Software Reliability: Models

Durga Patel
M.Tech Scholar
Department of Computer Science
Banasthali University Rajasthan, 304022 India

Pallavi
M.Tech Scholar
Department of Computer Science
Banasthali University Rajasthan, 304022 India

ABSTRACT
Need of complex system is increased more rapidly. Size and complexity of computer systems has grown during last past few years in very impressive manner. Different Software reliability models have discovered since last 30 years. There has lot of work is done in field of software reliability estimation. Some of important models have been discussed in the literature review of the paper. Various dimensions have discussed on which reliability models is based. Those models have reviewed has reflect infinite and finite failures. On the other hand some of the models are based upon logarithmic distribution and they reflect infinite failures.

Keywords
Software reliability models (SWRM); software reliability models dimensions (SWRMD); software reliability (SWR); taxonomy of SRGM.

1. INTRODUCTION
In general software reliability has two categories of models: Deterministic and probabilistic models. The study of defined distinct operators and operands in the program is employed as deterministic and probabilistic is one which represents failure occurrences and fault removal as probabilistic events. As software development is moving towards component-based software design as a result we need software reliability modelling.

2. LITERATURE REVIEW
A. Year 2001
In this paper author have explained the architecture-based approach to reliability assessment of software system and the common requirement of architecture based models are identified and the classification is proposed. The architecture-based models are classified into three categories: State-based model, Path-based models, and Additive-based models. The architecture-based approach is used for the quantitative assessment of component-based software system.

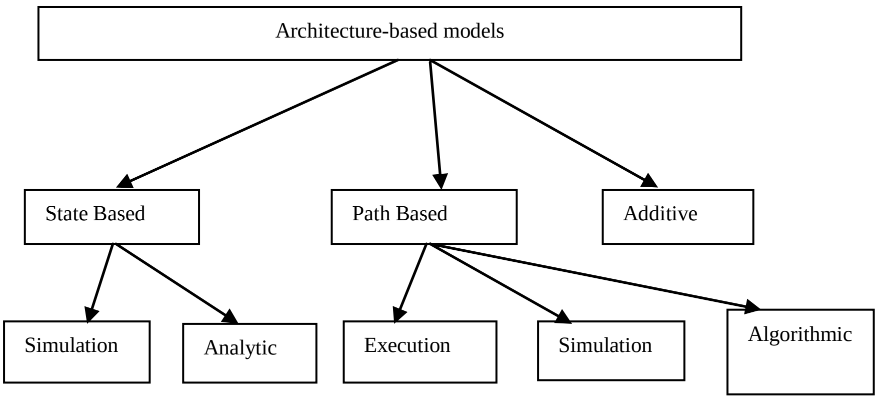

Figure 1: Architecture Based Models

Critic: The overall motive of paper is to present the existing architecture-based approach to assess the reliability of the software in the present and existing approaches of the software engineering.

B. Year 2003
The authors 'Hoang Pham*, Xuemei Zhang' have proposed a model for software reliability that is incorporates with testing coverage information. This model is based on nonhomogeneous Poisson process (NHPP) and can used to estimate and predict the software reliability of the product in a quantitative manner and we also examine the goodness-of-fit and estimation power of the model.[12]

Critic: In the paper comparison of NHPP based models is presented and software cost model is also explained which estimate the testing cost, fault removal cost, and risk can cost due to potential problems remaining in the uncovered codes.

C. C. Year 2012
a) The author 'Razeef Mhod & Mohsin Nazir' has summarize some existing software reliability growth models, provides critical analysis of underlying assumptions and assess the applicability of these models during the software development cycle. Software reliability growth model is a technique used to assess the reliability of the software product in quantitative manner and this model have good performance in terms of goodness-of-fit, predictability and so forth.[7]

***Critic*:** The overall research effort provided the overview of some existing software reliability growth models. There are four classes of analytical models, along with their underlying assumptions.

b) In this paper author 'Gaurav Aggarwal & Dr. V.K Gupta' have categories software reliability models into two ways: static model and dynamic model. Observation of the temporary behavior of debugging process during testing phase is known as dynamic models. Modeling and analysis of program logic is done on the same code in static models. Software reliability growth model describes about error detection in software reliability.

Critic: The study of this paper explains that on which factor software reliability depends are: failure intensity function, and mean value function. Some experiment has been done with the data on the existing models to prove there formula's.

D. *Year 2014*

a) In this paper author B. Anniprincy & Dr .S. Sridhar have Propose two dimensional Software reliability growth model by using Cobb-Douglas production function for capturing effect of testing time and testing coverage on the number of fault removed in the software system. S-shaped model is develop in the paper which we have reviewed.[10]

Critic**:** In this paper a general approach have bee developed which is used in driving more general models based on simple assumptions, constant with the basic software reliability growth modelling based on NHPP.

A. Ideal Developed Models

Markov Models (State-Based Model):

Models of this class use the control flow graph to represent the architecture of the system. It is assumed that the transfer of control between modules has a Markov property. It gives knowledge of the modules at the control at any given time.

In this state-based models is ones which is used to represent the application architecture either in DTMC (Discrete time Markov chain) and CTMC (Continuous time Markov chain)[11][14]

1. **DTMC-Based Models:-** This model uses the number of test cases as a unit of fault detection period.

Table 1: Overview of DMTC-Based Models

DTMC-Based Models	Architecture Model	Failure Model	Solution Methods
DTMC 1	Absorbing DTMC(A-DTMC)	Reliabilities	Composite
DTMC 2	Absorbing DTMC(A-DTMC)	Reliabilities	Hierarchical
DTMC 3	Absorbing DTMC(A-DTMC)	Constant failure rates	Hierarchical
DTMC 4	Absorbing DTMC(A-DTMC)	Time-dependent failure intensities	Hierarchical
DTMC 5	Irreducible DMTC(I-DMTC)	Reliabilities	Composite
DMTC 6	Irreducible DMTC(I-DMTC)	Reliabilities	Hierarchical
DMTC 7	Irreducible DMTC(I-DMTC)	Constant failure rates	Hierarchical

2. **CTMC-Based Models:-** The model uses machine execution time means cpu time or calendar time as a unit.[11]

Table 2: Overview of CTMC-Based Models

CTMC-Based Models	Architecture Model	Failure Model	Solution Methods
CTMC 1	Absorbing CTMC (A-CTMC)	Constant failure rates	Composite
CTMC 2	Absorbing CTMC (A-CTMC)	Constant failure rates	Hierarchical
CTMC 3	Absorbing CTMC (A-CTMC)	Reliabilities	Hierarchical
CTMC 4	Absorbing CTMC (A-CTMC)	Time-dependent failure intensities	Hierarchical
CTMC 5	Irreducible CTMC (I-CTMC)	Constant failure rates	Composite
CMTC 6	Irreducible CTMC (I-CTMC)	Constant failure rates	Hierarchical
CMTC 7	Irreducible CTMC (I-CTMC)	Reliabilities	Hierarchical

B. Software Reliability Growth Models:

The models refers to those models that try to predict software reliability from test data and these model show a relationship between fault detection data and know mathematical functions such as logarithmic or exponential functions. The model which describes about error detection in software reliability is called software reliability growth models.[1],[2],[3][7],[4]

There are various types of models which are based on SRGM are as follow:

Jelinski-Moranda Model:- The model was first introduced as software relaibility growth model in Jelinski and Moranda (1972).There are certain features of this model are : This is a contnious time-independently distributed inter failure time and independent and identical error behaviour model.[8][6]

Goel-Okumoto Model:- Goel and Okumoto proposed this model first and it is one of the most populor NHPP model in the field of software relaibility modelling and it is also called exponential NHPP model.[8],[6]

Generalized Goel NHPP Model:

This is the generalization of Goel-Okutmoto model and is proposed by Goel to descirbe the situtaion that software failure intensity increases slightly at the beinging and then begins to decrease. [8],[6]

Inflected S-Shaped Model:- This model is proposed by ohba and its underlying concept is that software relaibility growth becomes S-Shaped if faults in a program are mutually dependent and some faults are not detectable before some others are removed and this model solves a technical problem in the Goel-Okumoto model.[8],[6]

Logistic Growth Curve Model:- This model is developed to predict economic population growth could also be applied to perdict software relaibility growth.Logistic growth curve model is one of them and it has an S-Shaped curve.[8],[6]

Musa-Okumoto Model:- In this model one property is incoporated which is explained by Musa-Okumoto. They have observed that the reduction in failure rate resulting from repair action following early failures are often greater because they tend to the most frequently ocurring once.[6],[8]

Yamada Delayed S-Shaped Model:- It is the model with the modification of the non-homogeneous Possion process to obtain a S-Shaped curve for the cumulative number of failures detected such that the failure rate intially increases and later decays.[8]

Table 3 :List Of Key Assumption By Model Category

S.No	Model Name	Proposed By	Proposed Year	Model type	Ref.	Comments
1	Jelinski-Moranda Model	Jelinski-Moranda	1972	-	Ref. [8],[6]	This is continuous time – independently distributed inter failure time model and independent and identical error behavior model
2	Goel-Okumoto Model	Goel & Okumoto	1979	Concave	Ref. [8],[6]	It is also called exponential model or Musa model
3	Goel-Shaped Model		-	-	-	Modification of G-O model to make it S-shaped.
4	Generalized Goel NHPP Model	Goel	-	-	Ref. [8],[6]	Gole proposed generalized model with additional parameter c ,that reflect quality of testing.
5	Inflected S-Shaped Model	Ohba	-	Concave	Ref. [8],[6]	It is same as G-O model .So solve all technical condition with G-O.
6	Logistic Growth Curve Model		-	S-Shaped curve	Ref. [8],[6]	The model developed to predict economic population growth
7	Musa-Okumoto Model	Musa – Okumoto	1984	-	Ref. [8],[6]	-
8	Gompertz Growth Curve Model			S-Shaped	Ref. [8]	Used by Fujitsu,Numazu Works.
9	Yamada Delayed S-Shaped Model	Yamada	1983,1985, 1986	S-Shaped	Ref. [8]	Modification of non – homogeneous Poisson process to obtain s-shaped curve for failure detection.
1	Yamada	Yam	-	Concav	Ref.	Attempt to account for

0	exponential	ada		e	[8]	testing –effort
11	Yamada Imperfect Debugging Model	Yamada et.al	-	Concave	Ref. [8]	Assume exponential fault constant function and constant fault detection rate.
12	Yamada Raleigh	Yamada	-	S-shaped	-	Attempts to account for testing effort.
13	Modified Duane Model	Duane	1962	-	Ref. [8]	-
14	Weibull-Type Testing-Effort Function Model	Weibull	1983	Concave	Ref. [8]	Same as G-O for cm 1, Testing effort functional model.

Table 4: Mean Value And Intensity Of Various Models

Model Name	Mean Value Function	Intensity Value Function
Jelinski-Moranda Model	$m(t) = n0\ 1 - \exp\ -\phi t$	$\lambda i = (N - k)\mu$
Goel-Okumoto Model	$m\ (t) = a\ 1 - exp\ -bt$, $a > 0, b > 0$ a= expected total number of fault b= fault detection rate	$\lambda\ (t) = ab * exp\ -bt$ $a > 0, b > 0$
Generalized Goel NHPP Model	m(t) = a(1-exp([-bt^c]) , $a > 0, b > 0, c > 0$ a= expected total number of fault b,c =reflect quality of testing	$\lambda\ (t) = abctc{-}1\ exp\ -btc$, $a > 0, b > 0, c > 0$
Inflected S-Shaped Model	$m\ (t) = a *(1 - exp\ [-bt\ ]\ /1 + \psi\ r * exp\ -bt)$ n$\psi\ (r) = 1 – r/\ r$ $a > 0, b > 0, r > 0$ a= expected total number of fault R = rate of detectable fault b =fault detection rate	$\lambda\ (t)$ =($abexp\ [-bt$]($1 + \beta t$) / ($1 + \beta * exp[\ -bt]$) ^{2} $a > 0, b > 0, \beta > 0$
Logistic Growth Curve Model	$m\ (t) = a\ /1 + k * exp\ –[bt]$ $a > 0,\ b > 0, k > 0$ a = expected total number of fault K,b = estimated by fitting the failure data.	$\lambda\ (t) =$ $abexp\ -bt\ \ 1 + k * exp\ -bt\ \ 2$ $a > 0, b > 0, k > 0$
Musa-Okumoto Model	$m\ (t) = a * ln(1 + bt)$, $a > 0, b > 0$ a = expected total number of fault b = fault detection rate	$\lambda\ (t) =\ ab\ (1 + bt)$ $a > 0, b > 0$
Gompertz Growth Curve Model	m(t)=ak^{bt}	$\lambda\ (t)$=abln(k)k$^{exp[-bt]}$ exp [-bt]

	$a>0, 0<b<0, 0<k<1$ a= expected total number of fault b= estimated using regression analysis	$a>0, 0<b<0, 0<k<1$
Yamada Delayed S-Shaped Model	m(t)= a(1-(1+bt) * exp[-bt]), a>0,b>0 a = expected total number of fault to be detected b = fault detection rate	$\lambda(t)$= ab^{2}t*exp[-bt], a>0,b>0
Yamada exponential	m(t)=a*(1-exp[-rα (1-exp[-βt])]) a>0,b>0,α>0, β>0 a = total number of fault to be detected α = fault introduction rate r, β =constants	$\lambda(t)$=ara(exp[-rα(1-exp[-βt])])*exp[-βt] a>0,b>0,α>0, β>0
Yamada Imperfect Debugging Model	m(t)=a*b*(exp[αt] -exp[-bt]/α +b) a>0,b>0,α>0 a = total number of fault to be detected b =fault detection rate α = fault introduction rate	$\lambda(t)$=a*b*(α*exp[αt] +b*exp[-bt]/α +b) a>0,b>0,α >0
Yamada Raleigh	m(t)=a(1-exp[-rα (1-exp[-βt^2/2])]) a>0,r>0,α >0, β>0 a= total number of fault to be detected α = fault introduction rate r, β =constants	$\lambda(t)$=araβt(exp[-rα(1-exp[-βt^2/2])])*exp[-βt^2/2] a>0,r>0,α >0, β>0
Modified Duane Model	m(t)=a{1-(b/b+t)^c)} a>0,b>0,c>0 a = total number of fault to be detected	$\lambda(t)$=acb^c (b+t)$^{-(1+c)}$, a>0,b>0,c>0
Weibull-Type Testing-Effort Function Model	m(t)=a(1-exp[-ba(1-exp{-βt$^\gamma$}])]) a,b,α, β, γ >0 a=total number of fault to be detected b = fault detection rate α =Total number of test effort β = scale parameter γ = shape parameter	------------

NHPP Models:- NHPP Models have been successfully used in studying hardware reliability problems. These are also termed as fault counting models and can be either finite failure or infinite failure models, depending upon how they are specified. In these models the number of failures experienced or examined so far follows NHPP distribution. NHHP class of models is close relative of the homogenous Poisson model. In class of models difference is that here the expected failures is allowed to vary with time.[5],[10],[9]

Basic Assumptions of NHHP Models

There are some of the basic assumptions (apart from some

special ones for the specific models) assumed for NHHP models are as follows:[5]

- The software system is subjected to failure when execution caused by faults remaining in the system.
- Amount of fault detected at any time is proportional to the number of faults remaining in the software.
- The software failure rate is equally affected by fault remaining in the software.
- When failure occurs, repair efforts starts and fault causing failure is removed with certainty.
- According to failure detection point of view all faults are mutually independent.

- All the proportionalities regarding failure occurrence/fault isolation/fault removal is constant.
- There exists an equivalent fault detection/fault removal at user/manufacturer end corresponding to the fault detection/removal phenomenon at the manufacture/ user end.
- NHPP modelled the phenomenon of fault detection/removal.

Some of the assumptions may not hold their ground.

Comments on using NHHP Models

NHHP models are widely practitioners. The application of NHHP to reliability analysis can be found in elementary literature on reliability. Estimates of the parameters are easily obtained by using either the method of maximum likelihood estimation (MLE) or least squares estimation (LSE).[5]

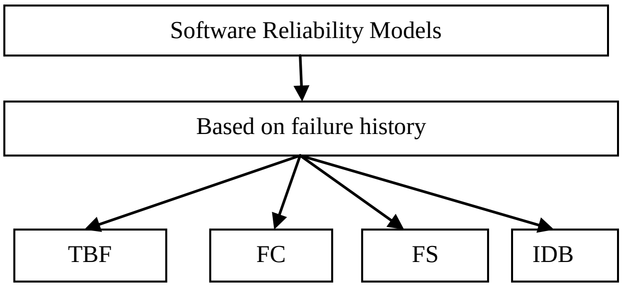

Figure 2 : Classification of Models

D. Classification Based on Failure History

The existing SWRMs are classified into four main classes on the basis of failure history.[7],[14]

- Time between Failure Models (TBF Models)
- Fault Count Models (FC Models)
- Fault Seeding Models (FS Models)
- Input domain based Models (IDB Models)

TBF Models:- In this class of models; process under study is the time between failures. It assumed that the time between (i-1)th and ith failures is a random variable. Estimates of these parameters are obtained from the observed values of TBFs and the parameter of SWR are obtained from the fitted models.[7]

FC Models:- The random variable of interest is the number of faults (failures) occurring during specfied time intervals is called FC Models. It is assumed that failure counts follow a known Stochastic process. The time is used it amy be calender time or may be CPU time.[14]

FS Models:- In this model we tested and observed number of seeded and indigenous fault is counted. The method of MLE and combinatory an estimate of the fault content of the program prior to seeding prior to seeding is obtained and then from the value of SWR parameter are computed.[7]

IDB Models:- In this model approach, a set of test cases is generated from the input covering the opertional profile of the input. Input domian is partitioned into set of equivalent classes.[14]

E. Classification Based on Data Requirements

- Empirical Models.
- Analytical Models.

Empirical Models:- Empirical SWR model used develops relationship or a set of relationship between SWR measures and a suitable software metrics such as program complexity using empirical results available from past data.[7][14]

Analytical Models:- This model requires some form of data gathered from software failure. It always based on fitting of suitable distrubution with required assumptions for simplicity on aset of data gathered during software testing.

Table 5: Overview of Models based on Failure History

Time Between Failure (TBF) Models	1. It's a independent times between failure.
	2. Each fault has equal probability.
	3. After each occurrence fault are removed.
	4. At the time of correction new faults are introduce.
	5. Ex. J-M De-Eutrophication, Schnick and Wolverton, Goel and Okumoto Imperfect Debugging, Littlewood-Verall Bayesian Models
Fault Count (FC) Models	1. Fault or failure in specified time interval.
	2. Testing during intervals is reasonably homogenous.
	3. Numbers of fault detected during non-overlapping intervals are independent of each other.
	4. Estimate software reliability mean time by fault count.
	5. Ex. Generalized Poisson Model, Goel-Okumoto NHPP Model, IBM Binomial and Poisson Models, Logarithmic Poisson Execution Time Model, Musa Okumoto
Fault Seeding (FS) Models	1. A known number of faults are "seed".
	2. Seeded faults are randomly distributed in the program
	3. A Program has unknown number of indigenous faults.
	4. Indigenous and seeded faults have equal probabilities of being detect.
	5. Ex. Lipow model, Mills seeding model, Basin model.
Input Domain Based (IDB) Models	1. Test cases are generated from the input covering.
	2. Estimate software reliability by failure observed in test cases.
	3. Random testing is used.
	4. Input domain can be partitioned into equivalent classes.
	5. Input profile distribution is known.
	6. Ex. Bastani Model, Nelson Model, Ramamoorthy.

3. FUTURE SCOPE

"The next step will be to analyse the software reliability using metrics along with the implementation of hybrid model using old techniques of reliability models".

4. CONCLUSION

In this paper various software reliability models are identified used in software engineering to calculate the reliability of software. Software Reliability has become more and more important in determining the capability of software within the time period. In this paper various Software Reliability Models are studied on the basis of dimensions along with their working, advantages and disadvantages .In the end many other parameters have been discussed which help in determining the reliability of software that are known as metric

5. ACKNOWLEDGMENTS

I would like to take this opportunity to express my gratitude to all those people who have in various ways helped us including the college students. My sincere thanks goes to Dr. Saurabh Mukherjee(Associate Professor,Department of Computer Science,Banasthali Vidhyapeeth) for sharing their wisdom, knowledge and expertise to better understand the issues related to this work.

6. REFERENCES

[1] Alan Wood, September 1996,"Software Reliability Growth Model", Part Number:130056.

[2] Chin-Yu Huang, Sy-Yen Kuo, June 2007, " An Assessment of Testing – Effort Dependent Software Reliability Growth Mondels ", IEEE Transaction Vol-56, No-2, pp- 198-211.

[3] A.Yadav & R. A. Khan, Nov 2009 "Critical Review On Software Reliability Models",Int. J. Recent Trends in Engineering and Technology, Vol -2, No. 3, pp-114-116.

[4] P. K. Kapur , H. Pham, Sameer Anand and Kalpana Yadav,March 2011," A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation",IEEE Transaction, Vol-60, No.1, pp-331-340.

[5] Richard Lai, Mohit Garg, June 2012,"A Detailed Study of NHPP Software Reliability Models",Jouranal of Software ,Vol.7, No.6, pp-1296-1306.

[6] Gaurav Aggarwal,Dr.V. K Gupta,January 2014 "Software Reliability Growth Model",International Journal of Advanced Research in Computer Science and software engineering, Vol.4, Issue.1,pp- 475-479.

[7] Razeef Mohd,Mohsin Nazir,September 2012,"Software Reliability Growth Models: Overview and Applications",Journal of Emerging Trends in Computing and Information Sciences,Vol.3, No.9, pp-1309-1320.

[8] Mohd. Anjum, Md. Asraful Haque, Nesar Ahmad,January 2013 "Analysis and Ranking of Software Reliability Models Based on Weighted Criteria Value",I.J.Information Technology and Computer Science, pp-1-14.

[9] Dr. R. Satya Prasad, Shaheen G. Krishna Mohan , October 2013,"Two Step Approach For Software Reliability :HLSRGM",International Journal Of Computer Trends and Technology,Vol-4,Issue.10,pp-3575-3579.

[10] B.Anniprincy & Dr.S.Sridhar ,2014 "Two Dimasional Software Reliability Growth Models Using Cobb-Douglas Production Function and Yamada S-Shaped Model", Journal Of Software Engineering and Simulation ,Vol.2,Issue.2,pp-01-11.

[11] Swapna S.Gokhale & Kishor S.Trivedi , December 2006,"Analytical Models For Architecture-Based Software Reliability Predication: A Unification Framework", IEEE Transactions, Vol.55, No.4, pp-578-590.

[12] Hoang Pham, Xuemei Zhang, 2003"NHPP Software Reliability and cost models with testing coverage ", ELSEVIER ,pp.443-454.

[13] Katerina Goseva-Popstojanaova, Kishor S. Trivedi, 2001"Architecture based approach to reliability assessment of software systems", ELSEVIER, pp-179-204.

[14] Latha Shanmugam , DR. Lilly Florence, October 2012"An Overview of Software Reliability Models ",Vol.2,Issue.10,pp-36-42.