Software Reliability Assessment Based on Different Software Faults

Gayathry G[1], ThirumalaiSelvi R [2]

[1]Research Scholar, Bharathiar University, Coimbatore, India.
[2]Assistant Professor, Department of Computer Science, Govt. Arts College for Men, Nandanam, Chennai, India.

Abstract— Software reliability is one of the most important characteristics of software quality. Assessing the software reliability is the critical task in development of a software system. A number of Software Reliability Growth Models (SRGM) based on Non-Homogeneous Poisson Process (NHPP) have been proposed in the past decades. These models estimate reliability measures such as failure rate, number of remaining faults and software reliability. This paper develops a SRGM model based on NHPP. Model parameters are estimated using Maximum Likelihood Estimation (MLE) method. Software reliability is measured through numerical results on three software projects.

Keywords— *Non-Homogeneous Poisson Process, Failure Intensity, Parameter Estimation*

I. INTRODUCTION

A computer system is composed of hardware and software components. Extensive research has been done in both the area of hardware and software reliability. Software reliability is the probability of failure-free operation of software in a specified environment during specified duration [1]. It is a key factor in software development and software quality. Reliability of software has been studied in terms of Software Reliability Growth Models (SRGM). These models describe the relationship among the calendar testing time, amount of testing -effort required and the number of software errors detected.

SRGM is a mathematical model of how software reliability improves as faults are detected and repaired [2]. These models used during the testing phase of software development life cycle. Two types of failure data namely time-domain data, interval-domain data are taken as input. The individual time at which the failure has occurred is recorded in time-domain data. The number of failures occurs during fixed time period is recorded in interval-domain data. During the past three decades various research activities have been conducted, to assess the reliability of software. Among all SRGMs, NHPP reliability models are widely used. NHPP based SRGM's are classified into continuous time models and discrete time models [3]. Continuous time models are based on calendar time or execution time. The discrete time models use the number of test case as fault detection period. This model predicts the reliability of software by assuming that the debugging process reduces the future fault occurrence count characterized by its mean value function. A large family of statistical models have been developed for estimating reliability. Most of the existing models are purely based on the observation of software product failures where they require a considerable amount of failure data to predict the reliability accurately.

II. PARAMETER ESTIMATION

Parameter estimation is one of the most significant part in software reliability prediction. We develop expressions in order to estimate the parameters of proposed model based on time domain data. Parameter estimation is achieved by applying a well-known estimation known as Maximum Likelihood Estimation (MLE) [4]. MLE find out the parameters that maximize the probability of data. These methods are versatile and applicable to most of the reliability models and for different types of data.

The estimates of parameters 'a 'and 'b' obtained by solving the following equations [5]

$$a = \frac{\mathrm{n}}{(1 - \mathrm{e}^{-\mathrm{bs}}\mathrm{n})}$$

$$\frac{n}{b} = \sum_{i=1}^{n} \mathrm{Si} + \frac{\mathrm{nsn}}{(1 - \mathrm{e}^{-\mathrm{bs}}\mathrm{n})}\mathrm{e}^{-\mathrm{bs}}\mathrm{n}$$

III. PROPOSED NEW MODEL

In this paper, we propose a new model to estimate reliability of software. A simple random failure-based model developed to predict software reliability.

Following assumptions considered:

- Consider the failures caused by both hardware and software
- Software failures removed by Goel-Okumoto model and hardware faults removed by replacing faulty hardware.

Step 1:
Estimation of Failure Rate
Failure rate estimated by adding the failures due to software and failures due to hardware.
The Equation for Failure Rate is given in equation (1)

$$\lambda (t) = \lambda_{hw} (t) + \lambda_{sw} (t) \quad (1)$$

Substituting the Goel-Okumoto failure rate of software in the above equation (1) we get equation (2)

$$\lambda (t) = \lambda_{hw} (t) + abe^{-bt} \quad (2)$$

Step 2:
Function for mean value calculation
The mean value function written as

$$m(t) = e^{-\lambda hw (t)} + a(1-e^{-bt}) \quad (3)$$

Step 3:
Estimation of Reliability
The software reliability R(x|t) defined as probability of failure free operation of a software for a specified time interval i.e. (t, t+x) in a specified environment. The resultant equation to calculate the reliability is given in equation (4).

$$R(x|t) = e^{-(\lambda hw * x)} + e^{-a\,(e^{-bt} - e^{-b(t+x)})} \quad (4)$$

The reliability of software estimated using above said equation.

IV. DATA ANALYSIS

Three different data sets are considered to evaluate the method of performance based on mean value function. The set of software errors analyzed here is borrowed from real software development project as published in Pham (2006), Xie(2002),AT & T data set[6][7]. The data are named as Phase1, Phase2, Phase 3 test data. The data sets are summarized in the below table.

Table 1: Phase1 Test Data

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

Table2: Phase2 Test Data

Error Number	Time Between Errors (Hours)
1	30.02
2	1.44
3	22.47
4	1.36
5	3.43
6	13.2
7	5.15
8	3.83
9	21
10	12.97
11	0.47
12	6.23
13	3.39
14	9.11
15	2.18
16	15.53
17	25.72
18	2.79
19	1.92
20	4.13
21	70.47
22	17.07
23	3.99
24	176.06
25	81.07
26	2.27
27	15.63
28	120.78
29	30.81
30	34.19

Table 3: Phase 3 Test Data

Error Number	Time Between Errors
1	5.5

2	1.83
3	2.75
4	70.89
5	3.94
6	14.98
7	3.47
8	9.96
9	11.39
10	19.88
11	7.81
12	14.6
13	11.41
14	18.94
15	65.3
16	0.04
17	125.67
18	82.69
19	0.46
20	31.61
21	129.31
22	47.6

V. Distribution of Interval Domain Data Failures

The values of parameter estimates obtained by MLE method are given in table form:

Table 4: Parameters Estimated through MLE

Datasets	No. of Samples	Estimates of 'a'	Estimates of 'b'
NTDS	26	29.71851	0.008314
Xie	30	31.6982	0.00396
AT & T	22	23.38078	0.004161

The mean value and failure intensity function are calculated for the considered data sets. These values are tabulated in Table 5.1,5.2,5.3.

Table 5.1: Mean Value, Failure Intensity of Phase1 Test Data

Error Number	Cumulative Error	Mean Value	Failure Intensity
1	9	2.142558	0.229266
2	21	4.760967	0.207497
3	32	6.942174	0.189362
4	36	7.687167	0.183168
5	43	8.932751	0.172812
6	45	9.275519	0.169963
7	50	10.107912	0.163042
8	58	11.369821	0.152550
9	63	12.116940	0.146339
10	70	13.112078	0.138065
11	71	13.249571	0.136922
12	77	14.050953	0.130260
13	78	14.180673	0.129181
14	87	15.300875	0.119868
15	91	15.772463	0.115947
16	92	15.887929	0.114987
17	95	16.228625	0.112154
18	98	16.560929	0.109339
19	104	17.201180	0.104069
20	105	17.304817	0.103207
21	116	18.389733	0.094187
22	149	21.107995	0.071587
23	156	21.594807	0.067540
24	247	25.906288	0.031694
25	249	25.969154	0.031172
26	250	26.000196	0.030914

Table 5.2: Mean Value, Failure Intensity of Phase2 Test Data

Error Number	Cumulative Error	Mean Value	Failure Intensity
1	30.02	3.552891	0.111455
2	31.46	3.712930	0.110821
3	53.93	6.095518	0.101386
4	55.29	6.233033	0.100842
5	58.72	6.576583	0.099481
6	71.92	7.856010	0.094415
7	77.07	8.337323	0.092509
8	80.9	8.688959	0.091116
9	101.9	10.525007	0.083845
10	114.87	11.585032	0.079648
11	115.34	11.622432	0.079500
12	121.57	12.111658	0.077562
13	124.96	12.372838	0.076528
14	134.07	13.057586	0.073816
15	136.25	13.217815	0.073182
16	151.78	14.320094	0.068817
17	177.5	16.002922	0.062153
18	180.29	16.175375	0.061470
19	182.21	16.292950	0.061004
20	186.34	16.542851	0.060015
21	256.81	20.233275	0.045401
22	273.88	20.982659	0.042433
23	277.87	21.150638	0.041768
24	453.93	26.445737	0.020799
25	535	27.888083	0.015088
26	537.27	27.922179	0.014953
27	552.9	28.148809	0.014055
28	673.68	29.498134	0.008712

29	704.49	29.750830	0.007711
30	738.68	29.997419	0.006735

Table 5.3: Mean Value, Failure Intensity of Phase3 Test Data

Error Number	Cumulative Error	Mean Value	Failure Intensity
1	5.5	0.529004	0.095086
2	7.33	0.702351	0.094364
3	10.08	0.960375	0.093291
4	80.97	6.687635	0.069460
5	84.91	6.959077	0.068330
6	99.89	7.951422	0.064201
7	103.36	8.172601	0.063281
8	113.32	8.790000	0.060712
9	124.71	9.465381	0.057901
10	144.59	10.570149	0.053305
11	152.4	10.979770	0.051600
12	167	11.710711	0.048559
13	178.41	12.251824	0.046307
14	197.35	13.095220	0.042798
15	262.65	15.542435	0.032615
16	262.69	15.543739	0.032609
17	388.36	18.735050	0.019330
18	471.05	20.087528	0.013703
19	471.51	20.093826	0.013677
20	503.12	20.498931	0.011991
21	632.43	21.698124	0.007001
22	680.03	22.000468	0.005743

VI. RESULTS OBTAINED IN THE FORM OF GRAPH

The following control chart represents the software failure phenomena on the basis of given inter-failures time data. By placing the error number shown in Table 1,2,3, on x axis and the calculated failure intensity on y axis we obtained Figure1, Figure2 and Figure3.

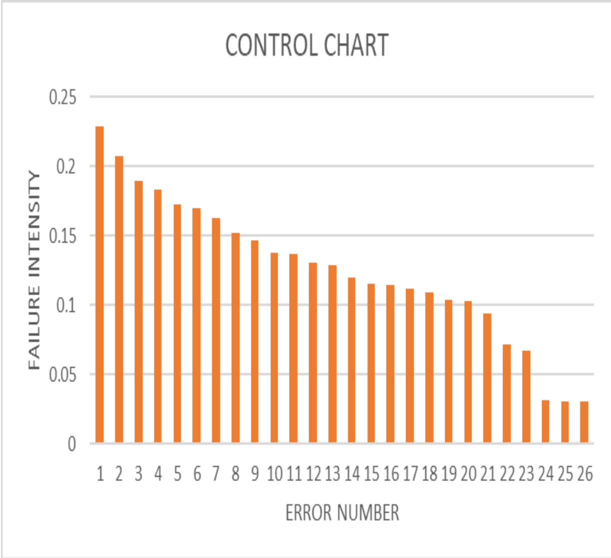

Figure1: Phase1 Test Data

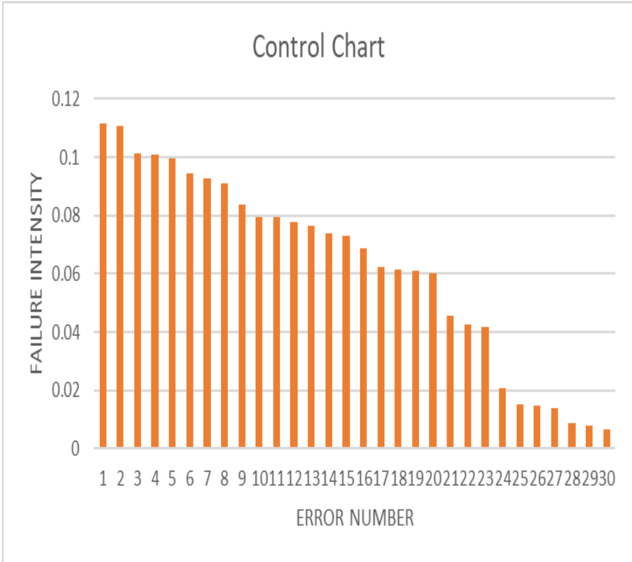

Figure2: Phase2 Test Data

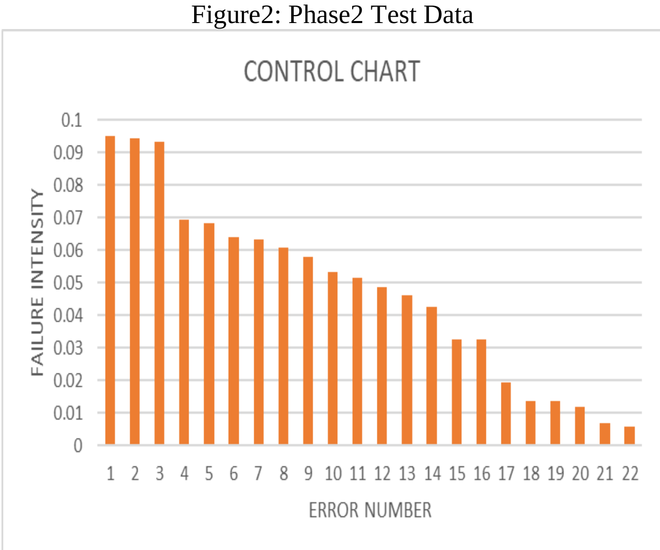

Figure3: Phase3 Test Data

From the graph, it is clear that the failure intensity is somewhat stabilized at the end of error number 22 to 30 which indicates the completion of development phase. Thus, the software is ready to release.

VII. ESTIMATION OF RELIABILITY

To Calculate the failure rate caused by hardware, the following hardware components failure rate shown in Table 3 is taken into account.

Table 3: Hardware Component Failure [8]

Hardware Component No.	Failure Rate (per month)
H1	0.027778
H2	0.025
H3	0.028571
H4	0.02381
H5	0.016667
H6	0.041667
H7	0.034483
H8	0.027778

The values of hardware failure rate $e^{-0.225753}$ substituted in equation (4). The estimator of reliability function of the software-based system is given by:

$$R(x|t) = e^{-(\lambda hw * x)} + e^{-a}(e^{-bt} - e^{-b(t+x)})$$

The reliability of Phase1 test data when a=-29.71851, b=0.008314, t=1, x=0.1 is

$R(x|t) = e^{-0.225753 * t} + e^{-29.71851} (e^{-0.008314*t} - e^{-0.008314*(t+x)})$

then R(x|t) =0. 9908

The reliability of Phase2 test data when a=-31.6982, b=0.00396, t=1, x=0.1 is

$R(x|t) = e^{-0.225753 * t} + e^{-31.6982} (e^{-0.00396*t} - e^{-0.00396*(t+x)})$

then R(x|t) =0.8021

The reliability of Phase3 test data when a=-23.38078, b=0.004161, t=1, x=0.1 is

$R(x|t) = e^{-0.225753 * t} + e^{-23.38078} (e^{-0.004161*t} - e^{-0.004161*(t+x)})$

then R(x|t) =0.7979

VIII. CONCLUSION

The early detection of software failure will improve the quality of software. The analysis based on reliability shows that Phase1 Test Data is more reliable than the other two data set. The methodology adopted in this paper is a simple method for model validation. It is the best choice for an early detection of software failures. The model is very convenient for practitioners of software reliability.

REFERENCES

[1] Alan Wood, “Software Reliability Growth Models”, In 1996.

[2] D.Swamydoss, Dr.Kadhar Nawaz “Enhanced Version of Growth Model in Web Based Software Reliability Engineering” in JGRCS Vol.2,No.12,Dec.2011.

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

[4] R. Satya Prasad, K. R. H. Rao and R. R. L Kantha, “Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC”, International Journal of Computer