A Novel Assessment to Achieve Maximum Efficiency in Optimizing Software Failures

An SRGM with Exponential Log-Normal Distribution

Jagadeesh Medapati [1], Prof Anand Chandulal J [2], Prof Rajinikanth T V[3]
Assistant Professor, Department of CSE, GITAM Institute of Technology, Visakhapatnam, Andhra Pradesh, India[1]
Professor, Department of CSE, GITAM Institute of Technology, Visakhapatnam, Andhra Pradesh, India[2]
Professor, Department of CSE, SREENIDHI Institute of Science and Technology, Telangana, India[3]

Abstract—Software Reliability is a specialized area of software engineering which deals with the identification of failures while developing the software. Effective analysis of the reliability helps to signify the number of failures occurred during the development phase. This in turn aid in the refinement of the failures occurred during the development of software. This paper identifies a novel assessment to detect and eliminate the actual software failures efficiently. The approach fits in an exponential log normal distribution of Generalized Gamma Mixture Model (GGMM). The approach estimates two parameters using the Maximum Likelihood Estimate (MLE). Standard Evaluation metrics like Mean Square Error (MSE), Coefficient of Determination (R2), Sum of Squares (SSE), and Root Means Square Error (RMSE) were calculated. The experimentation was carried out on five benchmark datasets which interpret the considered novel technique identifies the actual failures on par with the existing models. This novel software reliability growth model which is more effectual in the identification of the failures significantly and facilitate the present software organizations in the release of software free from bugs just in time.

Keywords—Software reliability; failure rate; reviews; software cost; optimization

I. Introduction

Software reliability deals with the process of analyzing the failures obtained during the process of designing software. This methodology helps to evaluate the reliability of software grounded on the developed model and where it takes a generated failure into account and formulates a basis for the identification of reliability process. However, these methods help to underline the present methodology of the software, identifying the Mean Time To failure (MTTF), identify the Mean Absolute Error (MAE) and understand the Mean Square Error (MSE). However, no serious attempts were made to initiate software failures during the initial development phases that help in the total analysis of the system together with a procedure by minimizing the failures such that at the failure-free software can be released just in time. As the number of failures increases, the present literature formulates various strategies and presented diverse thoughts where different models have been constituted with the only objective to identify the software failures and develop strategies to refine the failures, which are entitled as review procedures in software industries with a core intention to minimize the software failures. If the number of failures increases the number of reviews to diminish failures increase substantially making it difficult for the software to release just-in-time.

This increases the overheads of the software cost indirectly. Also in the approaches being followed by the traditional methods of estimating the reliability, the developers are only concentrating on the failures generate. However there is no serious attempt in analyzing the failure notified is a true fault or failure generated due to some of the inside errors such as network fault, data transmission failure, other failures at the internal source and because of the internal failures the end output may be tinted as a failure. Neglecting this basic ideology of analyzing a true failure and an accidental failure, the present traditional systems are evaluating the efficiency of the developed software.

Also, the traditional approaches being followed by software team in estimating the failures is totally dependent on the knowledge base present in the literature i.e., the failure rate is totally based on the supervised learning approach where the assessment is carried out mainly based on the knowledge source. However, whenever a new novel software is to be designed no such knowledge respiratory will be available and as such identification of the failure together with the clear-cut distinction among the true failure and actual failure seemed to be a potentially challenging task.

The present article makes an attempt in this direction by full filling the gaps and meeting the above two objectives listed viz., discrimination of true failures and actual failure, identification of the failure in the software where no such history is available. This article also proposes an approach wherein the failure rate can be minimized and the true failure is thereby reflected. This approach is totally based on the derived mathematical model based on Exponential Logarithmic Normal Distribution (ELND). This article is structured as follows:

Section 2, Background Study precisely highlights the numerous research carried out in the area of software reliability. Section 3 of the article gives zest of the ELND approach and its necessity; the datasets considered were presented in Section 4. The methodology is illustrated in Section 5 of the article, Section 6 deals with various performance metrics were considered in order to analyze the efficiency of the developed model. In Section 7 of the article, the results derived were summarized and discussed. In the concluding Section 8, concludes the work presented in the above sections.

II. Background Study

In order to drive the developed software's towards perfection, every software company tries to adopt the policies of software reliability life cycles with the objective to develop reliable software. In general practice after the software is developed and is assumed to be clear for implementation; the testing phase is conducted generally called as the review. In these reviews, the probability of the failures can be notified. If this failure probability is high steps are to be initiated to substantially bring down the failure rate considerably before releasing the software to the market. Many models are presented in the literature by taking this issue and formulating the objectives like developing user-friendly software, developing software which is fully functional, enhances the capability and ensures maintainability. With these objectives, the software developing should be carried out to prepare failure-free software satisfying the user's requirement. Of late many models showcased in the literature presents models that fulfil the objectives of the user requirements. Some of the predominant models in this area of research that are coined initially are from [1], by proposing the initial study of software reliability and have published and presented a good number of papers to benefit the potential researchers working in this field. Markov Birth death Process is utilized in shaping the failure probability and also suggested methodologies to identify the failure rate. The falls in this regard are identified by using the binomial distribution and Weibull distribution was considered for identifying the mean value function. The research in this area is further taken into life by [5], [6] and [7]. The errors if at all exeunt are fixed and failure intensity is proportional to the number of remaining failures[5]. A pictorial view of the failure rates and has thrown an insight to identify that the failure rate may decay during different time intervals[6]. Bayesian method of approach is followed by [7] which a derivation for estimating the effect of failures on the software cost. Every failure rate can be projected as a two-class discrete time model, where the first class represents the error detection process and the second class is utilized for estimating the future error. In these works, the authors have assumed that the failure rate formulates a geometric progression [14].

The second level of research in this direction was initiated by [16] and [17] in the research carried out by the authors the estimation of the failure rates were based on measures of dispersions and are limited to the central limit theorem. Authors have also formulated models based on hyper geometric distribution to derive a model that can find the optimal number of failures from a developed software product.

A new direction for estimating the reliability was proposed by [23], where the authors have developed a model namely Gomptez distribution and this methodology is proven to be a most validating method for estimation of the failures. Research is also extended not only using the Non-Homogeneous Poisson Processes but other distributions like a family of Pareto distribution was carried out by [24], [25] and [26], where the authors have formulated new ideologies for estimating the failure rates and identify the mean time to failure. Latest studies were also published where most of the works are based on Weibull distribution, generalized Laplacian distribution, Raleigh distribution and Gaussian distribution. These models are also confined to the study of reliability basing on the error rates.

However, in spite of rigorous research in this area, most of the works presented by the earlier authors are confined to the study of the impact of failure rate and some articles tried to project the time between the failures. No serious attempt was witnessed in the literature to minimize the error rate or to discriminate the true error from the actual error. This article is framed to fulfil this objective in the most novel approach.

III. Exponential Logarithmic Normal Distribution

In order to estimate the failures, it is necessary to understand the pattern of the failures. This analysis of the pattern helps to signify the true failures and the possible non-failures. However, it is to be notified exactly. For this purpose, many models have been present in the literature [2]-[4] [8] [11]-[15] [18]-[22]. However, these models failed to attribute the analysis of the true failure as it is evident that every initial data in the failure data model assumes exponential distribution and hence the article we have considered Exponential Logarithmic Normal Distribution. The Probability Density Function (PDF) for fitting the ELND is given by

$$\mathrm{f}(p, q) = q(\mathrm{e}^{-p\mathrm{x}})\,\mathrm{If}\, x > 0; \quad (1)$$

= 0 otherwise.

Where 'x' represents a failure

Here the values of p and q are estimated using the methodology of lease square and by using the formulae

$$\sum \mu_i = np + q\sum t_i \text{ and} \quad (2)$$

$$\sum \mu_i t_i = p\sum t_i + q\sum t^2{}_i \quad (3)$$

IV. Datasets

In order to present the proposed methodology, we have considered two datasets, namely, [9] and [10] for highlighting the proposed model. The first dataset of Tandem consists of failure data executed in four releases, Release 1 to Release 4. Each of the releases consisted of the failures generated. In the second dataset considered for the experimentation namely, Brooks & Motely contain a failure data set. These datasets are considered for the presentation of the proposed model is given below.

Labels in the Table I, TW represents the Test Weeks, EH represents the Execution Hours and ND represents the No. of defects. Labels in the Table II, TW represents the Test Weeks, EH represents the Execution Hours and AD represents the No. of defects.

TABLE I. ORIGINAL FAILURES IN TANDEM DATASET

TW	Release 1		Release 2		Release 3		Release 4	
	EH	ND	EH	ND	EH	ND	EH	ND
1	519	16	384	13	162	6	254	1
2	968	24	1186	18	499	9	788	3
3	1430	27	1471	26	715	13	1054	8
4	1893	33	2236	34	1137	20	1393	9
5	2490	41	2772	40	1799	28	2216	11
6	3058	49	2967	48	2438	40	2880	16
7	3625	54	3812	61	2818	48	3593	19
8	4422	58	4880	75	3574	54	4281	25
9	5218	69	6104	84	4234	57	5180	27
10	5823	75	6634	89	4680	59	6003	29
11	6539	81	7229	95	4955	60	7621	32
12	7083	86	8072	100	5053	61	8783	32
13	7487	90	8484	104	9604	36		
14	7846	93	8847	110	10064	38		
15	8205	96	9253	112	10560	39		
16	8564	98	9712	114	11008	39		
17	8923	99	10083	117	11237	41		
18	9282	100	10174	118	11243	42		
19	9641	100	10272	120	11305	42		
20	10000	100						

TABLE II. ORIGINAL FAILURES IN BROOKS AND MOTELY DATASET

W	EH	AD
1	7.25	7
2	10.42	29
3	17.5	61
4	24.83	108
5	32.08	134
6	44.66	159
7	64.58	175
8	117.08	223
9	164.26	259
10	259.36	312
11	315.11	369
12	374.36	408
13	417.94	479
14	462.69	559
15	505.02	624
16	580.02	681
17	642.85	771
18	716.43	831
19	759.18	888
20	799.85	978
21	896.6	1024
22	985.18	1081
23	1041.93	1110
24	1121.18	1150
25	1194.68	1166
26	1260.01	1184
27	1327.84	1221
28	1444.76	1236
29	1532.84	1244
30	1610.92	1272
31	1648.84	1278
32	1689.92	1283
33	1744.42	1286
34	1807.42	1289
35	1846.92	1301

V. METHODOLOGY

The data for the experimentation of the proposed model is presented in the above section, each of these datasets is considered and for each dataset the initial estimates of the parameters of the proposed Exponential Logarithmic Normal Distribution, *p* and *q* are estimated. Using the method of Least Square Estimation and the values so obtained are presented below:

Using these estimates the analysis of the proposed model is considered.

Here the first dataset Tandem is considered containing four releases 1 to 4 is presented along with the second failure dataset considered Brooks & Motely in the above Tables I and II.

Against each of the dataset, the analysis is carried out in a phased manner wherein the first phase the true failures are estimated and the experimentation are processed to minimize the failure rate given in Tables III.

Against each of the data released, the number of the actual defects highlighted is considered and using these defects the actual failures are predicted and are presented as below:

Labels in Table IV to Table VIII, TW represent the Test Weeks, ND represents the No. of Defects, PD represents Predicted Defect, RES represents the Residual and Fault classifies whether the failure is a True failure or not.

TABLE III. ESTIMATED VALUES OF PARAMETERS P AND Q FOR THE DATASETS CONSIDERED

Datasets Considered	p	q
Tandem Release 1	135.845	0.078
Tandem Release 2	179.573	0.063
Tandem Release 3	49.339	0.237
Tandem Release 4	605.941	0.005
Brooks & Motely	11981.548	0.004

TABLE IV. ACTUAL FAILURES FOR TANDEM DATASET RELEASE-1

Observations	TW	ND	PD	RES	Fault
Failure 1	1	16	10.18	5.82	N
Failure 2	2	24	19.598	4.402	N
Failure 3	3	27	28.309	-1.309	Y
Failure 4	4	33	36.368	-3.368	Y
Failure 5	5	41	43.823	-2.823	Y
Failure 6	6	49	50.719	-1.719	Y
Failure 7	7	54	57.099	-3.099	Y
Failure 8	8	58	63	-5	Y
Failure 9	9	69	68.459	0.541	N
Failure 10	10	75	73.509	1.491	N
Failure 11	11	81	78.181	2.819	N
Failure 12	12	86	82.502	3.498	N
Failure 13	13	90	86.5	3.5	N
Failure 14	14	93	90.198	2.802	N
Failure 15	15	96	93.618	2.382	N
Failure 16	16	98	96.783	1.217	N
Failure 17	17	99	99.71	-0.71	Y
Failure 18	18	100	102.418	-2.418	Y
Failure 19	19	100	104.923	-4.923	Y
Failure 20	20	100	107.241	-7.241	Y

In this process, the residuals are identified where the actual notified errors are subtracted from the predicted errors and the process carried out on the two datasets namely Tandem and Brooks & Motely are tabulated in Table IV to Table VIII. The Fault column in every table specifies the outcome of the proposed model on the datasets and it clearly specifies how best the proposed model have identified the true failures and in turn reduce the failure rate when compared to the original dataset.

TABLE V. ACTUAL FAILURES FOR TANDEM DATASET RELEASE-2

Observations	TW	ND	PD	RES	Fault
Failure 1	1	13	11.036	1.964	N
Failure 2	2	18	21.393	-3.393	Y
Failure 3	3	26	31.114	-5.114	Y
Failure 4	4	34	40.238	-6.238	Y
Failure 5	5	40	48.801	-8.801	Y
Failure 6	6	48	56.837	-8.837	Y
Failure 7	7	61	64.38	-3.38	Y
Failure 8	8	75	71.459	3.541	N
Failure 9	9	84	78.104	5.896	N
Failure 10	10	89	84.34	4.66	N
Failure 11	11	95	90.192	4.808	N
Failure 12	12	100	95.685	4.315	N
Failure 13	13	104	100.84	3.16	N
Failure 14	14	110	105.679	4.321	N
Failure 15	15	112	110.22	1.78	N
Failure 16	16	114	114.482	-0.482	Y
Failure 17	17	117	118.482	-1.482	Y
Failure 18	18	118	122.237	-4.237	Y
Failure 19	19	120	125.76	-5.76	Y

TABLE VI. ACTUAL FAILURES FOR TANDEM DATASET RELEASE-3

Observations	TW	ND	PD	RES	Fault
Failure 1	1	6	10.397	-4.397	Y
Failure 2	2	9	18.603	-9.603	Y
Failure 3	3	13	25.08	-12.08	Y
Failure 4	4	20	30.192	-10.19	Y
Failure 5	5	28	34.227	-6.227	Y
Failure 6	6	40	37.411	2.589	N
Failure 7	7	48	39.925	8.075	N
Failure 8	8	54	41.909	12.091	N
Failure 9	9	57	43.474	13.526	N
Failure 10	10	59	44.71	14.29	N
Failure 11	11	60	45.685	14.315	N
Failure 12	12	61	46.455	14.545	N
Failure 13	13	36	47.063	-11.06	Y
Failure 14	14	38	47.542	-9.542	Y
Failure 15	15	39	47.921	-8.921	Y
Failure 16	16	39	48.22	-9.22	Y
Failure 17	17	41	48.455	-7.455	Y
Failure 18	18	42	48.641	-6.641	Y
Failure 19	19	42	48.788	-6.788	Y

TABLE VII. ACTUAL FAILURES FOR TANDEM DATASET RELEASE-4

Observations	TW	ND	PD	RES	Fault
Failure 1	1	1	2.863	-1.863	Y
Failure 2	2	3	5.713	-2.713	Y
Failure 3	3	8	8.549	-0.549	Y
Failure 4	4	9	11.372	-2.372	Y
Failure 5	5	11	14.182	-3.182	Y
Failure 6	6	16	16.978	-0.978	Y
Failure 7	7	19	19.761	-0.761	Y
Failure 8	8	25	22.531	2.469	N
Failure 9	9	27	25.288	1.712	N
Failure 10	10	29	28.031	0.969	N
Failure 11	11	32	30.762	1.238	N
Failure 12	12	32	33.48	-1.48	Y

TABLE VIII. ACTUAL FAILURES FOR BROOKS AND MOTELY DATASET

Observations	TW	ND	PD	RES	Fault
Failure 1	1	7	44.292	-37.29	Y
Failure 2	2	29	88.42	-59.42	Y
Failure 3	3	61	132.386	-71.39	Y
Failure 4	4	108	176.188	-68.19	Y
Failure 5	5	134	219.829	-85.83	Y
Failure 6	6	159	263.309	-104.3	Y
Failure 7	7	175	306.627	-131.6	Y
Failure 8	8	223	349.786	-126.8	Y
Failure 9	9	259	392.785	-133.8	Y
Failure 10	10	312	435.625	-123.6	Y
Failure 11	11	369	478.307	-109.3	Y
Failure 12	12	408	520.831	-112.8	Y
Failure 13	13	479	563.197	-84.2	Y
Failure 14	14	559	605.407	-46.41	Y
Failure 15	15	624	647.462	-23.46	Y
Failure 16	16	681	689.36	-8.36	Y
Failure 17	17	771	731.104	39.896	N
Failure 18	18	831	772.693	58.307	N
Failure 19	19	888	814.129	73.871	N
Failure 20	20	978	855.412	122.59	N
Failure 21	21	1024	896.541	127.46	N
Failure 22	22	1081	937.519	143.48	N
Failure 23	23	1110	978.346	131.65	N
Failure 24	24	1150	1019.02	130.98	N
Failure 25	25	1166	1059.55	106.45	N
Failure 26	26	1184	1099.92	84.079	N
Failure 27	27	1221	1140.15	80.853	N
Failure 28	28	1236	1180.23	55.775	N
Failure 29	29	1244	1220.15	23.846	N
Failure 30	30	1272	1259.94	12.065	N
Failure 31	31	1278	1299.57	-21.57	Y
Failure 32	32	1283	1339.06	-56.06	Y
Failure 33	33	1286	1378.4	-92.4	Y
Failure 34	34	1289	1417.6	-128.6	Y
Failure 35	35	1301	1456.65	-155.6	Y

VI. Performance Evaluation Metrics

In order to evaluate the outputs derived from the proposed model, we have considered the following metrics such as Mean Squared Error (MSE), R^2, Sum of Squares Error (SSE) and Root Mean Squared Error (RMSE). The formulas for the calculation of the above metrics are given by

Mean Squared Error

$$MSE = \frac{\sum\left(\left|Actual\ Failure_i - Estimated\ Failure_i\right|\right)^2}{n-1} \quad (4)$$

Mean Absolute Percent Error

$$MAPE = \frac{\sum \frac{\left|Actual\ Failure_i - Estimated\ Failure_i\right|}{Actual\ Failure_i} X100}{n} \quad (5)$$

Error of Sum of Squares

$$SSE = \sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2 \quad (6)$$

Coefficient of Determination

$$R^2 = 1 - \frac{SSE_{res}}{SSE_{tot}} \quad (7)$$

Root Mean Square Error

$$RMSE = \sqrt{\frac{\sum\left(\left|Actual\ Failure_i - Estimated\ Failure_i\right|\right)^2}{n-1}} \quad (8)$$

VII. Results and Discussions

The results of the performance evaluation metrics are showcased in the following Table IX.

From the above Table IX, it can be clearly seen that the MSE is less for the Release 1 and the R^2, is almost approaching 1, which signify that the model performs better.

The SSE metrics and RMSE metrics also showcase significant measures. This showcases that the proposed methodology is delivering an outstanding performance in predicting the failures. The experimentation carried out across the two datasets namely, Tandem and Brooks & Motely were represented below.The figures showcase the experimentation carried out across the datasets with respect to the individual failure dataset.

TABLE IX. Actual Failures for TANDEM Dataset Release-4

Dataset Considered	MSE	R^2	SSE	RMSE
Tandem Release 1	11.317	0.988	192.388	3.364
Tandem Release 2	26.088	0.982	417.408	5.108
Tandem Release 3	116.630	0.664	1866.081	10.800
Tandem Release 4	3.861	0.980	34.749	1.965
Brooks and Motely	9659.621	0.966	309107.871	98.283

Fig. 1, Fig. 4, Fig. 7, Fig. 10 and Fig. 13 depict the actual failures of various datasets. It can be clearly seen that for the values which lie above the curve were reported as failures but not a failure in the original. The present model is novel to identify the true failures and thus drives our attempt in novel nature.

Fig. 2, Fig. 5, Fig. 8, Fig. 11 and Fig. 14 depict the predicted failures of various datasets. The same set of failures at the respective time were even predicted.

Fig. 3, Fig. 6, Fig. 9, Fig. 12 and Fig. 15 depict the residuals evaluated for various datasets. This clearly showcases the entire methodology and the results keep it on track so that the novelty of the entire concept is justified. The failures that are identified were displayed for the datasets considered. The residuals were calculated across every observation and were presented for the datasets considered.

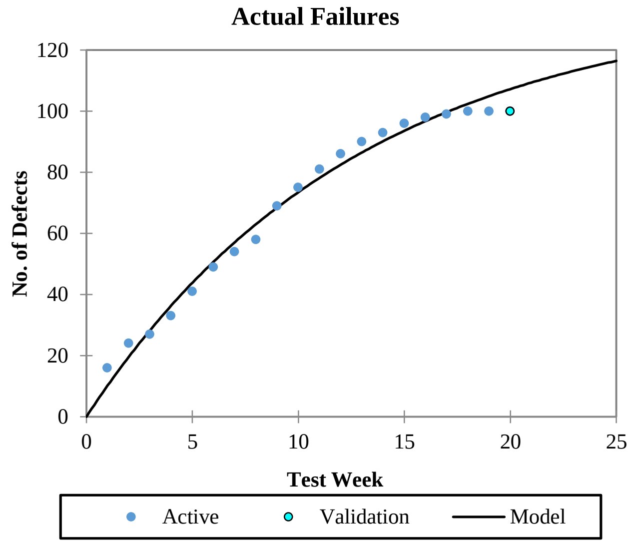

Fig. 1. Actual Failures vs No. of Defects for the TANDEM Release 1.

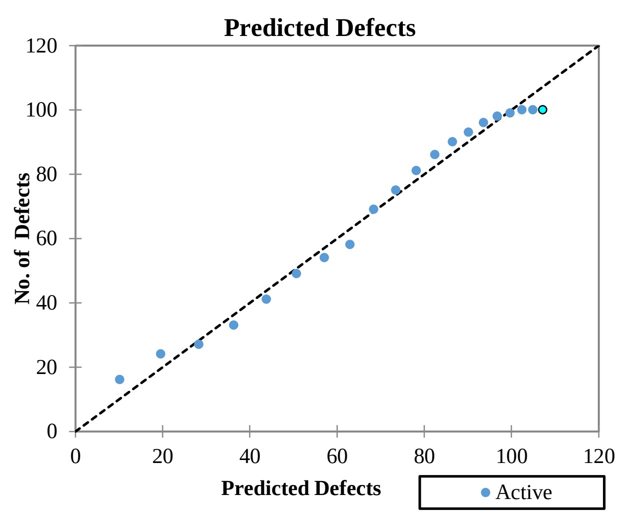

Fig. 2. Predicted Defects vs No. of Defects for the TANDEM Release 1.

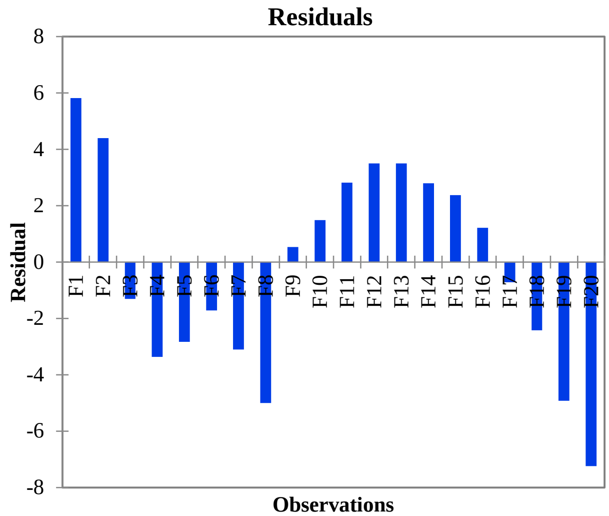

Fig. 3. Observations versus Residuals for the TANDEM Release 1.

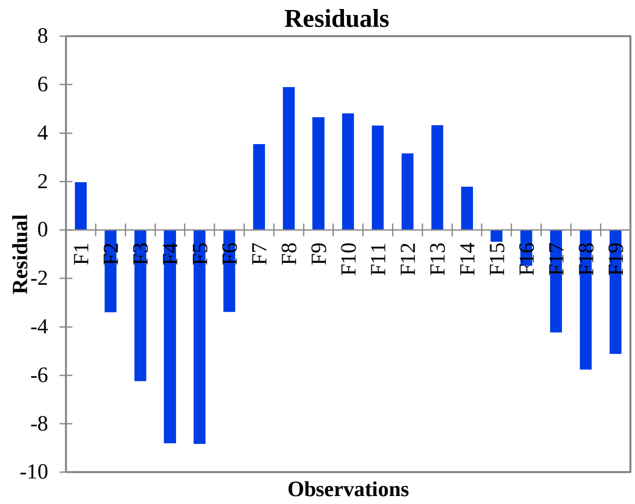

Fig. 6. Observations Versus Residuals for the TANDEM Release 2.

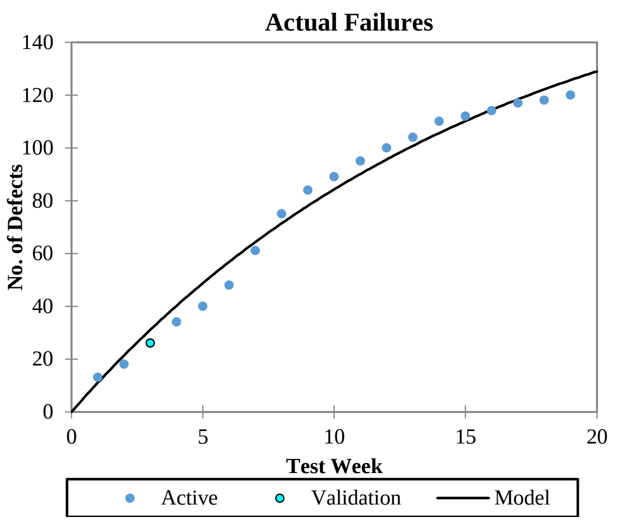

Fig. 4. Actual failures vs No. of Defects for the TANDEM Release 2.

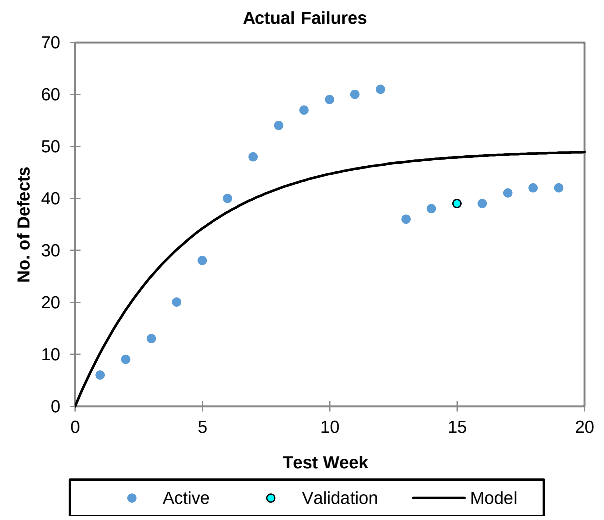

Fig. 7. Actual Failures vs No. of Defects for the TANDEM Release 3.

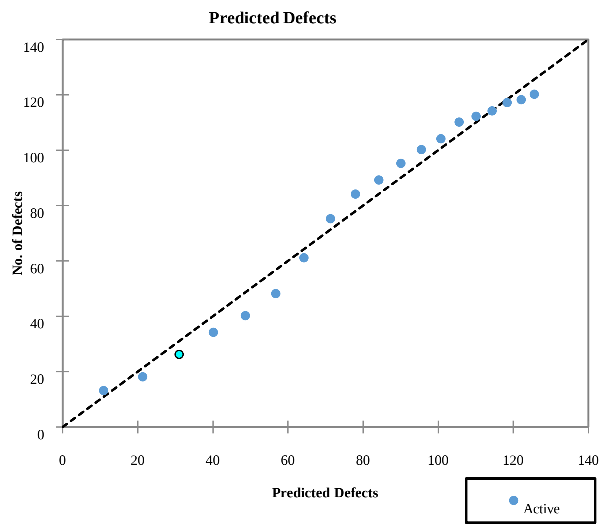

Fig. 5. Predicted Defects vs No. of Defects for the TANDEM Release 2.

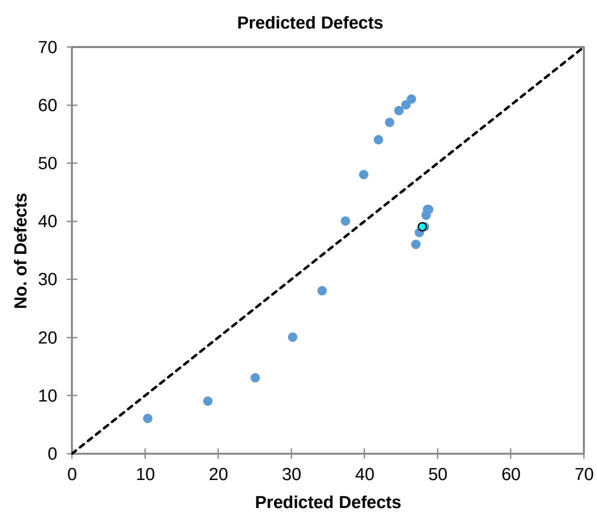

Fig. 8. Predicted Defects vs No. of Defects for the TANDEM Release 3.

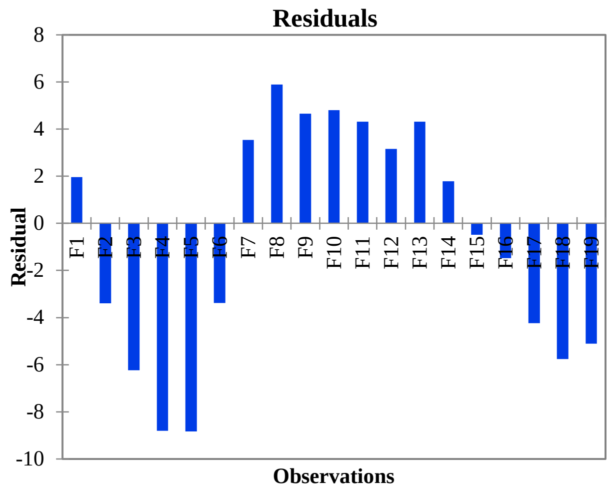

Fig. 9. Observations versus Residuals for the TANDEM Release 3.

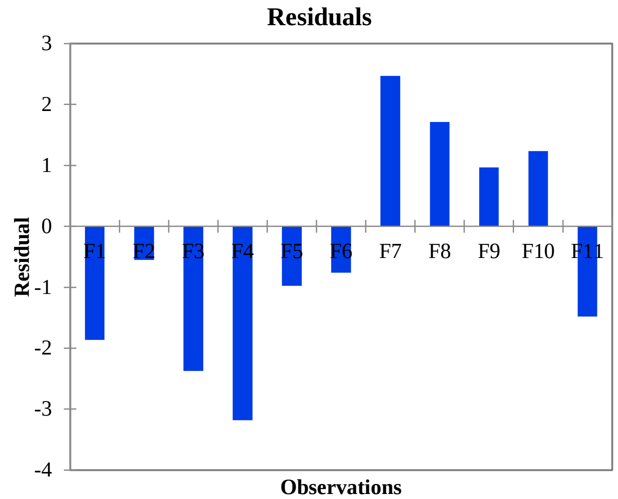

Fig. 12. Observations versus Residuals for the TANDEM Release 4.

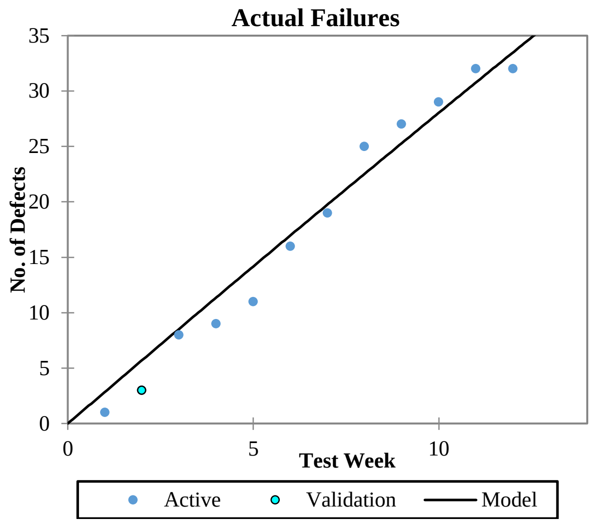

Fig. 10. Actual failures vs No. of Defects for the TANDEM Release 4.

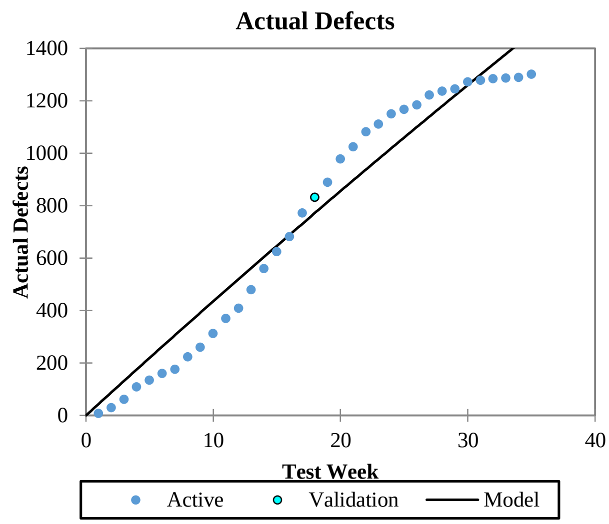

Fig. 13. Actual failures vs No. of Defects for Brooks & Motely.

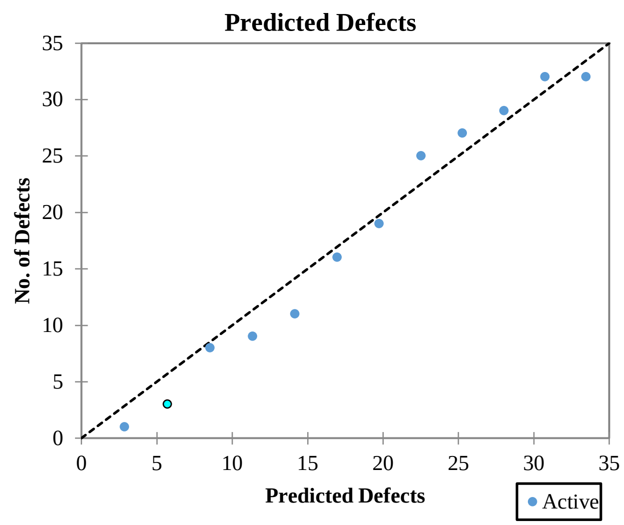

Fig. 11. Predicted Defects vs No. of Defects for the TANDEM Release 4.

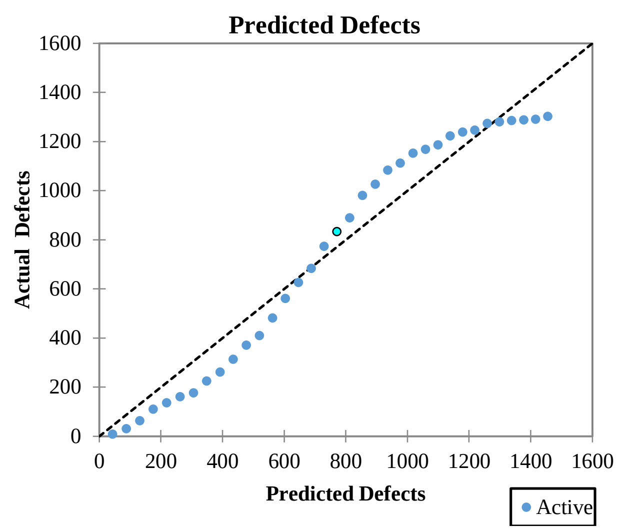

Fig. 14. Predicted Defects vs No. of Defects for Brooks & Motely.

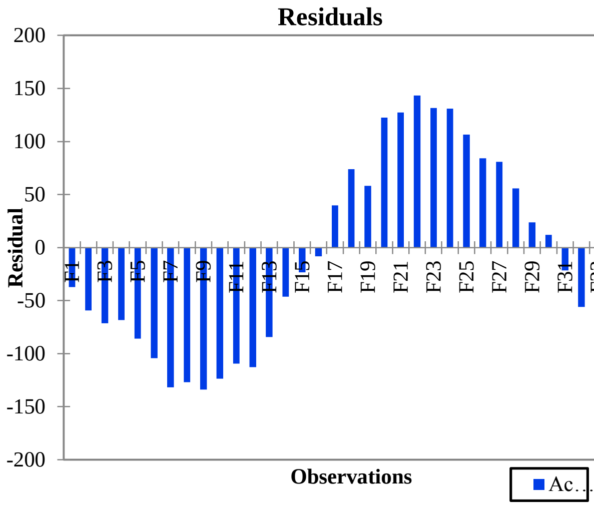

Fig. 15. Observations versus Residuals for Brooks & Motely.

VIII. Conclusion

In this article an ideology is presented which is novel for the minimization of failures and also facilitating the software developer to understand the actual failures that are derived from the project because of some of the technical flaws and also highlighted the predicted failures, which are not the failures but reported as failures due to the issues of technicality or human failures. The works presented in this article on two benchmark datasets helps to understand the potentiality of the model. The results also attribute the significance of the model and this model can be implemented into a software firm helps to not only minimize the review times but also helps to release the software just in time together with enhancing the profit budget.

Acknowledgment

We would also like to show our gratitude to *Dr Srinivas Yarramalle*, Professor, Department of Information Technology, GITAM Deemed to be University, Visakhapatnam for sharing his pearls of astuteness with us throughout the course of this research, and we are also enormously grateful to *Dr Naveen K Kuppilli* Assistant Professor, Department of Information Technology, GITAM Deemed to be University, Visakhapatnam for his remarks on a former adaptation of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed persons.

References

[1] Hudson, G. R.; Programming Errors as a Birth-Death Process, Technical Report SP-3011, System Development Corp., 1967.

[2] Pham, H., Software Reliability, Springer-Verlag, New York, 2000.

[3] Michael R. Lyu; Handbook of software reliability and system reliability, McGraw-Hill, Inc., Hightstown, NJ, 1996.

[4] John D. Musa, Software Reliability Engineering: More Reliable Software Faster and Cheaper, Authorhouse, 2004.

[5] Jelinski, Z.; Moranda. P. B.; Software Reliability Research," (Statistical Computer Performance Evaluation), W. Freiberger, Ed., Academic Press, Inc., New York, 1972, 465-484.

[6] Shooman, M. L.; W. Freiberger, ed., Statistical Computer Performance Evaluation Academic Press, W.Freiberger (ed.), New York, 1972.

[7] Littlewood, B.; Verrall, J. L.; A Bayesian Reliability Growth Model for Computer Software, Proceedings, IEEE Symposium on Computer Software Reliability, New York, 1973, 22, 332-346.

[8] Goel, A.L.; Okumoto, K.; Time-Dependent Error Detection Rate Model for Software Reliability and Other Performance Measures, IEEE Transactions on Reliability, 1979, 28, 206-211.

[9] A. Wood, Predicting software reliability, IEEE Computer, 11, 1996, 69–77.

[10] Brooks WD.; Motley RW.; Analysis of discrete software reliability models – technical report, New York: Rome Air Development Centre; 1980.

[11] Read, C.B.; Gompertz Distribution‖, Encyclopedia of Statistical Sciences, 1983, Wiley, New York.

[12] Ohba, M, Software reliability analysis models," IBM Journal of Research and Development, IBM Journal, Research Development, 1984, 28, 428-443.

[13] Ohba.M; Inflection S-shaped software reliability growth model, Stochastic Models in Reliability Theory, Springer- Verlag Merlin, 1984, 144 – 162

[14] Yamada, S.; Osaki, S.; S-Shaped Software Reliability Growth Models and Comparisons, Applied Stochastic Models and Data Analysis, 1985, 11, 1431-1437.

[15] Musa, J.D.; Iannino, A.; Okumoto, K.; Software Reliability: Measurement, Prediction and Application, McGraw-Hill, 1987.

[16] Brown D. B., Correction to A Cost Model for Determining the Optimal Number of Software Test Cases, IEEE Transactions on Software Engineering, 15(6), 824, 1989.

[17] Tohma, Y. ; Jacoby, R. ; Murata, Y. ; Yamamoto, M.; Hyper-geometric distribution model to estimate the number of residual software fault, Proc. COMPSAC-89, Orlando, 1989,610-617.

[18] Kapur PK; Some modelling peculiarities in software reliability, International conference on quality, reliability and infocom technology, New Delhi, India; 2007, 155–65.

[19] Kapur PK; On modelling failure phenomenon of multiple releases of a software in operational phase for product and project type software – a theoretical framework,International conference on quality, reliability and infocom technology; 2007, 605–18.

[20] Pham, H., Software Reliability, Springer-Verlag, New York, 2000.

[21] Kuo, S.Y.; Hung, C.Y.; Lyu, M.R.; Framework for modeling software reliability, Using Various Testing-Efforts and Fault-Detection Rates, IEEE Transactions on Reliability, 2001, 50, 310-320.

[22] Khan, M. G. M.; Ahmad, N.; Rafi, L. S.; Modeling and analysis of software reliability with Burr type X testing-effort and release determination Proc of the CASON-2008, IEEE Computer Society, China, 2008, 759-762.

[23] Ohishi, K.; Okamura, H.; Dohi, T.; J of Systems and Software, 2009, 82. 535-543.

[24] SatyaPrasad, R.; Naga Raju, O.; Kantam, R.R.L.; SRGM with Imperfect Debugging by Genetic Algorithms I. J. of Software Engineering & Applications, 2010, 1, 66-79.

[25] Satya Prasad, R.; Ramchand H Rao, K.; Kantam, R.R.L.; Software Reliability with SPC, International Journal of Computer Science and Emerging Technologies, 2(2),2011, 233-237.

[26] Kantam, R.R.L; Subrahmanya Ravikumar, M.; Limited Failure Censored Life Test Sampling Plan in Burr Type X Distribution. Journal of Modern Applied Statistical Methods, 15, 2016, 428-454.

Authors' Profile

Jagadeesh Medapati received M.Tech from GITAM Deemed to be University, Visakhaptnam in 2009, currently pursuing his PhD from JNT University, Hyderabad. He is currently working as an Assistant Professor in the Department of Computer Science & Engineering, GITAM Institute of Technology, GITAM Deemed to be University, Visakhapatnam, India. His research interest includes Software Reliability, Software Engineering, Theoretical Computer Science and Computational Intelligence.

Prof Anand Chandulal Jasti received M.Tech from JNT University, Hyderabad in 1992, PhD from JNT University, and Hyderabad in 2007. He is currently working as Professor in Department of Computer Science & Engineering, GITAM Institute of Technology, GITAM Deemed to be University, Visakhapatnam, India. He has several publications in national & International journals and conferences. his research interest includes Text Mining Security, Soft Computing and Optimization Technologies

Prof TV Rajanikanth received M.Tech from Osmania University, Hyderabad in 2001, PhD from Osmania University, and Hyderabad in 2007. He is currently working as Professor in Department of Computer Science & Engineering, and Dean for Research & Development of SREENIDHI Institute of Science and Technology, Hyderabad, Telangana, India. His contribution towards research made him to publish several research articles in renowned national & International journals and conferences. His research interest includes Data warehouse & mining, Semantic Web, Spatial Data mining, ANN.