Nat.Volatiles&Essent.Oils,2021;8(5):5427-5434

Measurement Of Software Reliability With SPC Using Arranged Sample Approach

[1]V.Rama Krishna [2]M R Narsinga Rao and [3]T Subhmastan Rao

[1]Professor, Vignan's Foundation for Science Technology and Research, Vadlamudi, Guntur vramakrishna2006@gmail.com
[2]Professor, K L University, Vaddeswaram, Guntur, India, manda.ramanarasingarao@gmail.com
[3]Associate Professor, CMR Technical Campus, Hyderabad, mastan1061@gmail.com

Abstract
In software industry, to release the product to the customers is a crucial factor and depends on conditional failure rate. The associated reliability models will be developed using Arranged sample or Non-Homogeneous Poisson process (NHPP) on the occurrence times of bugs in softwre debugging. Arranged samples are very much used in the problems like detection of outliers, system reliability and life testing. Process Control is useful to forecast failures in the software operation to see progress in the software reliability. Online process improvement methods are widely used in software industry to monitor the software process. In this paper we put a procedure for Arranged sample based on cumulative value between times of failures with Generalized Half Logistic Distribution type - I's mvf.
Key words: Arranged sample, Statistical Process Control (SPC), Generalized Half Logistic Distribution (GHLD), NHPP, mean value function (mvf).

1. Introduction

Cycle of Software Reliability [12][13][14] is far from being a simple operation and cannot be monitored easily. A variety of authors have proposed using SPC for tracking device processes over the last years. Many others have been highlighted latent problems in its use[1]. The paper's key thrust is to give a set of recommendations to the user in proper usage of SPC in software development. SPC's come over the years to be widely used, among others, in the processing industries for the production and production of process Enhancement[11]. An attempt is made is to practice SPC techniques in developing software process for the quality of software[2]. It's said SPC may be successfully applied to several software development processes including reliability of the software product. In the manufacturing industry SPC is well adopted in measuring the quality. In software generally development practices are more process-centered than product-centered, making straightforward implementation of SPC is difficult. The use of SPC has been for software reliability is the topic of many researchers' studies. Many of the studies suggest the use of SPC by amending general SPC standards to satisfy software development technical requirements[2] (Burr and Owen[3]); Carleton and Flora[4]). It is especially worthy of notice that Burr and Owen give seminal guidance by delineating currently in use of online process improvement methods for effective and reasonable tools to SPC. It will be a process management that embodies defining, development and improvement of processes [5].

2. Arranged sample:

In different functional conditions the order figures are used. Its use in characterizing problems like outlier's identification, linear estimation, device reliability analysis, life-testing, survival. We can see different applications in many books[6]. Please order Numbers discuss the features and uses of random variables and related functions. The usage of Arranged sample is significant when malfunction occur time is slower. Let Y denote a continuous random variable with pdf f(y) and cdf F(y), and let (Y1 , Y2 , …, Yn) denote a random sample of size n drawn on Y. The original sample observations regarding magnitude may be unordered. A corresponding ordered sample shall be produced with transformation. Letting (Y(1), Y(2), ..., Y(n)) denote the random sample ordered such that Y(1) < Y(2) < ... < Y(n); then (Y(1), Y(2), ..., Y(n)) collectively known as the parent Y-derived Arranged sample. The various distributional characteristics can be from Balakrishnan and Cohen [7]. Data on inter-fault time represents the lapse of time between each consecutive failure. In contrast, if reasonable waiting times for failures are not a big issue, we can make a set of interfault time data in set of size 4 or 5 which are not overlapping and add the failure duration within each set. For example, if there is a data of 100 interfault times we can make a set of them in 20 size 5 Disjoint Sets. The sum total in each set would indicate the lapse of time in a set of size 5 between every 5th Arranged sample. Generally for size 'n' inter-failure results, If r is < 'n' and then we can divide the data appropriately into the period shows the 'k' disjoint sets (k = n / r) and the cumulative sum in each set for any rth failure. The distribution of probability for such a period of time will be the one of rth ordered statistics in the size r set will be equal to the distribution rth power of m(t). The entire process comprises the mathematical model formulation of the mvf and the knowledge of its parameters. If you know the parameters, then can be taken as it is for further analysis unless the parameters are known by estimation techniques using sample data by any allowable, efficient distribution method. That's it important because the limits of the process control depend on the function of the mean value which depends on the internal value perfect constants of the distribution.

3. Model formulation

We use the Arranged sample method to measure parameter values and control limits, The Generalized Half Logistic Distribution type – I [8][15] was considered.

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

To get r^{th} Arranged sample, take $(m(t))^r$

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

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

Derivation with respect to s_k we get,

$$m'(s_k) = \frac{2a^r\,\theta r e^{-bs_k}(1-e^{-bs_k})^{\theta r-1}}{(1+e^{-bs_k})^{\theta r+1}}$$

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

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

$= -m(s_n) + \sum_{k=1}^{n} log\left[\frac{2a^r\,\theta r e^{-bs_k}(1-e^{-bs_k})^{\theta r-1}}{(1+e^{-bs_k})^{\theta r+1}}\right]$

$= -a^r\left[\frac{1-e^{-bs_k}}{1+e^{-bs_k}}\right]^{\theta r} + \sum_{k=1}^{n}\left[log\, log\, 2 + log\, log\, b + r\, log\, log\, a + log\, log\, \theta + log\, log\, r - bs_k\right.$]

$+(\theta r-1)\log(1-e^{-bs_k}) - (\theta r+1)\log(1+e^{-bs_k})]$

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

$$= \frac{r}{a}\left[-a^r\left[\frac{1-e^{-bs_k}}{1+e^{-bs_k}}\right]^{\theta r} + n\right]$$

$$\frac{1}{L}\frac{\partial L}{\partial a} = 0 \Rightarrow a^r = n\left[\frac{1-e^{-bs_k}}{1+e^{-bs_k}}\right]^{\theta r} \qquad 3.1$$

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

On simplification,

$$g(b) = -\frac{2a^r\, \Theta r e^{-bs_k}\left(1-e^{-bs_k}\right)^{\Theta r-1}}{\left(1+e^{-bs_k}\right)^{\Theta r+1}} + \frac{n}{b} - ns_k + \sum_{k=1}^{n} \left[\frac{2be^{-bs_k}\, r - e^{-bs_k}}{1-e^{-2bs_k}}\right] \qquad 3.2$$

$$g'(b) = \frac{2ne^{-bs_n}\left[(1-bs_n)\left(1-e^{-2bs_n}\right) - 2b\ 2\ e^{-2bs_n}\right]}{\left(1-e^{-2bs_k}\right)^2} + \frac{n}{b^2}$$

$$+\, 2\theta r \sum_{k=1}^{n} \frac{e^{bs_k} - e^{-bs_k} - bs_k\left(e^{bs_k} - e^{-bs_k}\right)}{\left(e^{bs_k} - e^{-bs_k}\right)^2} - 2n \sum_{k=1}^{n} \frac{e^{2bs_k} - 1 - 2bs_k e^{2bs_k}}{\left(e^{2bs_k} - 1\right)^2} \qquad 3.3$$

4.T**ime between failures served are observed using control -chart**

For an effective statistical process control, selection of the appropriate control chart has considerable importance for the given data, context and requirement [9]. We have basically two types of online process improvement methods. Process parameters are measured on continuous scale. Control charts for variables like X-bar chart and R-chart serve to m the monitor process parameters. Process characteristics such as good or bad, accept or reject etc., can be handled by attribute online process improvement methods. For dealing failures in a process we have p-chart and np-chart and u-chart can be used to study the malfunction phenomenon in the process for the given inter –failure time data.

5. **Estimation of parameters and online process improvement methods**

For a given data using equations (3.1),(3.2),(3.3), the parameters in m(t) are deduced by using the Newton-Raphson method. The equation for mvf of Generalized Half Logistic distribution type - I is given by

$$F(t) = a\left[\frac{1-e^{-bt}}{1+e^{-bt}}\right]^{\Theta} \qquad 5.1$$

The limits are deduced on removing the term a in 5.1. Equate the resultant function to 0.99865, 0.00135, 0.5 successively for‘t’, to obtain the corresponding control limits, central line.

$F(t) = \left[\frac{1-e^{-bt}}{1+e^{-bt}}\right]^{\Theta}$ =0.99865 on solving for $t = \frac{7.98656}{b}$ = t_U for Θ = 2 5.2

Similarly we can obtain t_L, t_C on equating F(t) to 0.00135, 0.5

$F(t) = \left[\frac{1-e^{-bt}}{1+e^{-bt}}\right]^{\Theta}$ =0.99865 on solving for $t = \frac{8.37743}{b}$ = t_U for Θ = 3

Similarly we can obtain t_L, t_C on equating F(t) to 0.00135, 0.5

If the point above the $m(t_U)$ (5.2)(UCL) is an alarm signal. A point below the $m(t_L)$(LCL) is an indication of better quality of software. A point within the control limits indicates stable process.

6. **Failures Chart development**:

The values of m(t) at T_c, T_u, T_L for n inter-failure data are computed. m(t)’s Consecutive differences are calculated, which leads to n-1 arguments. A graph with times of inter-failure 1 to n-1 on X-axis, Consecutive differences of m(t)’s constitute n-1 values of on Y-axis, and $m(T_L)$, $m(T_U)$ and $m(T_C)$ are 3 lines parallel to X-axis respectively constitutes failures control chart to assess the software failure phenomena on the basis of the given inter-failures time data.

7. **Illustration**

Failures control chart for failure software process in a product is illustrated with an example here. The time between failures of software product are presented in Table 1[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 Constants of the model are estimated and their control limits of 4 and 5 order

Data set	Θ	Order	a	b	$m(t_U)$	$m(t_C)$	$m(t_L)$
Table 1	2	4	2.61835	0.000659	2.61476	1.10476	0.00312
		5	2.19564	0.000112	2.19873	0.93225	0.00235
	3	4	2.97656	0.000865	2.97326	1.5249	0.00763
		5	2.71321	0.000314	2.71519	1.42143	0.00467

Table 3 Consecutive differences of 4 order m (t)'s and 5 order m (t)'s for Θ = 2

Θ = 2

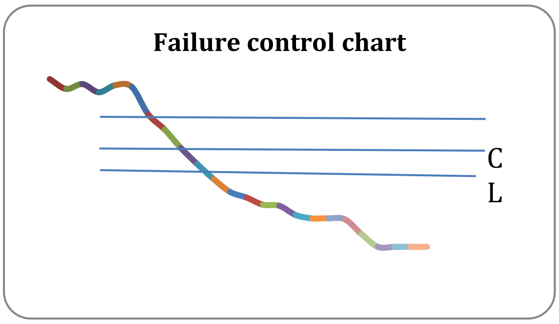
Fault	4 – Order cumulatives	m(t)s	Consecutive differences of m(t)s	5 – Order cumulative	m(t)s	Consecutive differences of m(t)s
1	227	0.51272756	0.19791225	342	0.40677887	0.1817077
2	444	0.71063981	0.18628075	571	0.58848657	0.186281
3	759	0.89692056	0.192003	968	0.77476757	0.1920003
4	1056	1.08892356	0.1823744	1986	0.96676787	0.1811834
5	1986	1.27129795	0.1907618	3098	1.14795127	0.1783087
6	2676	1.46205977	0.1887625	5049	1.32625997	0.1898507
7	4434	1.65082226	0.1564658	5324	1.51611067	0.1741975
8	5089	1.80728806	0.13854326	6380	1.69030817	0.1477769
9	5389	1.94583132	0.11657861	7644	1.38308506	0.1273431
10	6380	2.06240993	0.0975146	10089	1.96542817	0.1077727
11	7447	2.15992453	0.07985436	10982	2.07321287	0.0730000
12	7922	2.23977889	0.06478913	12559	2.14622386	0.0523000
13	10258	2.30456802	0.05815766	14708	2.19851587	0.0001195
14	11175	2.36272568	0.04975623	16185	2.19862146	0.0000799
15	12559	2.41248191	0.04984625	17758	2.19873427	0.0000124
16	13486	2.46232816	0.03751286	20567	2.19871265	0.0000021
17	15277	2.49984102	0.03356894	25910	2.19871465	0.0000012
18	16358	2.53340996	0.03368957	29361	2.19871541	0.0000023
19	18287	2.56709953	0.0324586	37642	2.19871826	0.0000005
20	20567	2.59955813	0.01520729	42015	2.19871877	0.0000015
21	24127	2.61476542	0.0000013	45406	2.19872028	0.0000004
22	28460	2.61476672	0.00000106	49416	2.19872066	0.00000011
23	32408	2.61476778	0.000001	53321	2.19872175	0.00000092

24	37654	2.61476878	0.00000008	56485	2.19872267	0.00000045
25	42015	2.61476907	0.00000005	62661	2.19872312	0.00000010
26	42296	2.61476912	0.000000033	74364	2.19872416	0.00000011
27	48296	2.61476945	0.000000017	84566	2.19872534	
28	52042	2.61476962				

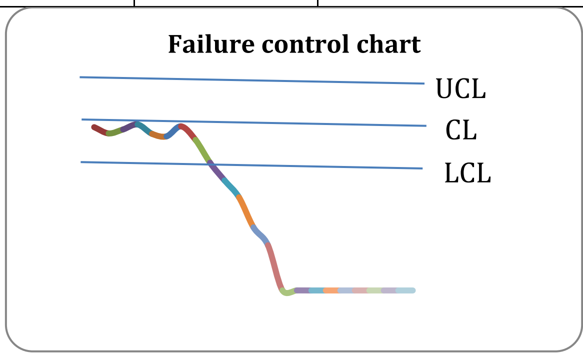
Table 4 Consecutive differences of 4 order m (t)'s and 5 order m (t)'s for ϴ = 3

ϴ = 3						
Fault	4 – Order cumulatives	m(t)s	Consecutive differences of m(t)s	5 – Order cumulatives	m(t)s	Consecutive differences of m(t)s
1	227	0.72054327	0.18670337	342	0.6247427	0.1851681
2	444	0.90724664	0.1764168	571	0.8099108	0.1864213
3	759	1.08366344	0.1830045	968	0.9963321	0.1964532
4	1056	1.26666802	0.1731854	1986	1.1927853	0.18679545
5	1986	1.43985342	0.1816729	3098	1.3758075	0.18124555
6	2676	1.63985342	0.1875246	5049	1.5608263	0.17645825
7	4434	1.80905092	0.1674859	5324	1.73728455	0.1645858953
8	5089	1.97653368	0.1276920	6380	1.9018658	0.15647896
9	5389	2.12599113	0.0987592	7644	2.06645533	0.13213451
10	6380	2.2536832	0.0816894	10089	2.22293429	0.1142356
11	7447	2.35244248	0.0789562	10982	2.355096879	0.0952346
12	7922	2.43413193	0.0691579	12559	2.46930439	0.07124151
13	10258	2.51308813	0.0597618	14708	2.56453897	0.0571247
14	11175	2.58224609	0.0897078	16185	2.6357805	0.0230476
15	12559	2.64200795	0.0394865	17758	2.6929052	0.0010472
16	13486	2.69186951	0.0345894	20567	2.7159528	0.00091324
17	15277	2.73171578	0.0378945	25910	2.717000	0.00050123

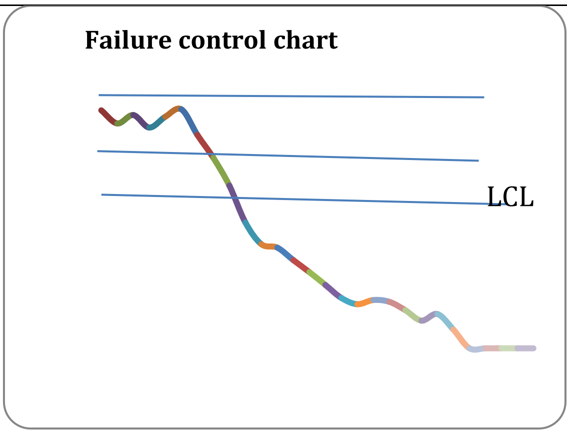
18	16358	2.77120231	0.0364589	29361	2.7171324	0.00021732
19	18287	2.80579206	0.0298657	37642	2.71841447	0.00008476
20	20567	2.84368662	0.0216847	42015	2.71863179	0.00004113
21	24127	2.88014559	0.0268745	45406	2.71871655	0.00001247
22	28460	2.91001133	0.0146543	49416	2.71875768	0.00000457
23	32408	2.93169603	0.0000021	53321	2.71877015	0.00000048
24	37654	2.93169603	0.0000006	56485	2.71877472	0.00000012
25	42015	2.95857052	0.0000015	62661	2.71877526	0.00000006
26	42296	2.97322483	0.0000019	74364	2.71877532	0.00000081
27	48296	2.97324672	0.0000031	84566	2.71877613	
28	52042	2.97324742				



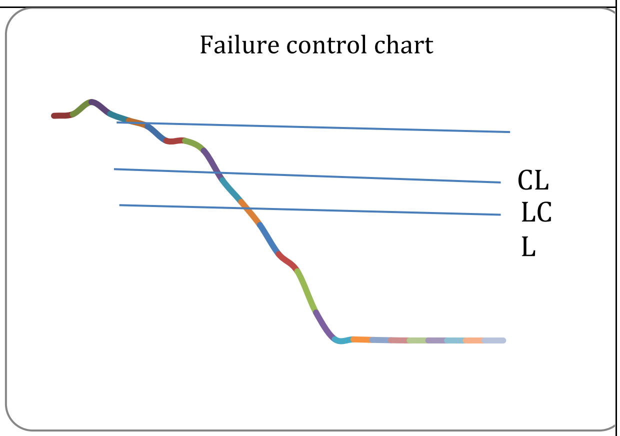
Θ = 2 Order =4, Consecutive differences of m(t) on y-axis



Θ = 2 Order = 5, Consecutive differences of m(t) on y-axis



Θ = 3 Order =4, Consecutive differences of m(t) on y-axis



Θ = 3 Order =5, Consecutive differences of m(t) on y-axis

Fig.1 Failure online process improvement methods for different θ and different order

8. CONCLUSION

The Failures Online process improvement methods of Fig.1 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 infer that our control method given in this paper with arranged sample approach suggesting a positive advice for its use to assess whether the development is in the state of control.

REFERENCES

1. [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 ;Arranged sample 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; Arranged sample and Inference; Academic Press inc.;1991.
8. V. Rama Krishna, R R L Kantam and T Subhamastan Rao “A Software Quality measurement using Generalized Half Logistic Distribution”, International Journal of Advanced Science and Technology (2020), Vol. 29, No.3, pp. 9665-9669.
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. Fan Li and Ze Long Yi, A New Software Reliability Growth Model: Multigeneration Faults and a Power-Law Testing-Effort Function, Mathematical Problems in Engineering, Vol. 2016
13. C. Jin and S.-W. Jin, “Parameter optimization of software reliability growth model with S-shaped testing-effort function using improved swarm intelligent optimization,” *Applied Soft Computing*, vol. 40, pp. 283–291, 2016.
14. DalilaAmara, Latifa BenArfa Rabai Towards a New Framework of Software Reliability Measurement Based on Software Metrics”, Volume 109, 2017, Pages 725-730
15. Lutfiah Ismail A turk1 & Wejdan Saleem Al ahmadi “Comparative Study of the Non-Homogeneous Poisson Process Type-I Generalized Half-Logistic Distribution”, International Journal of Statistics and Probability; Vol. 7, No. 6; November 2018