

MODIFIED GOEL - OKUMOTO MODEL FOR ESTIMATING SOFTWARE RELIABILITY**Dr. Blessy Thankachan¹, Mr. Sachin Jain²**

Abstract: Software Reliability Growth Model (SRGM) is one of the fundamental techniques used to assess reliability of software quantitatively. Software reliability growth model is required to have a good performance in terms of goodness of fit, predictability, and so forth. A number of models have been proposed during the past decades for reliability analysis of a software system. But many of them are insufficient to correctly analyze the exact reliability of the software. Moreover the calculation of the time period up to which the testing has to be done before the release of software for public use is not possible to predict through these models. Therefore in this chapter we have presented a Modified Goel-Okumoto Model through which a threshold value is calculated which will provide the software development team with the minimum time period up to which the testing has to be done and after that the software can be released for public use. But the applicability of SRGM highly depends upon the correct estimation of parameters. Estimation of model parameters is usually different according to different estimation methods. In this paper we have also will summarized about the parameters and the significance of these parameters on the predictability of reliability by the Software Reliability Growth Models (SRGMs).

Keywords: - Goel-Okumoto Model, Software Reliability Growth Models, Roundness factor.

Introduction: Any software reliability growth model (SRGM) can be considered to be a mathematical expression which fits the experimental data. It may be obtained by simply observing the overall trend of reliability growth. But some of the models can also be obtained analytically by making some assumptions about the software testing and debugging. Some of the assumptions are to keep the analysis tractable; other can be more fundamental in nature and can constitute modeling of the testing and debugging process itself. Goel discussed the applicability and limitations of software reliability growth models during the software development life cycle in [2]. The parameters of the model also have an important role in the interpretation of the testing process and the reliability of the software. A clear understanding of the underlying parameters gives us a valuable insight into the process [4].

1. If we know how a parameter arises, it can be estimated even before testing begins. Such a priori values when are estimated using past experience, it can be used to do the preliminary planning and resource allocation before testing begins[5].
2. The use of SRGMs suggests that in the beginning of testing, the initial test data yields unstable parameter values. In such cases, values can be estimated using static information, which can serve as a check. These parameter values can also be used to stabilize the projections adding to the information obtained by the dynamic defect detection data.
3. Many a times iterative techniques can be used to estimate the parameter values. The values obtained depend on the initial estimates that are required for numerical computation. Use of a priori values as the initial estimate would initiate the search and lead closer to the values sought.
4. Parameters that have an interpretation symbolize the testing and debugging process quantitatively. Their values can give us an understanding of the process. They may help at reveal about how the inherent defect density can be reduced or how testing can be made more efficient.

*Corresponding author

¹Jaipur National University, Jaipur-Agra Bypass, Near New RTO Office, Jagatpura, Jaipur-302017, India

E-mail: blessyt218@gmail.com

Published on Web 30/12/2021, www.ijsonline.org

This chapter examines the parameters of the exponential models. A new interpretation of the parameters of the exponential model is presented.

Implication of Modified Goel-Okumoto Model

All software reliability growth models are approximations of real processes, thus none can be considered perfect, but Goel-Okumoto model (Exponential) is found to be applicable for a variety of software. Therefore this model has been chosen for further study.

The Goel-Okumoto Model predicts the predicted error with the help of roundness factor and the time of error reporting. But it is insufficient to further analyze the reliability of the software. Therefore, modification in the model has been done which includes a new variable, the threshold value (α). The threshold value (α) gives the minimum time after which the software is considered to be reliable or the maximum time up to which the software is highly vulnerable and is not ready for release for public use. The **modified Goel-Okumoto Model** is presented as follows:

$$\mu(t) = E_E(1 - e^{-bt}) \quad (\text{Original Goel-Okumoto Model})$$

$$\mu'(t) = E^E(1 - e^{-b'(t-\alpha)}) \quad \text{where } b' > 0$$

(Modified Goel-Okumoto Model) (1)

$\mu'(t)$ = Mean value function or the predicted number of bugs at time t

E_E = Expected total number of defects in the code

b' = **Mean value (Roundness factor)**

t = Calendar time/ execution time/ number of test runs

α = **Threshold value**

Further the reliability of the software can be estimated by the Reliability Factor (RF). Further in this study reliability factor at two instances is done for comparison. First at the threshold time and at final testing time. It is found out that the reliability at the two instances is not varying to a larger extent. Which indicates that the after the threshold time the software is approaching reliability and is ready for public dispatch.

Parameter estimation is a discipline that provides methods for efficient use of data for aiding in mathematical modeling of phenomenon and estimations of constants appearing in these models [1]. Most of the parameter estimation can be related to four optimization problems [6]:

- Criterion: the choice of the best function to optimize (maximize or minimize)
- Estimation: the choice of the best function to optimize the chosen function
- Design: Optimal implementation of the chosen method to obtain the best parameters estimates.

- Modeling: The determination of the mathematical model which best describes the system from which data are measured, including the model of the error processes.

Further, the reliability of the software can also be estimated by the alternate quantities of the failure data. Typically, software reliability predictions take into account factors such as the size and complexity of a program. These predictions can be used to analyze the reliability of the software.

Significance of the Parameters in Modified Goel-Okumoto Model

Here we derived a method to evolve the parameters (E_E , b' and α) of this model.

Total Expected Errors (E_E)

It has been observed that for a specific development environment and for the same software development team and for the same development/testing period, the error density encountered is same [4]. E_E is the number of errors which is expected to be in the software when the testing period starts. As the testing continues errors are detected and debugged. In this model it is considered that as errors are debugged new errors are not introduced. E_E is computed on various factors which can affect the software development i.e the quantity which has to be assumed is influenced by several independent factor. Some of the factors which are discussed below are:

- Phase factors
- Programming Team Factor
- Maturity factor
- Structure Factor
- Requirements volatility factor

Mean value (Roundness factor) (b'): Roundness is a measure of the compactness of a shape. A circle is the most compact shape, so the more compact a shape is, the more closely it resembles a circle. Roundness is a ratio and therefore a dimensionless number. The roundness factor may be calculated for any two dimensional shape as long its area and perimeter is known.

The roundness factor (b') in SRGM depict the rate at which the error is decreasing in the software while testing or the roundness factor (b') represent the failure rate of the software. The **modified Goel - Okumoto** model predicts the number of errors which the software might encounter. As the testing proceeds the errors goes on decreasing (it is assumed that the errors are rectified as they are encountered).

For Exponential Model (increasing form) of equation 1 it posses the following features:

- It is asymptotic in nature i.e $y = a$ to right
- The curve passes through the coordinates (0,0)

- a is the upper limit
 - The value will Increase, but bounded above by y=a
- Exponential decay models of this form can model sales or learning curves where there is an upper limit. This is done by subtracting the exponential expression from one and multiplying by the upper limit. Exponential decay models of this form will increase very rapidly at first, and then level off to become asymptotic to the upper limit.

Threshold Value (α): The threshold value represents that time period which is considered to be the maximum time after which the software is considered to be reaching its maximum reliability. Up to the threshold time the maximum errors would have been found out and debugged (It is assumed that as no new errors are introduced while the errors are debugged). After the threshold time the software can be considered to be the main version.

Computing the values of b and α for Modified Goel-Okumoto Model: Computing the value of b and α, from equation 1

$$\mu(t) = E_E (1 - e^{-b(t-\alpha)}), \text{ where } b > 0 \text{ therefore,}$$

$$\frac{\mu(t)}{E_E} = (1 - e^{-b(t-\alpha)})$$

$$e^{-b(t-\alpha)} = 1 - \frac{\mu(t)}{E_E}$$

Taking natural log (ln) on both sides, we have

$$\ln(e^{-b(t-\alpha)}) = \ln(1 - \frac{\mu(t)}{E_E})$$

$$-b(t-\alpha) = \ln(1 - \frac{\mu(t)}{E_E}) \quad \text{because } \ln(e^x) = x$$

$$u = b\alpha - b't \quad (4) \quad (u = -\ln(1 - \frac{\mu(t)}{E_E}))$$

$$\sum u = n\alpha b' - b'\sum t \quad (5)$$

$$\sum t u = \alpha b' \sum t - b'\sum t^2 \quad (6)$$

Solving both equations (5) and (6)

$$b' = \frac{\sum u \sum t - n \sum t u}{n \sum t - (\sum t)^2} \quad (7)$$

From equation 4

$$\alpha = (u + b't) / b' \quad (8)$$

Reliability Factor

$$\text{Reliability Factor (RF)} = 1 - (E_R / E_E) \quad [3] \quad (9)$$

E_E = Total Expected Errors

E_E is the total number of errors which is expected to be in the software when the testing period starts. E_E is estimated depending upon various factors and has been discussed above.

$E_R = E_E$ – Actual Error detected and rectified = Residual Errors

It represents the errors which are considered to be in the software even after testing is completed with reference to the Total Expected Error.

Since E_E is the maximum error which is expected in the software prior to testing. And since the residual error (E_R) is the difference between the Total Expected Error (E_E) and Actual Errors detected it can be utmost be equal to E_E or less than E_E . On the basis of this idea the ratio, E_R / E_E will be less and $1 - (E_R / E_E)$ will be less than 1. This ratio gives the idea of the validity of the model and therefore $1 - (E_R / E_E)$ can be considered as Reliability Factor (RF) on the basis of modified Goel-Okumoto Model.

Thus Reliability Factor (RF) is the measure for software reliability [3]. Its value will vary between 0 and 1. If RF=1, then the software under consideration is perfect, however, if RF=0, then the software is highly vulnerable. When RF approaches close to 1 then the software can be considered as reliable.

Case Study

Debain Operating System: To further analyze the behavior of b and α data set has been taken from the bug reports by Debain [8] users, which are hoisted at the website of Debain Operating System. Debain is a Computer operating system composed of software packages released as free and open source software primarily under the GNU General Public License along with other free software license [9]. The data set consists of the error reported by different users working on multicore architecture for 109 months [Appendix A].

$$b' = \frac{\sum u \sum t - n \sum t u}{n \sum t - (\sum t)^2} \quad (\text{from 7})$$

$$n = 109$$

$$\sum tu = -3724.84 \quad (\text{From Table 3})$$

$$\sum u = -51.0252 \quad (\text{From Table 3})$$

$$\sum t = 5995 \quad (\text{From Table 3})$$

$$\sum t^2 = 437635$$

$$(\sum t)^2 = 35940025$$

$$b' = \frac{-51.0252 * 5995 - (109 * (-3724.84))}{109 * 437635 - (5995)^2}$$

$$b' = \frac{711903.634}{11762190}$$

$$b' = 0.060524$$

$$\alpha = \frac{-0.92773 + 0.060524 * 109}{0.060524} = \frac{5.669386}{0.060524}$$

$$\alpha = 93.671$$

$$\alpha = 94 \text{ months}$$

The predicted errors at 94 months are $151.472 = 151$
(From Table 3)

And the actual errors found out and rectified is 171

$$\text{Total Expected Error } (E_E) = 180$$

$$\text{Residual Error } (E_R) = 180 - 151 = 29$$

$$\text{Reliability Factor (RF)} = 1 - 29/180 = .83$$

Table 1: Comparison of RF at Threshold time and after completing the testing time

Time (in Months)	Reliability Factor (RF)
94 Months	.83
109 Months	.88

Mozilla Thunderbird: Mozilla Thunderbird is a free [8], open source, cross-platform email and news client developed by the Mozilla Foundation. The project strategy is modeled after Mozilla Firefox, a project aimed at creating a web browser. On December 7, 2004, version 1.0 was released, and received over 500,000 downloads in its first three days of release, and 1,000,000 in 10 days [9] [10].

The bug report available on the official website of Apache OOo Bugzilla has been traced to find the bug report generated by the user [7]

Calculating the values of b and α

$$b' = \frac{\sum u \sum t - n \sum tu}{n \sum t^2 - (\sum t)^2} \quad \text{(from 7)}$$

$$n = 108$$

$$\sum tu = -4943.69 \quad \text{(From Table 4)}$$

$$\sum u = -62.0767 \quad \text{(From Table 4)}$$

$$\sum t = 5886 \quad \text{(From Table 4)}$$

$$\sum t^2 = 425754$$

$$(\sum t)^2 = 34644996$$

$$b' = \frac{-62.0767 * 5886 - (108 * -4943.69)}{108 * 425754 - (5886)^2}$$

$$b' = \frac{899301.52}{11336436}$$

$$b' = 0.07932$$

$$\alpha = \frac{-1.65321 + 0.07932842 * 108}{0.07932842} = \frac{6.91425936}{0.07932842}$$

$$\alpha = 87.15$$

$$\alpha = 87 \text{ months}$$

The predicted errors at 87 months = 217 (From Table 4)

And the actual errors found out and rectified is 206

$$\text{Total Expected Error } (E_E) = 225$$

$$\text{Residual Error } (E_R) = 225 - 217 = 8$$

$$\text{Reliability Factor (RF)} = 1 - 8/225 = 0.9644$$

Table 2: Comparison of RF at Threshold time and after completing the testing time

Time (in Months)	Reliability Factor (RF)
87 Months	.96
108 Months	.98

In both the examples it is clearly seen that more than 90 percent of the errors has been detected by the threshold time (α). Therefore during the testing period after reaching the threshold time (α) the software is attaining reliability and further testing will enhance the reliability of the software to a greater level. Further the reliability of the software can be calculated by the Reliability Factor.

Conclusion: Thus the Reliability Factor (RF) of the Software Reliability Growth Model (SRGM) clearly depicts the authenticity or the reliability of the software under consideration. The value b' will always lie between 0 and 1. If the value of RF approaches 0 then the Modified G-O Model predicts that the number of residual errors is high and the reliability of the software is highly vulnerable. But at the end of the testing period, as the errors are being detected and rectified, the number of residual errors decreases with respect of total Expected Errors. As the residual errors decreases the reliability of the software goes on increasing. And it is being clearly depicted by the RF.

Appendix A

$\mu(t) = E_E (1 - e^{-bt})$ (From Original Goel-Okumoto Model) therefore,

$$\frac{\mu(t)}{E_E} = (1 - e^{-bt})$$

$$e^{-bt} = 1 - \frac{\mu(t)}{E_E}$$

Taking natural log (ln) on both sides, we have

$$\ln(e^{-bt}) = \ln(1 - \frac{\mu(t)}{E_E})$$

$$-bt = \ln(1 - \frac{\mu(t)}{E_E}) \quad \text{(because } \ln(e^x) = x)$$

and

$$b = \frac{-\ln(1 - \frac{\mu(t)}{E_E})}{t}$$

Table 3: Reliability calculation of Debian Operating System

Time (t) (in months)	Actual Errors Reported	G-O Model			u	t ²	tu
		Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors (Rounded Off)	Residual Errors			
1	3	3.493291685	3	177	-0.008511	1	-0.008511
2	5	6.918788443	7	173	-0.017023	4	-0.034045
3	6	10.27780598	10	170	-0.025534	9	-0.076602
4	6	13.57163448	14	166	-0.034045	16	-0.136181
5	10	16.80153906	17	163	-0.042556	25	-0.212782
6	12	19.96876032	20	160	-0.051068	36	-0.306406
7	18	23.07451476	23	157	-0.059579	49	-0.417053
8	24	26.11999528	26	154	-0.06809	64	-0.544723
9	27	29.10637161	29	151	-0.076602	81	-0.689414
10	27	32.03479082	32	148	-0.085113	100	-0.851129
11	31	34.90637768	35	145	-0.093624	121	-1.029866
12	36	37.72223515	38	142	-0.102135	144	-1.225626
13	40	40.48344478	40	140	-0.110647	169	-1.438408
14	41	43.19106712	43	137	-0.119158	196	-1.668213
15	46	45.84614217	46	134	-0.127669	225	-1.91504
16	48	48.4496897	48	132	-0.136181	256	-2.17889
17	49	51.00270973	51	129	-0.144692	289	-2.459763
18	52	53.50618285	54	126	-0.153203	324	-2.757658
19	56	55.96107062	56	124	-0.161714	361	-3.072575
20	57	58.36831596	58	122	-0.170226	400	-3.404516
21	60	60.72884346	61	119	-0.178737	441	-3.753479
22	61	63.04355979	63	117	-0.187248	484	-4.119464
23	62	65.31335402	65	115	-0.19576	529	-4.502472
24	65	67.53909794	68	112	-0.204271	576	-4.902503
25	67	69.72164647	70	110	-0.212782	625	-5.319556
26	70	71.86183788	72	108	-0.221294	676	-5.753632
27	73	73.96049423	74	106	-0.229805	729	-6.20473
28	78	76.01842159	76	104	-0.238316	784	-6.672851
29	78	78.03641038	78	102	-0.246827	841	-7.157994
30	80	80.01523571	80	100	-0.255339	900	-7.66016
31	83	81.95565763	82	98	-0.26385	961	-8.179349
32	84	83.85842145	84	96	-0.272361	1024	-8.71556
33	86	85.72425798	86	94	-0.280873	1089	-9.268794
34	86	87.5538839	88	92	-0.289384	1156	-9.83905

Time (t) (in months)	Actual Errors Reported	G-O Model			u	t ²	tu
		Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors (Rounded Off)	Residual Errors			
35	87	89.34800195	89	91	-0.297895	1225	-10.42633
36	90	91.10730124	91	89	-0.306406	1296	-11.03063
37	92	92.83245749	93	87	-0.314918	1369	-11.65196
38	94	94.52413333	95	85	-0.323429	1444	-12.2903
39	94	96.18297852	96	84	-0.33194	1521	-12.94567
40	94	97.80963021	98	82	-0.340452	1600	-13.61806
41	94	99.40471318	99	81	-0.348963	1681	-14.30748
42	99	100.9688401	101	79	-0.357474	1764	-15.01391
43	102	102.5026117	103	77	-0.365985	1849	-15.73737
44	103	104.0066172	104	76	-0.374497	1936	-16.47786
45	105	105.4814341	105	75	-0.383008	2025	-17.23536
46	107	106.9276291	107	73	-0.391519	2116	-18.00989
47	112	108.3457574	108	72	-0.400031	2209	-18.80144
48	119	109.7363639	110	70	-0.408542	2304	-19.61001
49	125	111.0999827	111	69	-0.417053	2401	-20.43561
50	127	112.4371375	112	68	-0.425564	2500	-21.27822
51	128	113.7483418	114	66	-0.434076	2601	-22.13786
52	131	115.0340994	115	65	-0.442587	2704	-23.01453
53	131	116.2949041	116	64	-0.451098	2809	-23.90821
54	132	117.5312401	118	62	-0.45961	2916	-24.81892
55	133	118.7435823	119	61	-0.468121	3025	-25.74665
56	137	119.9323964	120	60	-0.476632	3136	-26.6914
57	140	121.0981389	121	59	-0.485143	3249	-27.65318
58	140	122.2412577	122	58	-0.493655	3364	-28.63198
59	143	123.3621918	123	57	-0.502166	3481	-29.6278
60	147	124.4613717	124	56	-0.510677	3600	-30.64064
61	147	125.5392197	126	54	-0.519189	3721	-31.67051
62	153	126.5961496	127	53	-0.5277	3844	-32.7174
63	153	127.6325675	128	52	-0.536211	3969	-33.78131
64	153	128.6488715	129	51	-0.544723	4096	-34.86224
65	155	129.6454519	130	50	-0.553234	4225	-35.9602
66	155	130.6226915	131	49	-0.561745	4356	-37.07518
67	159	131.5809656	132	48	-0.570256	4489	-38.20718
68	163	132.5206423	133	47	-0.578768	4624	-39.3562
69	164	133.4420826	133	47	-0.587279	4761	-40.52225
70	167	134.3456403	134	46	-0.59579	4900	-41.70532

Time (t) (in months)	Actual Errors Reported	G-O Model			u	t ²	tu
		Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors (Rounded Off)	Residual Errors			
71	170	135.2316625	135	45	-0.604302	5041	-42.90541
72	171	136.1004895	136	44	-0.612813	5184	-44.12252
73	171	136.952455	137	43	-0.621324	5329	-45.35666
74	171	137.7878863	138	42	-0.629835	5476	-46.60782
75	171	138.6071042	139	41	-0.638347	5625	-47.876
76	171	139.4104234	139	41	-0.646858	5776	-49.16121
77	171	140.1981525	140	40	-0.655369	5929	-50.46343
78	171	140.9705939	141	39	-0.663881	6084	-51.78268
79	171	141.7280445	142	38	-0.672392	6241	-53.11896
80	171	142.4707951	142	38	-0.680903	6400	-54.47225
81	171	143.199131	143	37	-0.689414	6561	-55.84257
82	171	143.9133319	144	36	-0.697926	6724	-57.22991
83	171	144.6136722	145	35	-0.706437	6889	-58.63427
84	171	145.3004209	145	35	-0.714948	7056	-60.05566
85	171	145.9738418	146	34	-0.72346	7225	-61.49406
86	171	146.6341934	147	33	-0.731971	7396	-62.9495
87	171	147.2817295	147	33	-0.740482	7569	-64.42195
88	171	147.9166987	148	32	-0.748993	7744	-65.91142
89	171	148.539345	149	31	-0.757505	7921	-67.41792
90	171	149.1499075	149	31	-0.766016	8100	-68.94144
91	171	149.7486206	150	30	-0.774527	8281	-70.48199
92	171	150.3357145	150	30	-0.783039	8464	-72.03955
93	171	150.9114145	151	29	-0.79155	8649	-73.61414
94	171	151.4759418	151	29	-0.800061	8836	-75.20575
95	171	152.0295132	152	28	-0.808572	9025	-76.81439
96	171	152.5723414	153	27	-0.817084	9216	-78.44004
97	171	153.1046348	153	27	-0.825595	9409	-80.08272
98	171	153.6265979	154	26	-0.834106	9604	-81.74242
99	171	154.1384311	154	26	-0.842618	9801	-83.41915
100	171	154.6403311	155	25	-0.851129	10000	-85.11289
101	171	155.1324907	155	25	-0.85964	10201	-86.82366
102	171	155.6150988	156	24	-0.868152	10404	-88.55145
103	171	156.0883409	156	24	-0.876663	10609	-90.29627
104	171	156.5523987	157	23	-0.885174	10816	-92.0581
105	171	157.0074504	157	23	-0.893685	11025	-93.83696
106	171	157.4536709	157	23	-0.902197	11236	-95.63285

Time (t) (in months)	Actual Errors Reported	G-O Model			u	t ²	tu
		Predicted Errors μ(t) (G-O Model) μ(t)=a (1-e ^{-bt})	Predicted Errors (Rounded Off)	Residual Errors			
107	171	157.8912314	158	22	-0.910708	11449	-97.44575
108	171	158.3203002	158	22	-0.919219	11664	-99.27568
109	172	158.7410419	159	21	-0.927731	11881	-101.1226
5995					-51.025	437635	-3724.8

Table 4: Reliability calculation of Thunderbird

Time (t) (in months)	Actual Errors Reported (E _A (t))	Predicted Errors μ(t) (G-O Model) μ(t)=a (1-e ^{-bt})	Predicted Errors μ(t) (Rounded Off)	Residual Errors	u	t ²	tu
1	1	8.388735	8	217	-0.00193	1	-0.00193
2	2	16.46471	16	209	-0.00388	4	-0.00776
3	3	24.23959	24	201	-0.00583	9	-0.01749
4	4	31.72459	32	193	-0.00779	16	-0.03116
5	8	38.93053	39	186	-0.01572	25	-0.07861
6	10	45.86781	46	179	-0.01974	36	-0.11846
7	16	52.54644	53	172	-0.03204	49	-0.22425
8	19	58.97607	59	166	-0.03832	64	-0.30652
9	20	65.16599	65	160	-0.04043	81	-0.36386
10	25	71.12512	71	154	-0.05115	100	-0.51153
11	26	76.86208	77	148	-0.05333	121	-0.58662
12	30	82.38514	82	143	-0.06215	144	-0.74577
13	34	87.70229	88	137	-0.07115	169	-0.92494
14	42	92.8212	93	132	-0.08973	196	-1.25624
15	46	97.74926	98	127	-0.09933	225	-1.48994
16	49	102.4936	102	123	-0.10667	256	-1.70672
17	58	107.061	107	118	-0.12947	289	-2.20092
18	59	111.4582	111	114	-0.13207	324	-2.37734
19	62	115.6914	116	109	-0.13999	361	-2.6599
20	65	119.7668	120	105	-0.14806	400	-2.96125
21	68	123.6902	124	101	-0.15628	441	-3.28194
22	70	127.4674	127	98	-0.16185	484	-3.56072
23	72	131.1037	131	94	-0.16749	529	-3.8523
24	73	134.6045	135	90	-0.17034	576	-4.08813
25	76	137.9747	138	87	-0.179	625	-4.47491
26	79	141.2193	141	84	-0.18783	676	-4.88357
27	81	144.3429	144	81	-0.19382	729	-5.23314
28	84	147.3501	147	78	-0.20296	784	-5.68298

Time (t) (in months)	Actual Errors Reported ($E_A(t)$)	Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors $\mu(t)$ (Rounded Off)	Residual Errors	u	t^2	tu
29	87	150.2451	150	75	-0.2123	841	-6.1568
30	88	153.0322	153	72	-0.21546	900	-6.46386
31	89	155.7154	156	69	-0.21864	961	-6.77795
32	92	158.2986	158	67	-0.22833	1024	-7.30659
33	93	160.7854	161	64	-0.23161	1089	-7.64308
34	99	163.1795	163	62	-0.25181	1156	-8.56161
35	101	165.4844	165	60	-0.25876	1225	-9.05663
36	103	167.7033	168	57	-0.26582	1296	-9.56962
37	105	169.8396	170	55	-0.273	1369	-10.101
38	106	171.8961	172	53	-0.27664	1444	-10.5122
39	107	173.876	174	51	-0.2803	1521	-10.9317
40	108	175.7821	176	49	-0.284	1600	-11.3599
41	110	177.6171	178	47	-0.29148	1681	-11.9509
42	113	179.3837	179	46	-0.30296	1764	-12.7245
43	115	181.0844	181	44	-0.31079	1849	-13.364
44	117	182.7217	183	42	-0.31876	1936	-14.0254
45	117	184.298	184	41	-0.31876	2025	-14.3441
46	122	185.8155	186	39	-0.33935	2116	-15.6099
47	124	187.2764	187	38	-0.34786	2209	-16.3495
48	127	188.6829	189	36	-0.36096	2304	-17.3259
49	129	190.0369	190	35	-0.36991	2401	-18.1257
50	131	191.3404	191	34	-0.37905	2500	-18.9527
51	133	192.5954	193	32	-0.38839	2601	-19.8081
52	137	193.8035	194	31	-0.4077	2704	-21.2004
53	137	194.9666	195	30	-0.4077	2809	-21.6081
54	137	196.0864	196	29	-0.4077	2916	-22.0158
55	139	197.1644	197	28	-0.41768	3025	-22.9726
56	142	198.2022	198	27	-0.4331	3136	-24.2538
57	142	199.2013	199	26	-0.4331	3249	-24.687
58	143	200.1631	200	25	-0.43837	3364	-25.4254
59	146	201.0891	201	24	-0.45456	3481	-26.8188
60	148	201.9806	202	23	-0.46569	3600	-27.9415
61	149	202.8389	203	22	-0.47137	3721	-28.7535
62	149	203.6651	204	21	-0.47137	3844	-29.2249
63	149	204.4605	204	21	-0.47137	3969	-29.6962
64	151	205.2263	205	20	-0.48295	4096	-30.9089
65	153	205.9635	206	19	-0.49485	4225	-32.1653
66	158	206.6733	207	18	-0.52611	4356	-34.7231

Time (t) (in months)	Actual Errors Reported ($E_A(t)$)	Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors $\mu(t)$ (Rounded Off)	Residual Errors	u	t^2	tu
67	159	207.3566	207	18	-0.53264	4489	-35.6868
68	160	208.0144	208	17	-0.53927	4624	-36.6703
69	162	208.6476	209	16	-0.55284	4761	-38.1461
70	162	209.2573	209	16	-0.55284	4900	-38.6989
71	164	209.8442	210	15	-0.56685	5041	-40.2465
72	167	210.4093	210	15	-0.58875	5184	-42.3903
73	169	210.9533	211	14	-0.60399	5329	-44.0916
74	171	211.477	211	14	-0.61979	5476	-45.8644
75	172	211.9812	212	13	-0.62791	5625	-47.093
76	174	212.4666	212	13	-0.64461	5776	-48.9905
77	178	212.9339	213	12	-0.68008	5929	-52.3665
78	185	213.3837	213	12	-0.75012	6084	-58.5096
79	189	213.8168	214	11	-0.79588	6241	-62.8745
80	192	214.2338	214	11	-0.83367	6400	-66.6935
81	198	214.6352	215	10	-0.92082	6561	-74.5863
82	200	215.0216	215	10	-0.95424	6724	-78.2479
83	201	215.3936	215	10	-0.97197	6889	-80.6736
84	204	215.7518	216	9	-1.02996	7056	-86.5169
85	204	216.0966	216	9	-1.02996	7225	-87.5469
86	205	216.4285	216	9	-1.05115	7396	-90.3991
87	206	216.7481	217	8	-1.07343	7569	-93.3883
88	209	217.0558	217	8	-1.14806	7744	-101.03
89	210	217.3519	217	8	-1.17609	7921	-104.672
90	210	217.6371	218	7	-1.17609	8100	-105.848
91	212	217.9116	218	7	-1.23824	8281	-112.68
92	212	218.1759	218	7	-1.23824	8464	-113.918
93	212	218.4303	218	7	-1.23824	8649	-115.156
94	214	218.6752	219	6	-1.31079	8836	-123.214
95	214	218.9111	219	6	-1.31079	9025	-124.525
96	214	219.1381	219	6	-1.31079	9216	-125.836
97	215	219.3566	219	6	-1.35218	9409	-131.162
98	216	219.567	220	5	-1.39794	9604	-136.998
99	218	219.7696	220	5	-1.50708	9801	-149.201
100	218	219.9646	220	5	-1.50708	10000	-150.708
101	219	220.1523	220	5	-1.57403	10201	-158.977
102	219	220.3331	220	5	-1.57403	10404	-160.551
103	220	220.5071	221	4	-1.65321	10609	-170.281
104	220	220.6746	221	4	-1.65321	10816	-171.934

Time (t) (in months)	Actual Errors Reported ($E_A(t)$)	Predicted Errors $\mu(t)$ (G-O Model) $\mu(t)=a(1-e^{-bt})$	Predicted Errors $\mu(t)$ (Rounded Off)	Residual Errors	u	t ²	tu
105	220	220.8358	221	4	-1.65321	11025	-173.587
106	220	220.9911	221	4	-1.65321	11236	-175.241
107	220	221.1406	221	4	-1.65321	11449	-176.894
108	220	221.2845	221	4	-1.65321	11664	-178.547
5886					-62.0767	425754	-4943.69

References

[1] K.V Beck and K. J Arnold "Parameter estimation in Engineering and Science" Wiley series in probability and mathematical statistics, J Wiley New York, 1977

[2] A.L. Goel, 1985. Software Reliability Models: Assumptions, Limitations, and Applicability," IEEE Transaction on Reliability. 11:12, 1411

[3] Pankaj Nagar and Blessy Thankachan. 2012. Application of Goel Okumoto Model in Software Reliability Measurement, Issues and Challenges in Networking, Intelligence and Computing Technologies-ICNICT.

[4] Malaiya K. Yashwant and Denton Jason, "What do the Software Reliability Growth Model Parameter Represent?" Computer Science Department, Colorado State University

[5] Malaiya K. Yashwant, "Early Characterization of the Defect Removal Process, Proc. 9th Annual Software Reliability Symposium, May 1991

[6] Zhengyou Zhang "Parameter estimation techniques: A tutorial to applications with Conic Fitting" INRIA, France 2004

[7] <https://issues.apache.org/ooo/buglist.cgi?quicksearch=fixed+bugs+as+on+11/05/2012>

[8] "Debian and Mozilla - a study in trademarks". LWN.net.<http://lwn.net/Articles/118268/>. Retrieved 18 September 2010.

[9] Thunderbird breaks half a million downloads in three days, Mozilla Weblog (2004-12-10)

[10] Thunderbird 1.0 reaches 1,000,000 downloads in just 10 days!, Mozilla Weblog (2004-12-18)