International Journal of Advance Research in Science and Engineering
Vol. No.5, Special Issue No. (01), February 2016
www.ijarse.com
IJARSE
ISSN 2319 - 8354

SOFTWARE RELIABILITY AND ITS EXPONENTIAL FAILURE TIME GROWTH MODELS

Dr.N.Supriya[1], Dr.G Jose Moses[2], K.Jeswanth Kumar[3]

[1,2]Associate Professor, Department of CSE, Raghu Institute of Technology, Visakhapatnam, (India)

[3]Associate Professor, Department of CSE, KITS Engineering College, Andhra Pradesh, (India)

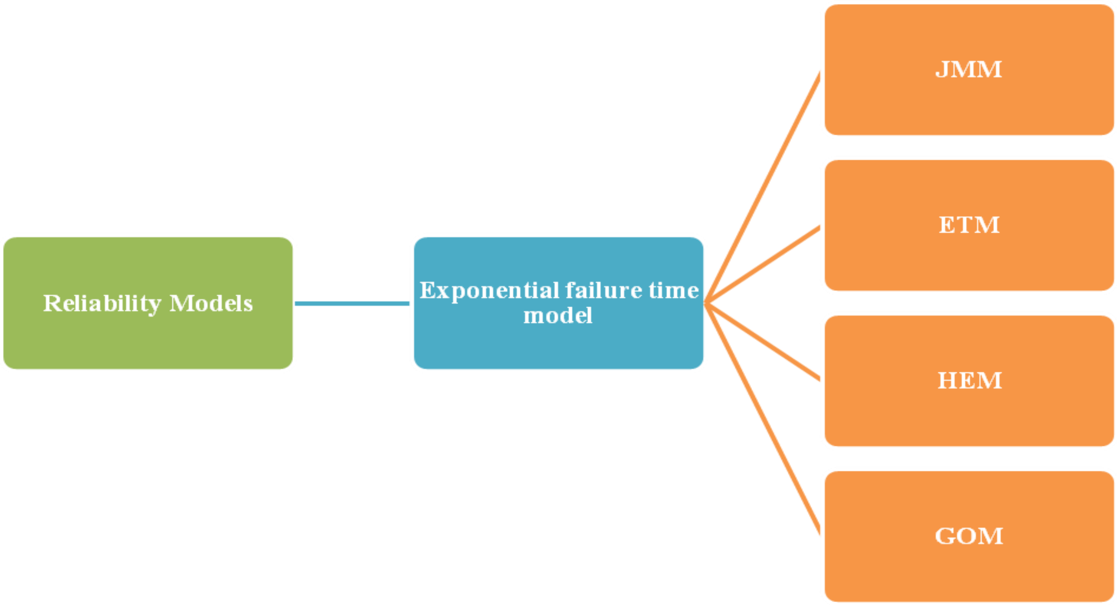
ABSTRACT

The aim of software engineering is to produce reliable software at less cost. Software reliability is one of the required qualities of software development process. Reliability of software is defined as the probability of expected operation over specified time interval. In this paper we discuss about the software reliability exponential failure time growth models and their utility. The utility of a software reliability growth model is related to its stability and predictive ability. Stability means that the model parameters should not significantly change as new data is added. Predictive ability means that the number of remaining defects predicted by the model should be close to the number found in field use.

***Keywords*: Software Engineering, Software Reliability, Stability, Predictive.**

I. INTRODUCTION

The aim of software engineering is to produce reliable software at less cost. With growth in size and complexity of software, several issues come into existence. Reliability is one of the required qualities of software development process. Reliability of software is basically defined as the probability of expected operation over specified time interval. Software Reliability is an important attribute of software quality with performance, functionality, usability, capability, maintainability, and documentation. Software reliability is defined as the failure free operation of computer program in a specified environment for a specified time. Software Reliability is hard to achieve, because the complexity of software tends to be high. Software reliability is a critical component of computer system availability, so it is important that its customers experience a small number of software failures in their production environments. Software reliability growth models can be used as an indication of the number of failures that may be encountered after the software has shipped and thus as an indication of whether the software is ready to ship. These models use system test data to predict the number of defects remaining in the software. Software reliability growth models have been applied to softwares of several releases over the past few years. The utility of a software reliability growth model is related to its stability and predictive ability. Stability means that the model parameters should not significantly change as new data is added. Predictive ability means that the number of remaining defects predicted by the model should be close to the number found in field use.



1.1 Exponential Failure Time Model (EFTM)

Exponential models comprise of all finite failure models. Poisson and Binomial are two categorization of EFTM [1]. The Binomial and Poisson types are based on per fault constant hazard rate. Hazards rate function is defined as the function of the remaining number of faults and the failure function is exponential.

$$H (Z) = f (RNF) +f((exp(FF))$$

Where, H (Z) = Hazard rate.

RNF=Renaming number of faults.

FF= Failure Function.

1.2 J-M Model (JMM)

The failure time is proportional to the remaining faults and taken as an exponential distribution [2]. During testing phase the number of failures at first is finite. Concurrent mitigation of errors is the main strength of the model and error does not affect the remaining errors. Error removal is all human behaviour which is irregular so it cannot be avoid by introducing new errors during the process of error removal.

$$(MTBF)_t =1/ (N-(i-1))$$

Where, N= Total number of faults. i= Number of fault occurrences. MTBF=Mean Time between failure.

t= Time between the occurrence of the $(i-1)^{st}$ and i^{th} fault occurrences.

1.3 Execution Time Model (ETM)

Musa‘s Basic model assumes that all faults are equally likely to occur and independent of each other [3]. The intensity function is directly proportional to the number of faults remaining in the program and fault correction is proportional to the number of failure occurrence rate.

$$\mu(t) = \beta_0 (1\text{-exp} (-\beta_1 t))$$

Where, μ (t) = mean value function at time t.

β_0 = Total number of faults.

1.4 Hyper Exponential Model (HEM)

The idea behind this model is that the different parts of the software experience an exponential failure rate. However the rate varies through these parts to ponder different behaviors Different failure rate are placed in different sections [4]

$$\lambda(t) = N \sum pi\beta i(\exp(-\beta_1 t))$$

Where, λ (t)=Failure Intensity Function.

t= number of failures. N= finite number of failures.

Pi=particular ith class. βi=total number of ith faults.

1.5 Goel-Okumoto Model (GOM)

Goel-Okumoto model [5] takes the number of faults per unit time as independent random variables. In this model the number of faults occurred within the time and model estimates the failure time. Delivery of software within cost estimates is also decided by this model.

$$\mu(t) = E_E (l\text{-}e^{-bt}), \text{ where } E_E \geq 0, b>0$$

μ(t) = Predicted number of defects at time t

E_E = Expected total number of defects in the code in infinite time (it is usually finite)

b = Roundness factor/shape factor = the rate at which the failure rate decreases

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

II. CONCLUSION

When a software system is designed, the major concern is the software quality. The quality of software depends on different factors such as software reliability, efficiency, cost etc. In this paper we presented various existing software reliability exponential failure time models to measure the software reliability.

III. ACKNOWLEDMENT

The authors wish to thank for the encouragement given by the management of Raghu Educational Institutions, Chairman Sri.Raghu Kalidindi, Secretary Smt.Rama Devi Kalidindi and Rahul Varma Kalidindi towards research initiatives.

REFERENCES

[1] A.Wood, Software Reliability Growth Models, Technical Report 96.1, *Tandem Computers*, September 1996, pp. 1-30.

[2] A. L. Reibman, M. Veeraraghavan, Reliability Modeling: An Overview for System Designers, *AT&T Bell Laboratories*, IEEE, 1991, pp.49-57.

[3] J. D Musa, A Theory of Software Reliability and Its Applications, *IEEE Transactions on Software Engineering*, vol. SE-1, no. 3, 1975, pp. 312-327.

[4] M. Ohba, Software Reliability Analysis Models, *IBM Journal of Research and Development*, vol. 21, no. 4, July 1984, pp. 428-443.

[5] Goel, A. L. and Kazuhira Okumoto, A Time Dependent Error Detection Model for Software Reliability and Other Performance Measures, *IEEE Trans. Reliability*, vol R-28, Aug 1979, pp 206-211.

[6] Liu, *The Hand Book of Software Reliability Engineering* (Mc Graw Hill Publications).

[7] Jelinski. Z, Moranda P.B., *Software Reliability Research Statistical Computer Performance Evaluation*, (W. Frebierger, New York, Academic Press).

AUTHOR'S PROFILE:

	**Dr.N.Supriya** working as Associate Professor, in the department of Computer Science and Engineering, Raghu Institute of Technology, Visakhapatnam. She obtained her M.Tech(CSE) and Ph.D(CSE) from ANU,Guntur. She has presented papers in national and International conferences. She is member of various professional bodies.
	**Dr. G. Jose Moses** working as Associate Professor (Ratified by JNTUK), in the department of Computer Science and Engineering, Raghu Engineering College, Visakhapatnam. He is Approved Research Supervisor for JNTUK. He obtained his B.Tech(CSE) from JNTUH, M.Tech(CSE) from ANU,Guntur, Ph.D(CSE) from ANU, Rajahmundry. He has presented and published papers in National and International conferences.
	Mr.K.Jeswanth Kumar is working Associate Professor, Department of CSE, KITS Engineering College, His area of Interest includes Computer Networks, Software Engineering. He has presented and published papers in National and International conferences.