ISSN: 2321-7782 (Online)
Volume 2, Issue 7, July 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A New Classification and Applicability of Software Reliability Models

Kuldeep Singh Kaswan[1]
Research Scholar, Department of Computer Science
Banasthali Vidyapith, Banasthali
Rajasthan – India

Sunita Choudhary[2]
Department of Computer Science
Banasthali Vidyapith, Banasthali
Rajasthan – India

Kapil Sharma[3]
Department of Computer Science & Engineering
Delhi Technological University
Delhi – India

Abstract: ***Today everyone demands for error free software, as we have already seen that even a single software error can results in a terrific loss and damage for life. It is the significance of software reliability in software development that today there are a number of reliability models available in software engineering. These models are classified into different categories and their classifications, are based on different parameters such as failure history, data requirements, number of failure counting, probabilistic assumptions, software structure, nature of debugging strategy, software development life cycle (SDLC) etc. These classifications have an important role in software reliability modeling. In this paper, we have discussed the existing classifications of software reliability models and further purposed a new classification scheme based on applicability of software reliability models according to the software development life cycle including the maintenance phase, which plays a critical role in software development now a days in the competitive world.***

Keywords: Software Reliability models, empirical models, Black box, White box, SRGM, SDLC, NHPP.

I. INTRODUCTION

Software engineering is an engineering discipline that is concerned with all aspects of software production. According to IEEE definition Software Engineering: is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.

Software reliability, an important attribute of software quality, can be defined as the probability of failure-free software operation for a specified period of time in a specified environment [1], [2], [3]. During different phases of software development life cycle software failures are introduced by the system analysts, designers, programmers and managers. To detect and remove these errors, the software system is tested. The quality of software system in terms of reliability is measured by the removal of these errors.

This paper is organized as follows: section II includes need of classification, existing classifications and potential gap between these classifications. In section III a new classification as per Software Development Life Cycle is introduced. Applicability of these software reliability models as per SDLC is given in section IV. Finally, the paper concludes with section V.

II. Related Work

Software reliability models have been used for a long time and have an interesting history. Till today different models have been proposed, discussed, modified and generalized while some models have also suffered from a lot of critiques. In this section we have reviewed the existing software reliability classification, its need and potential gap between the existing classifications.

Need of classification scheme

1. Software reliability models classification is useful in comparison of different reliability models
2. Different sets of models make it easy to obtain new models which are more realistic than the existing ones by identifying the unrealistic assumptions made for these existing models.
3. Help managers/management to select a group of software reliability models based on their requirement. This classification empowers managers to reject more than 80% of software reliability models as per the requirement and choose the particular model out of 20% remaining models.

Existing Software Reliability Models classification

Based on SDLC, Ramamoorthy and Bastani (1982) classified the software reliability models based on the software development life cycle on which these models are applicable. A model can be classified as development phase models, validation phase, operational maintenance phase models and correctness measure models [4].

According Musa (1984) the software reliability models based on the number of failure counting are divided into two categories as finite failure and infinite failure models [5]. Exponential, Weibull and Pareto software reliability models are finite failure category models while Geometric, Inverse Linear, Inverse Polynomial and power are family of infinite failure category models.

Ramamoorthy and Bastani (1986) classified the software reliability models according to the nature of debugging strategy. Another classification based on fault counting or non-fault counting models is also introduced by Bastani and Ramamoorthy.

According to Mellor (1987) the software reliability models have been classified according to software structure into different classes.

Based on probabilistic assumptions M. Xie divides the models into dynamics aspects and without dynamics aspects models. Dynamics describes the dynamics aspects of failure occurrence process. Markovian Models, NHPP and models based on Bayesian analysis are examples of dynamics aspects of failure occurrence, while the input domain models, fault seeding models and software metrics models are without dynamics aspects. [6]

Razeef and Nazir (2012) considered the four classes of the software reliability models as design phase models, Unit Testing phase models, Integration testing phase models, Acceptance testing and Operational Phase models [4]. A reliability model can be applied on more than one phase of software development life cycle.

Based on failure history, Shanmugam (2012) divide the software reliability models into four groups as time between failure models, Fault count models, Fault seeding models and Input domain based models. [7] Based on data requirements, Shanmugam also classified the models in two main groups as Empirical models and Analytic models. Miranda Models, Hallstead Models and Schneider Models are the example of empirical models. Analytical models further can be divided into two groups as Static models and Dynamic models.

Potential Gap in existing reliability models classification

1. As software reliability world is growing very fast so we need time to time updation in the existing reliability models.
2. Maintenance phase models not included in the existing software reliability classifications.

III. Purposed Classification as per Software Development Life Cycle

In this section we have purposed a new classification of software reliability models as per the software development life cycle [26] with the maintenance phase, as shown in Figure -1.

a) Early prediction Models

Early software reliability prediction is the prediction before testing in the absence of failure of failure data. Early reliability prediction attracts both software developers as well as managers as it provides an opportunity for the early identification of software quality, optimal development strategies and cost overrun [8].

b) Architecture Based Models

In this approach software is examined based on the design requirements and classified into different architectural styles. Overall system reliability can be measured by examined each individual architectural style. [9]

c) Mixed White-Box Models

It is the combination of early prediction models and architecture based models.

d) Mixed Black-Box Models

These models are the combination of software reliability growth models and input domain based models.

e) Software Reliability Growth Models

This model is applicable during the testing phase of software development and quantifies the software reliability in terms of estimated number of software error remaining in software or estimated time intervals between software failures. Software reliability growth is defined as the phenomenon that the number of software errors remaining in the system decreases with the progress of testing. For software growth modeling and analysis, the calendar time or CPU time is often used as the unit of software error detection period. [8]

f) Input Domain Based Models

An input-domain based model considers the software input space from which test cases are chosen and the studied quantity is the probability that a randomly chosen input datum according to the operational profile, will lead to a failure. [9]

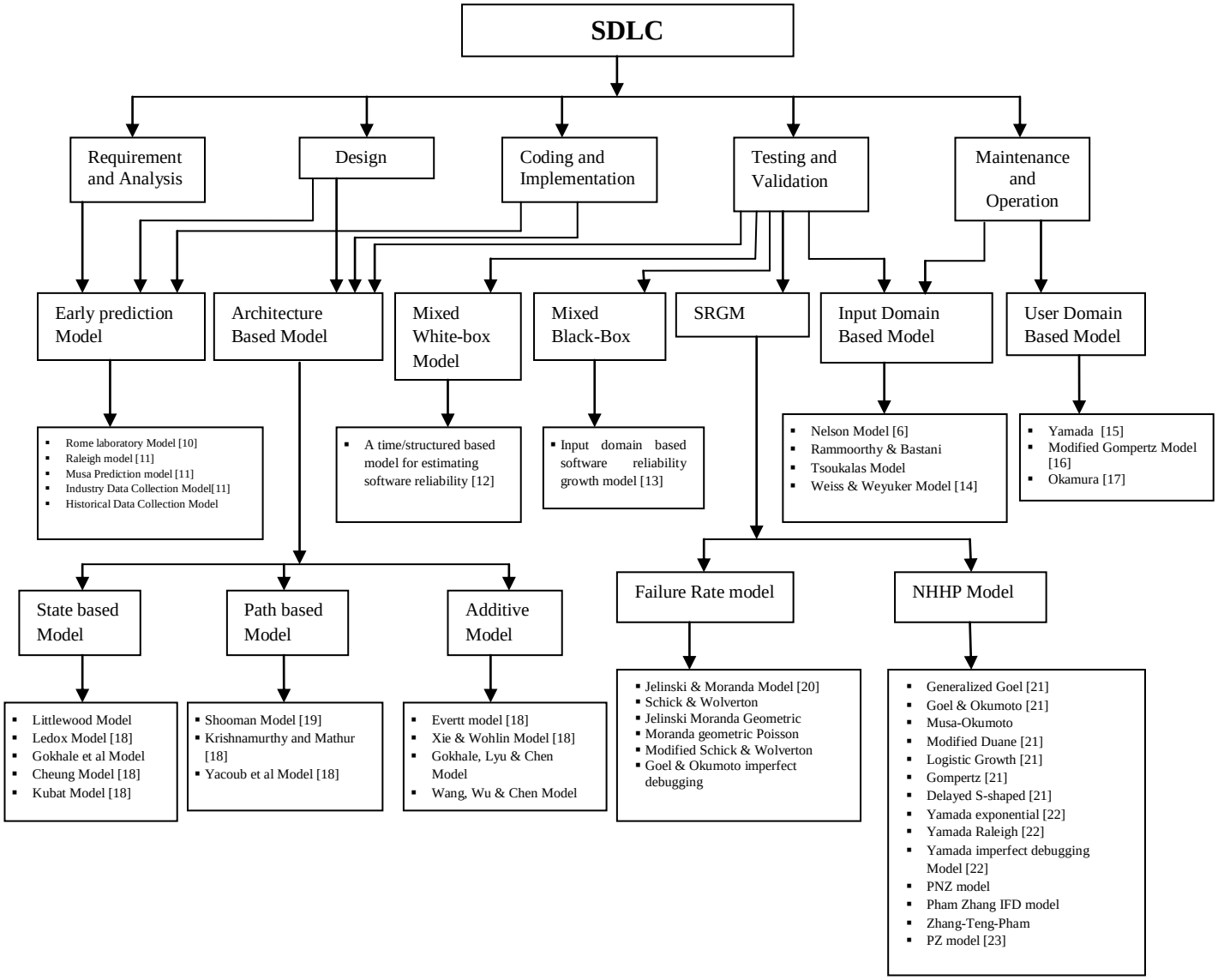

Fig.1 Software reliability models selection based on Software Development Life Cycle

g) User Domain Based Models

The software reliability in maintenance phase exhibits many types of systematic or irregular behaviors. These may include cyclic behavior as well as long-term evolutionary trends. The cyclic behavior may involve multiple periodicities and may be asymmetric in nature. [4]

IV. Applicability of Software Reliability Modeling as per Software Development Life Cycle

Requirement and Analysis Phase Models

These models applicability was proved by Rome Laboratory RL-TR-92-52. For this data was collected from a variety of sources, including the Software Engineering Laboratory.

In this experiment, 33 data sources represent 59 different projects, which include 9 factors: Application, Development Organization, Software anomaly management, Software traceability, Software quality, Software language, Software complexity, Software modularity and Software standards review, which are used to predict the fault density of the software application [24].

Design Phase Models

Wen-Li Wang [19] estimates architecture-based software reliability by designing a model, in this model we assess the reliability of each component, the operational profile, and the architecture of software, to validate the correctness of this model a simulator of an ATM bank system is utilized.

Coding and Implementation Phase Models

Both requirement and analysis phase models, and design phase models can be applicable on this phase.

Testing and Validation Phase Models

The input domain method was used for a preliminary estimation of the reliability of software used in a space application. The reliability estimation was limited to a subset of the system software modules due to the effort required by the application.

In particular we evaluated only those modules that are more critical for the instrument functionality, and examined only the usual operational condition, without considering software for diagnostic & failure containment. In practice, the reliability estimation was limited to 2.4 KLOC out of the 4.9 KLOC of the whole system [25].

Maintenance Phase models

Chen-Tsai-Chen [12] proves the applicability of proposed maintenance phase model through a numerical for which real tracking database records of a shop floor control system utilized in an electronic factory.

Yamada [15] assesses the software product reliability in the maintenance phase by developing a new method based on the empirical knowledge.

Okumura [17] also propose a new method based on accelerated life testing for software product reliability assessment in maintenance phase.

V. CONCLUSION

In this paper we discuss about the need of software reliability models classification, existing classifications, its drawbacks and need of a new classification scheme. Then we proposed a new classification of reliability models and discussed the applicability of these models on different phases of software development life cycle with maintenance phase.

References

1. Kapur, P.K. and Garg, R.B., "Cost reliability optimum release policies for a software system with testing effort, " Operations Research, Vol. 27, No. 2, pp. 109-116, 1990.
2. Musa J.D., Software Reliability Engineering: More Reliable Software, Faster Development and Testing, McGraw-Hill, 1999.
3. Musa, J.D., Iannino, A. and Komodo, K., Software Reliability: Measurement, Prediction and Application, McGraw-Hill, 1987.
4. C. V. Ramamoorthy and Farokh B. Bastani, "Software Reliability-Status and Perspectives", in IEEE Transactions on software engineering, Vol. SE-8, No. 4, July 1982.
5. J. D. Musa and K. Okumoto, "A Logarithmic Poisson Execution Time Model for Software Reliability Measurement", 0270-5257/84/0000/0230501.00© 1984 IEEE.
6. M. Xie, "software reliability modeling", World Scientific Pub. 1991
7. Latha Shanmugam, Dr. Lilly Florence, "In Overview of Software Reliability Models", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, Issue 10, Oct. 2012
8. Ajeet Kumar Pandey, Neeraj Kumar Goyal, "early Software reliability prediction: A fuzzy logic approach", ISBN 978-81-322-1175 4(print)
9. Wen-Li Wang. Dai Pan. Mei-Hwa Chen, "Architecture-Based Software Reliability Modeling", Computer Science Department. Suny Albany
10. Yung-Chung Chen, Shih-Ying Tsai and Peter Chen, "Reliability Models for Application Software in Maintenance phase", IEMS Vol. 7, No. 1, pp. 51-56, 2008.
11. C. Smidts, R. W. Stoddard, and M. Stutzke, "Software reliability models: An approach to early reliability prediction," IEEE Trans. Reliability, Vol. 47, No. 3, pp. 268–278, 1998.
12. S. S. Gokhale and K. S. Trivedi, "A time/structure based software reliability model", Analysis of Softw. Engineering, Vol. 8, pp. 85–121, 1999.
13. S. S. Gokhale, P. N. Marinos, and K. S. Trivedi, "Important milestones in software reliability modeling", Proc. of Softw. Engineering and Knowledge Engineering (SEKE '96), Lake Tahoe, NV, pp. 345–352, 1996
14. S. N. Weiss and E. J. Weyuker, "An extended domain-based model of software reliability", IEEE Trans. Software Engineering, Vol. SE-14, pp. 1512–1524, 1988.
15. Yamada, S., Tanio Y., and Osaki, S., "A software reliability evaluation method during operation phase", Trans IEICE, J72-D-I, 797-803, 1989.
16. Vladimir Zeljkovie, Nela Radovanovie and Dragomir Ilie, "Software Reliabilty: Models and Parameters Estimation", Science Technical Review, Vol. 61, No 2, pp. 57-60, 2011.

17. Okamura, H., Tadashi, D., and Shunji, O., "A reliability assessment method for software products in operational phase-Proposal of an accelerated life testing model", Electronics and Communications in Japan, Part 3, 84(8), 25-33, 2001.

18. K. Popstajanova and K. Trivedi, "Architecture based approach to reliability assessment of software systems," Performance Evaluation, Vol. 45, No. 2, June 2001

19. W. Wang, Y. Wu, and M. H. Chen, "An architecture-based software reliability model", in Proc. of Pacific Rim Dependability Symposium, Hong Kong, December 1999.

20. Z. Jelinski and P. B. Moranda, "Software Evaluation", W. Freiberger, Ed. New York: Academic Press, 1972.

21. C. Y. Huang, M. R. Lyu, and S. Y. Kuo, "A unified scheme of some non-homogenous Poisson process models for software reliability estimation", IEEE Trans. Softw. Engineering, Vol. 29, No. 3, pp. 261–269, March 2003.

22. H. Pham, "Software reliability and cost models: perspectives, comparison and practice", European J. of Operational Research, Vol. 149, pp. 475–489, 2003

23. H. Pham and X. Zhang, "An NHPP software reliability models and its comparison", International J. of Reliability, Quality and Safety Engineering, Vol. 14, No. 3, pp. 269–282, 1997.

24. McCall, W. Randell, J. Dunham, L.Lauterbach, "Software Reliability, Measurement, and Testing Software Reliability and Test Integration RL-TR-92-52", Produced for Rome Laboratory, Rome, NY, 1992

25. Alberto Pasquini, Elio De Agostino and Giuseppe Di Marco, "An Input-Domain Based Method to Estimate Software Reliability", IEEE Transactions on Reliability, Vol. 45, No. 1, 1996.

26. Kapil Sharma, Rakesh Garg, C. K. Nagpal, and R. K. Garg, "Selection of Optimal Software Reliability Growth Models Using a Distance Based Approach", IEEE Transactions on Reliability, Vol. 59, No. 2, June 2010

AUTHOR(S) PROFILE

Kuldeep Singh Kaswan was born in Haryana, India in 1982. He is pursuing a Doctors Degree in Computer Science under the faculty of Computer Science at Banasthali Vidyapith, Rajasthan. He has obtained his Master Degree in Computer Science and Engineering from Choudhary Devi Lal University, Sirsa (Haryana) and Bachelor Degree from Kurukshetra University, Kurukshetra (Haryana). His area of interests includes software engineering and reliability.

Sunita Choudhary was born in Rajasthan, India in 1978. She received Doctors Degree in Computer Science under the Faculty of Computer Science at the Banasthali University (Rajasthan), India. She has obtained his Bachelor of Science in Computer Science and Master of Computer Application Degrees from Banasthali University in 1999, and 2001 respectively. Presently, she is working as an associate professor in Department of Computer Science at Banasthali University, Rajasthan, India. Her research interests include Algorithms and Distributed Computing.

Kapil Sharma was born in Haryana, India in 1977. He received Doctors Degree in Computer Science and Engineering under the Faculty of Engineering and Technology at the M. D. University, Rohtak (Haryana), India. He has obtained his Bachelor of Engineering and Master of Technology Degrees in Computer Science & Engineering and Information Technology from M. D. University, Rohtak, and IASE, Rajasthan, India in 2000, and 2005 respectively. Presently, he is working as an associate professor in Department of Computer Science at Delhi Technological University, Delhi, India. His research interests include software engineering, and reliability.

