International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-5, Issue-3, May 2017
DOI: 10.21276/ijircst.2017.5.3.6

# Analysis of Various Software Reliability Models and Proposing a New Model of Software Reliability for Embedded Systems

**Dr. Rajender Kumar Sharma, Dr. Archana Kumar, Sapna Bajaj**

*Abstract:* **Software plays a major role in almost embedded systems. The examples of such softwares are Life- saving systems used in medical, mission critical systems used in missiles, launching of any satellites etc. apart from the functional complexities of available softwares, reliability of software with embedded systems has become a serious concern. Various techniques and methods are being formulated to predict or estimate the quality of such embedded software system from its structure in terms of quality and other parameters like reliability, safety and performance. However, the results of these quantitative structures for evaluation of reliability depend on design-time estimates for a series of model parameters. To overcome this problem, this paper presents a new design-time architecture evaluation method that includes uncertainties of various parameters. It first analyses some of available reliability models and proposes a new software reliability model for estimating, measuring and controlling software reliability of embedded system where various different parameters need to be estimated .**

*Index Terms:* **Embedded system, Failure Rate, Hazard Function, MTBF ,Reliability, Reliability Function, Software Reliability Growth Models.**

## I. INTRODUCTION

An embedded system is a computer system that has been designed to perform a few dedicated or specific tasks with real-time conditions. It is embedded as part of a complete device that includes hardware and different mechanical parts. As a general-purpose simple computer, such as a personal computer or laptop, is structured to be flexible and to meet a wide range of end-user requirements, embedded systems control many devices in day to day life of users.

Embedded systems contain different processing cores that are usually either microcontrollers or digital signal processors (DSPs). The important characteristics of Embedded Systems are being dedicated to handle a specific task. They usually requires very strong processors and suitable communication, for example air traffic control systems usually be considered as embedded system.

**Manuscript received May 20, 2017**

**Dr. Rajender Kumar Sharma**, Professor, Agra College of Engineering & Technology, Agra,

**Dr. Archana Kumar**, Director, Delhi Institute of Tech. & Mgt., Gannaur

**Sapna Bajaj,** Assistant professor B.M. Institute of Engg. & Tech., Sonepat

As the embedded system is dedicated to do specific tasks, designers can optimize it to decrease the cost and size of the product and increases the reliability and other parameters like performance. Some embedded systems are produced in bulk, benefiting from economies of scale .This enhances the need of Software Reliability in an embedded system environment.

Software Reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Software Reliability is an important factor that affects reliability of system. It is different from hardware reliability in that it shows the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of problems of Software Reliability. Software Reliability does not depends upon the time. The modelling technique has been used to formulate or compute the reliability of software but no appropriate quantitative methods have been developed till now to define Software Reliability without excessive limitations. Various strategies can be used to improve the reliability of software, but it is difficult to manage development time and budget with software reliability.

Software Reliability is an important feature of software quality, combined with capability, usability functionality, performance, serviceability, installability, documentation and maintainability. Software Reliability is difficult to achieve, because the high complexity of the software. As electronic gadgets are inceasing very fast, softwares are used in almost all broad fields. Most of softwares that control microprocessors or chip of these electronic devices and other are fixed in ROM. These softwares are known as embedded softwares. These types of softwares are written to control devices or machines that are not fully thought of as computer systems.It has hard time and memory constraints. As these software works on the particular hardware, it is quite difficult to test them . In embedded software systems, software faults may arise at integrated phase of hardware and software and software faults may occurs in testing software on specific hardware. We cannot fix those errors which cause software faults, whenever software errors are occurred, because we cannot write the program and data field software errors on the ROM. As more and more softwares are becoming the part of embedded systems, we should verify they don't embed errors into these systems . If not handled carefully, software reliability of a specific part can affect the reliability of the whole embedded software system. Ensuring software reliability is not an easy job. As difficult as the problem is, progress is still being made toward more reliable software. More standard components and better procedures are introduced in software engineering field. By now most of the researchers and technology have been focussed on increasing reliability of software or a hardware component in any environment and have got success in obtaining the targets . But as technology is taking new heights and embedded systems are effecting our lives much more than previous times through new applications every day, So there is a need to think of some new methods and techniques to make embedded software systems more reliable and robust. There are various software reliability growth models available till now. Software reliability modeling techniques are classified into two categories: Predicting Modelling and Estimating Modelling. Predicting modeling uses historical data to predict the reliability of softwares estimating modeling uses current data and data set from the current software development processes and don't use the historical data and can estimate the reliability of software at any time even during the development phase of software also. Software reliability growth models uses failure data from testing to predict the failure rate or Mean time between failure for the future. These SRGMs depends upon the assumptions about the fault rate during testing phase which can either be increasing, decreasing or some mixture of decreasing and increasing. Some SRGMs assume that there are fixed and finite number of inherent defects while others considers that defects are infinite. Some models requires more efforts for estimating parameters and others have only a few parameters to estimate. Some models requires the actual time in between each failure found in testing, while other models need to have the number of failures found during any given time interval such as an hour or a day.

## II. MEASUREMENTS OF SOFTWARE RELIABILITY

Various software reliability models used for estimating software reliability are:

***A).Jelinski-Moranda de-eutrophication model:*** The de-eutrophication model developed by Jelinski and Moranda in 1972 was one of the first software reliability growth models.It assumes that there are N software defects initially at the start of the testing procedure and each other is independent to the other defects.A defected fault is removed for sure and no other new faults are introduced during the debugging phase.So the failure rate of software or we can say, the Hazard Function is directly proportional to the number of errors present in the software at that time.So the Hazard Function during the time $t_n$ which is the time during (n-1) and $n^{th}$ Failure is represented as:

$Z(t_n) = \phi[N-(n-1)]$ ,

Where, $\phi$= Proportionality Constant and,
N= Software Faults at the start of testing.
A typical plot of Jelinski – Moranda de- eutrophication Model is represented as:

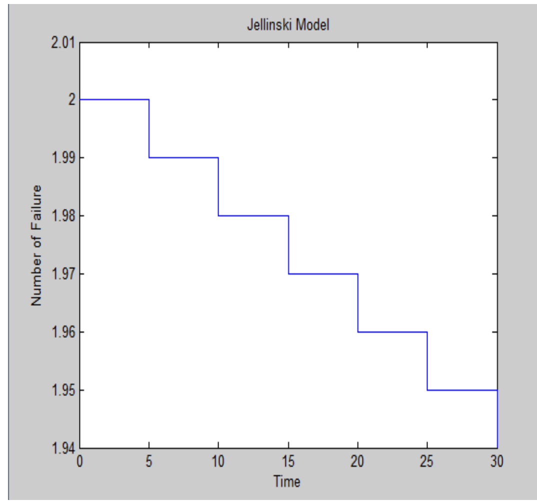



Fig 1: Jelinski-Moranda de-eutrophication model

***B) Goel –Okumoto model:***
The model was proposed by Goel and Okumoto in 1979 .Goel and Okumoto developed an imperfect debugging model which was the extension of Jelinski- Moranda Model.
Here, the number of faults in any software system at time n, X(n) are represented by Markov Process. Time between the change of fault X(n), depends upon the current faults available in the system.So the Hazard Function in Goel-Okumoto model during the time $t_n$ which is the time during (n-1) and $n^{th}$ Failure is represented as:

$Z(t_n) = [N- p\ (n-1)]\lambda$
Here, N= Initial Faults found in the software system,
P=Probability of imperfect Debugging,
$\lambda$= Failure rate per fault.
A typical plot of Goel- Okumoto Model is represented as:

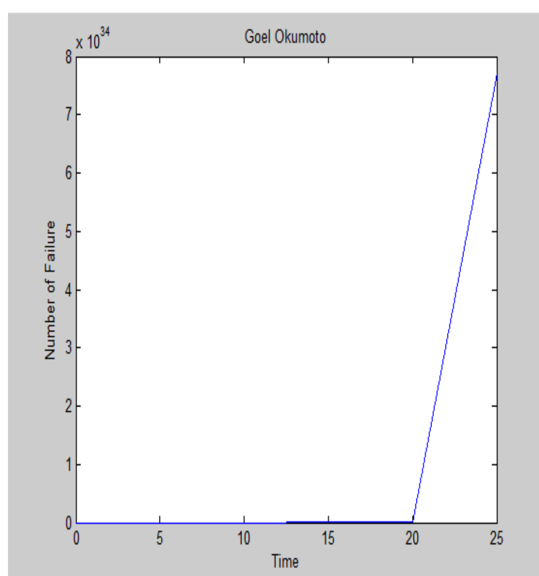



Fig 2 Goel –Okumoto model

ISSN: 2347-5552, Volume-5, Issue-3, May 2017
DOI: 10.21276/ijircst.2017.5.3.6

***C ) Littlewood software Reliability Model:***

Littlewood and Verrall considered a different approach to the development of the model for estimating the mean time between failures. They claimed that software reliability is not directly dependent upon the software faults occurred in the program.In this model the Hazard Function is dependent on the quality of the programmer and difficulty in the programming task.

A plot of Littlewood Software Reliability model is represented as:

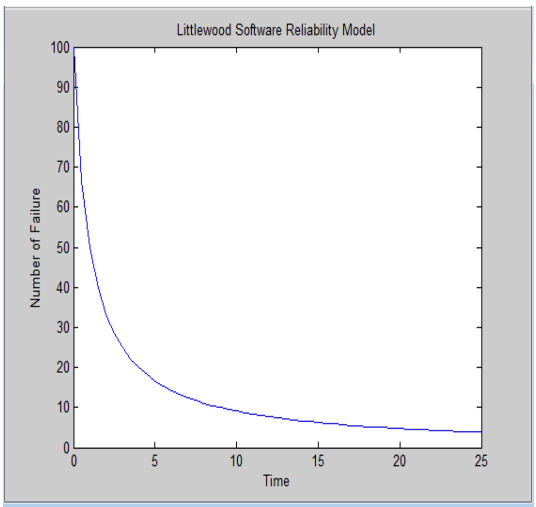



Fig 3: Littlewood Software Reliability model

**Problem associated with above reliability models:**

Although these SRGMs are widely used to predict the reliability of the developing software, still there are some shortcomings associated with these reliability models.These are:

1. The arguments which has been used in these software reliability growth models are not applicable on all types of softwares because software development process vary from environment to environment.some softwares work efficiently in one environment but not in others.So it is not possible to identify the real
errors of the software

2. MeanTime between failures in a software system are not dependent on each other so it becomes quite difficult to generate proper test cases.

3. New faults can be introduced during the fault removal process which can also leads to the failure of the software.

4. In these models, time is considered as the basic and most important unit for finding the failure rate and estimating the reliability but apart from this factor, some other factors like efficiency of programmer, complexity of code etc. should also be taken into consideration for estimating the reliability more accurately.

5. These models suggests that failure rate is directly proportional to the remaining number of available faults in the system, but it is not possible always as some modules contains more errors and some contains not even a single one.

## III. PROPOSED RELIABILITY MODEL

In this paper, we propose a new model to estimate reliability of the software by taking into consideration the above problems and effects of both software and hardware on software failures that occur in the embedded systems.

The proposed model for predicting the reliability of the embedded software is simple random failure based model.Using this model, reliability of software can be predicted at an early stage of software development which is more useful for estimating the feasibility of development of product in terms of time, cost , maintainability etc. and proper test cases can be generated at an early stage by using proper testing tools and procedures.

Before considering the simple random failure based model following assumptions are needed to be considered:

1. Consider the failures caused by both hardware and software and caused due to their interaction point also because embedded system is a combination of both hardware and software so we can't neglect any one of them.
2. Software faults or failures can be removed by Goel – Okumoto Model and hardware faults can be removed by replacing the faulty hardware.

Wherever the interfaces or interaction points where software and hardware meet or interact with each other can be removed by simple random failure based model.

Here the occurrence rate of failure in case of embedded softwares is estimated by summing the occurrence rate of software failure caused by hardware and failures caused by purely softwares.

Here

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

Here, $\lambda_{ES}(t)$ = failure rate of embedded system,
$\lambda_{sw/hw}$ = failure rate of software that causes due to hardware,
$\lambda_{sw}(t)$ = failure rate that causes purely due to software and
t = the time of failure.

Eq(1). can also be written as:

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

so the Function for calculating mean value $m_{ES}(t)$ for embedded system software can be written as:

$$m_{ES}(t)= \lambda_{sw/hw} \square t+ a\,(1-e^{-bt}) \quad (3)$$

eq(3) can be obtained by applying integration on eq(2).

So, by using eq(3)., the reliability function of embedded software $R_{ES}(x/t)$ where t is the time of testing($t\geq 0$, $x>0$), can be calculated as:

$$R_{ES}(x/t) = e^{-( \lambda sw/hw * x+ae-bt(1-e-bx))} \quad (4)$$

Here, eq.( 4) is our resultant equation which gives the reliability function $R_{ES}(x/t)$.

## IV. CONCLUSION

Here in this paper, we presented a new method for calculating the reliability of embedded software by generating a new software reliability growth model. This method provide us more accurate estimation for calculating software reliability of embedded softwares than other traditional methods and it is best suited for softwares where estimation of reliability at an early stage is quite important.

## REFERENCES

[1]Kapsu Kin, Chisu Wu “A software reliability Model in an Embedded System” pp.59-63.

[2]John Regehr “Real Time And Embedded Systems- teaching Reliability” in IEEE Computer Society Vol. 7, No. 5,May 2006, pp 1-3.

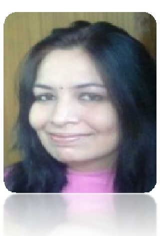
[3] Xin He, Yun-Jun Li “Software Reliability Analysis on Embedded System Based on SFMEA and SFTA Model”, 2012 International Conference on Systems and Informatics,pp 2471-73.

[4] Yanping Yu, Guoping Zheng, Zhengming Qian “Software Reliability Model Analysis including Internal Structure based on Bayesian Network”, Fourth IEEE International Conference on Cooperation and Promotion of Information Resources in Science and Technology” pp.247-250.

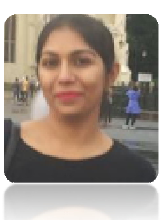
[5] Andriy Sydor “Modeling Indices of Reliability for Ramified Embedded Systems” MEMSTECH’2008, May 21-24, 2008, Polyana, Ukraine, pp.75-76.

[6] Anne Immonen, Antti Niskanen “Anne Immonen and Antti Niskanen”, Proceedings of the 2005 31st EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO-SEAA’05)IEEE.

Dr. Rajender Kumar Sharma is a Professor in Computer Science Department at Agra College of Engineering, Agra.He received his Ph.D degree from Xavier International University in 2011 .His research interests include biometric networks and software engineering.



Dr. Archana Kumar is Director and Professor at Delhi Institute of Technology and Management, Gannaur.She received her Ph.D degree from Delhi University in .Her research title is” AStudy in Software Reliability Growth Modeling and Distributed Development Environment.”



Sapna Bajaj is a Ph.D Computer Science scholar in MDU(Al-Falah School of Engg. And Tech.).She received her M.Tech degree in 2010.She is Assistant Professor in Computer Science Deptt. At BMIET, Sonepat.Her areas of Interest are Software Engineering, Computer Networks, Operating Systems etc.