e^{-m(t)}\prod_{i=1}^{n} m'(t_i) \qquad (3)$$

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

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

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

$$\sum_{i=1}^{n}\left(\log a + \log b + \log c + (c-1)\log t_i - (b+1)\log(t_i^{c}+1)\right)$$

Differentiating Log L with respect to 'a', and equating to 0, we get

$$a^{r} = n\left[\frac{\left(t^{c}+1\right)^{b}}{\left(t^{c}+1\right)^{b}-1}\right]^{r} \qquad (4)$$

Differentiating Log L with respect to 'b' and equating to '0'.

$$g(b) = \left[\left(\frac{nr}{(t+1)^{b}-1}\right)\log\left(\frac{1}{1+t}\right)\right] -$$

$$\sum_{i=1}^{n}\left[\frac{r-1}{\left(1+t_i\right)^{b}-1}\log\left(\frac{1}{1+t_i}\right)\right] + \frac{n}{b} - \sum_{i=1}^{n}\log(t_i+1) \qquad (5)$$

Again differentiating $g(b)$ with respect to 'b' and equating to 0.

$$\left(i.e., \frac{\partial^{2} LogL}{\partial b^{2}} = 0\right)$$

$$g'(b) = -nr\log\left(\frac{1}{1+t}\right)\frac{(t+1)^{b}\log(t+1)}{\left[(t+1)^{b}-1\right]^{2}} +$$

$$\sum_{i=1}^{n}\log\left(\frac{1}{1+t_i}\right)\frac{(t_i+1)^{b}\log(1+t_i)}{\left[(t_i+1)^{b}-1\right]^{2}} - \frac{n}{b^{2}} \qquad (6)$$

Differentiating Log L with respect to 'c' and equating to '0'.

$$\frac{\partial LogL}{\partial c} = 0$$

$$g(c) = -nr\frac{\log t}{\left(1+t^{c}\right)} +$$

$$\sum_{i=1}^{n}(r-1)\left(\frac{\log t_i}{1+t_i^{c}}\right) + \frac{n}{c} + \sum_{i=1}^{n}\left[\log t_i - \left(\frac{2}{1+t_i^{c}}\right)t_i^{c}.\log t_i\right] \qquad (7)$$

Again differentiating $g(b)$ with respect to 'b' and equating to $0\left(i.e., \frac{\partial^{2} LogL}{\partial c^{2}} = 0\right)$.

$$g'(c) = nr\left[2\log t.\frac{t^{c}}{(1+t^{c})^{2}}\right]$$

$$-\sum_{i=1}^{n}\left[(r-1)2\log t_i.\frac{t_i^{c}}{\left(1+t_i^{c}\right)^{2}}\right] - \frac{n}{c^{2}} +$$

$$\sum_{i=1}^{n}\left[\log t_i - 2\log t_i\left(t_i^{c}\log t_i\frac{1}{\left(1+t_i^{c}\right)^{2}}\right)\right] \qquad (8)$$

IV. ORDER STATISTICS

Order Statistics can be used in several applications like data compression, survival analysis, Study of Reliability and many others [13]. Let X denote a continuous random variable with probability density function f(x) and cumulative distribution function F(x), and let (X1, X2, …, Xn) 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), …, X(n)) denote the ordered random sample such that X(1) < X(2) < … < X(n); then (X(1), X(2), …, 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 [13].

V. DISTRIBUTION OF TIME BETWEEN FAILURES

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

'a', 'b' and 'c' are Maximum Likelihood Estimates of parameters and the values can be computed using iterative method for the given cumulative time between failures data [14] shown in Table 3. Using 'a', 'b' and 'c' values we can compute m(t).

A. Calculation of Control Limits

$$T_u = 0.99865$$

$$T_c = 0.5$$

$$T_l = 0.00135$$

These limits are converted to m(tu), m(tc)and m(tl) form and are used to find whether the software process is in control or not by placing the points in control charts. The parameter estimates and control limits are shown in Tables 1 and 2.

Model	Order	a	b	c
Burr type III	4	9.485651	0.099997	0.101228
	5	6.15610	0.099996	0.106223
Burr type XII	4	8.500413	0.927761	1.000346
	5	5.40000	0.939094	1.000277

Table 1: Parameter Estimates

Model	Order	$m(t_u)$	$m(t_c)$	$m(t_l)$
Burr type III	4	9.472845	4.742826	0.012806
	5	6.147789	3.07805	0.008311
Burr type XII	4	8.488938	4.250207	0.011471
	5	5.392709	2.700004	0.007289

Table 2: Control Limits

VI. DATA ANALYSIS AND RESULTS

The procedure of a failures control chart for failure software process will be illustrated with an example here.

Table 3 shows the time between failures of a software product.

Failure number	Time Between Failures(hrs)	Failure number	Time Between Failures(hrs)	Failure number	Time Between Failures(hrs)
1	3	47	6	93	2930
2	30	48	79	94	1461
3	113	49	816	95	843
4	81	50	1351	96	12
5	115	51	148	97	261
6	9	52	21	98	1800
7	2	53	233	99	865
8	91	54	134	100	1435
9	112	55	357	101	30
10	15	56	193	102	143
11	138	57	236	103	108
12	50	58	31	104	0
13	77	59	369	105	3110
14	24	60	748	106	1247
15	108	61	0	107	943
16	88	62	232	108	700
17	670	63	330	109	875
18	120	64	365	110	345
19	26	65	1222	111	729
20	114	66	543	112	1897
21	325	67	10	113	447
22	55	68	16	114	386
23	242	69	529	115	446
24	68	70	379	116	122
25	522	71	44	117	990
26	180	72	129	118	948
27	10	73	810	119	1082
28	1146	74	290	120	22
29	600	75	300	121	75
30	15	76	529	122	482
31	36	77	281	123	5509
32	4	78	160	124	100
33	0	79	828	125	10
34	8	80	1011	126	1071
35	227	81	445	127	371
36	65	82	296	128	790
37	176	83	1755	129	6150
38	58	84	1064	130	3321
39	457	85	1783	131	1045
40	300	86	860	132	648
41	97	87	983	133	5485
42	263	88	707	134	1160
43	452	89	33	135	1864
44	255	90	868	136	4116
45	197	91	724		
46	193	92	2323		

Table 3: Software failure data documented in Lyu (1996)

S.No	4-order C_TBF	m(t)	Successive Differences	F. No	4-order C_TBF	m(t)	Successive Differences
1	227	9.063012	0.022075	18	16358	9.188679	0.002814
2	444	9.085086	0.016993	19	18287	9.191492	0.002942
3	759	9.10208	0.010183	20	20567	9.194434	0.003958
4	1056	9.112262	0.01888	21	24127	9.198392	0.004047
5	1986	9.131143	0.008645	22	28460	9.202439	0.00315
6	2676	9.139788	0.014252	23	32408	9.205589	0.003601
7	4434	9.15404	0.003805	24	37654	9.20919	0.002605
8	5089	9.157845	0.001571	25	42015	9.211795	0.000158
9	5389	9.159416	0.004595	26	42296	9.211953	0.003125
10	6380	9.164012	0.004163	27	48296	9.215078	0.001746
11	7447	9.168175	0.001652	28	52042	9.216824	0.000619
12	7922	9.169827	0.006829	29	53443	9.217442	0.001285
13	10258	9.176656	0.002236	30	56485	9.218728	0.002392
14	11175	9.178891	0.003027	31	62651	9.22112	0.000807
15	12559	9.181918	0.001834	32	64893	9.221927	0.00362
16	13486	9.183752	0.003189	33	76057	9.225547	0.003461
17	15277	9.186942	0.001737	34	88682	9.229008	

Table 4: Burr III Successive Differences of 4th order mean value function m(t)

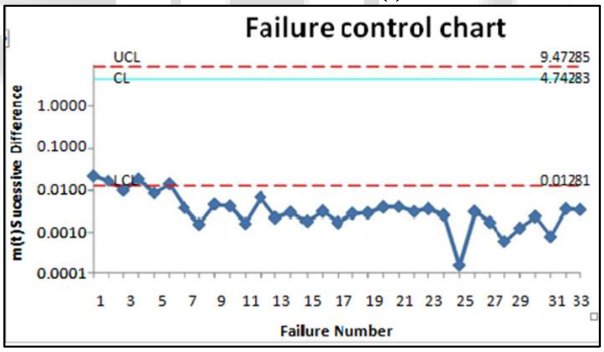

Fig. 1: Failure Control Chart of Table 4

Failure Number	5-Order Cumulative	m(t)	Successive Difference of m(t)
1	342	5.8967019	0.011044
2	571	5.9077458	0.010984
3	968	5.9187302	0.014332
4	1986	5.9330617	0.008513
5	3098	5.9415749	0.009046
6	5049	5.9506206	0.000963
7	5324	5.9515837	0.003258
8	6380	5.9548418	0.003212
9	7644	5.9580537	0.004849
10	10089	5.9629027	0.001462
11	10982	5.9643647	0.002294
12	12559	5.966659	0.002671
13	14708	5.9693304	0.001603
14	16185	5.9709335	0.001543

15	17758	5.9724762	0.002421
16	20567	5.9748968	0.003752
17	25910	5.9786491	0.002004
18	29361	5.9806533	0.003925
19	37642	5.9845787	0.001713
20	42015	5.9862912	0.001201
21	45406	5.9874919	0.001301
22	49416	5.9887927	0.001162
23	53321	5.9899545	0.000876
24	56485	5.9908304	0.001567
25	62661	5.9923972	0.002558
26	74364	5.994955	0.001898
27	84566	5.9968527	

Table 5: Burr III Successive Differences of 5th order mean value function m(t)

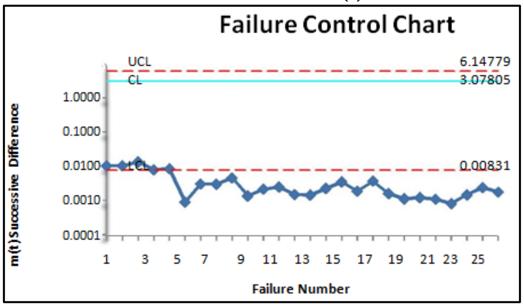

Fig. 2: Failure Control Chart of Table 5

S.No	4-order C_TBF	m(t)	Successive Differences	S.No	4-order C_TBF	m(t)	Successive Differences
1	227	8.445321189	0.025474408	18	16358	8.499368933	0.00010261
2	444	8.470795597	0.011595105	19	18287	8.499471543	9.7259E-05
3	759	8.482390702	0.004752892	20	20567	8.499568802	0.000116249
4	1056	8.487143594	0.005882841	21	24127	8.499685051	0.000103451
5	1986	8.493026435	0.001785099	22	28460	8.499788503	7.09275E-05
6	2676	8.494811534	0.002095369	23	32408	8.499859430	7.19534E-05
7	4434	8.496906903	0.000420763	24	37654	8.499931383	4.65737E-05
8	5089	8.497327666	0.000159701	25	42015	8.499977957	2.68299E-06
9	5389	8.497487367	0.000424185	26	42296	8.499980640	5.0083E-05
10	6380	8.497911552	0.000334393	27	48296	8.500030723	2.56049E-05
11	7447	8.498245945	0.000120842	28	52042	8.500056328	8.68576E-06
12	7922	8.498366787	0.000436288	29	53443	8.500065014	1.7427E-05
13	10258	8.498803075	0.000122968	30	56485	8.500082441	3.03037E-05
14	11175	8.498926043	0.000152693	31	62651	8.500112744	9.6395E-06
15	12559	8.499078736	8.5328E-05	32	64893	8.500122384	3.98104E-05
16	13486	8.499164065	0.000136478	33	76057	8.500162194	3.33161E-05
17	15277	8.499300543	6.83903E-05	34	88682	8.500195510	

Table 6: Burr XII Successive Differences of 4th order mean value function m(t)

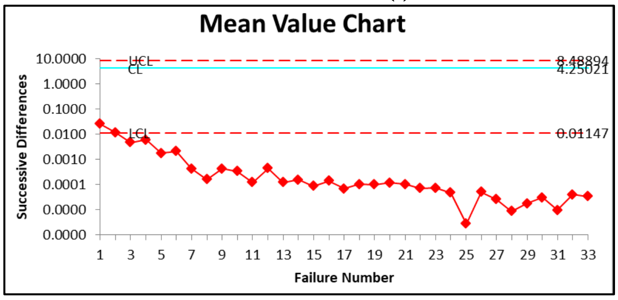

Fig. 3: Failure Control Chart of Table 6

Failure Number	5-Order Cumulative	m(t)	Successive Differences of m(t)
1	342	5.377568567	0.008556724
2	571	5.386125292	0.005418433
3	968	5.391543725	0.004148843
4	1986	5.395692568	0.001470162
5	3098	5.39716273	0.001043809
6	5049	5.398206539	8.71419E-05
7	5324	5.398293681	0.000266672
8	6380	5.398560353	0.000224784
9	7644	5.398785137	0.000278763
10	10089	5.3990639	7.16776E-05
11	10982	5.399135578	0.000102359
12	12559	5.399237937	0.000105075
13	14708	5.399343012	5.64766E-05
14	16185	5.399399489	5.0103E-05
15	17758	5.399449592	7.09189E-05
16	20567	5.399520511	9.35055E-05
17	25910	5.399614016	4.27732E-05
18	29361	5.399656789	7.1438E-05
19	37642	5.399728227	2.66576E-05
20	42015	5.399754885	1.72351E-05
21	45406	5.39977212	1.74142E-05
22	49416	5.399789534	1.45116E-05
23	55321	5.399804046	1.03284E-05
24	56485	5.399814374	1.72391E-05
25	62661	5.399831613	2.50183E-05
26	74364	5.399856632	1.63089E-05
27	84566	5.39987294	

Table 7: Burr type XII Successive Differences of 5th order mean value function m(t)

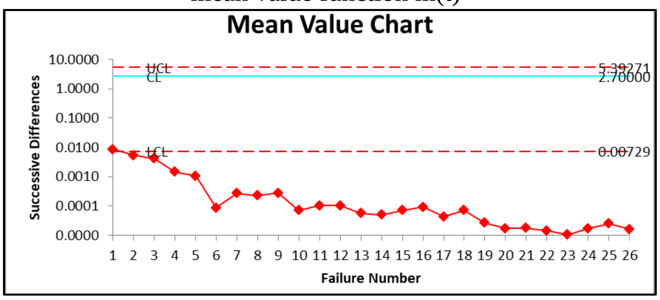

Fig. 4: Failure Control Chart of Table 8

Figures 1, 2, 3 and 4 are obtained by placing the time between failures cumulative data shown in tables 4, 5, 6 and 7 on y axis and failure number on x axis, and the values of control limits are placed on Mean Value chart.

VII. CONCLUSION

This paper presents a comparative study of two software reliability growth models i.e., Burr type III and Burr type XII using SPC technique is applied to sample data with order statistics approach. By observing the Mean Value control chart of Burr type III, we identified that the failure detection is situated at 3rd point of Table-4 for the corresponding m(t) in 4th order statistics and at 6th point of Table-5 for the corresponding m(t) in 5th order statistics, which is below m(tL).

Whereas by observing the Mean Value control chart of Burr type XII, we identified that the failure detection is situated at 3rd point of Table-6 for the corresponding m(t) in 4th order statistics and at 2nd point of Table-7 for the corresponding m(t) in 5th order statistics, which is below m(tL). It is significantly early detection of failure through Burr type XII model using Mean Value Chart. The software quality is determined by detecting failures at an early stage. The early detection of software failure will improve the software quality.

REFERENCES

[1] Lyu, M.R., (1996), "Handbook of Software Reliability Engineering", MCGraw-Hill, New York.
[2] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw -Hill, New York.
[3] Quadri, S.M.K and Ahmad, N., (2010), "Software Reliability Growth Modelling with new modified Weibull testing-effort and optimal release policy", International Journal of Computer Applications, Vol.6, No.12.
[4] MacGregor, J.F., Kourti, T., 1995. "Statistical process control of multivariate processes". Control Engineering Practice Volume 3, Issue 3, March 1995, Pages 403-414.
[5] 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.
[6] Burr (1942), "Cumulative Frequency Functions", Annals of Mathematical Statistics, 13, pp. 215-232.
[7] Pham. H., 2006. "System software reliability", Springer.
[8] Musa, J.D., Iannino, A., Okumoto, k., 1987. "Software Reliability: Measurement Prediction Application". McGraw -Hill, New York.
[9] Hee-cheul Kim., "Assessing Software Reliability based on NHPP using SPC", International Journal of Software Engineering and its Applications, vol.7,No.6 (2013), pp.61-70.
[10] Gutta Sridevi, C.M.Sheela Rani, "Comparison of Software reliability analysis for Burr distribution", Journal of Theoretical and Applied Information Technology (JATIT), 10th November 2015. Vol.81. No.1, ISSN: 1992-8645, E-ISSN: 1817-3195.
[11] K.Sobhana, R.Satya Prasad and Ch.Smitha Chowdary, "Monitoring Software Quality using SPC – An Order Statistics approach", International Journal of Applied Engineering Research, Research India Publications, Volume 10, No. 12 (2015).
[12] R.Satya Prasad, M.Anuradha and Gutta Sridevi, "Burr Type XII Software Reliability Growth Model: An Order Statistics Approach", International Journal of Pharmacy and Technology, Vol. 9, Issue No. 2, June 2017.
[13] Balakrishnan.N, Clifford Cohen; Order Statistics and Inference; Academic Press Inc; 1991.
[14] Xie, M., Goh. T.N., Ranjan.P., "Some effective control chart procedures for reliability monitoring" -Reliability engineering and System Safety 77 143 -150¸ 2002.
[15] Goel, A.L., Okumoto, K., 1979. Time-dependent error detection rate model for software reliability and other performance measures. IEEE Trans. Reliab. R-28, 206-211.