

## **Survey on Various Prototype used in Building Software Reliability Model**

**D. Swamydoss and Dr. G.M. Kadhar Nawaz**

*Adhiyamaan College of Engineering, India  
Sona College of Technology, India  
E-mail: swamy\_asir@yahoo.co.in*

### **Abstract**

Software reliability engineering is focused on engineering techniques for developing and maintaining software systems. In order to measure and predict the failure in the system, there are many models has available in the IT world. But all the models are developed based on the application. All the models have follows its own prototype and structure. Though software reliability has remained an active research subject over past 30 years still challenges and issues are exist. There is no common prototype available to build a reliability tool. This paper discussed about different structure used in different reliability models.

**Keywords:** Software Reliability, Basin Model, Moranda model.

### **Software Reliability Engineering**

Reliability is probably the most important of the characteristics inherent in the concept. It concerns itself with how well the software functions to meet the requirements of the customer. Reliability represents a user-oriented view of software quality. The definition of software reliability is the probability of failure-free operation of a computer program for a specified time in a specified environment. Reliability is a much wider measure. It is customer oriented rather than developer oriented. It relates to operation rather than design of the program hence it is dynamic rather than static. Reliability measures are much more useful than fault measures. This does not mean that some attention to faults is without value but the attention should be focused on faults as predictors of reliability and on the nature of faults. Understanding software reliability measurement and prediction has become a vital skill for both the software manager and software engineers.

## **Software Reliability Models**

Work on software reliability models started about twenty years back. Some probabilistic models based on various assumptions and statistics have been developed. Reliability design begins with development of models. On the basis of failure history the existing software reliability models can be grouped into four categories;

- Time between failure models
- Fault count models
- Fault seeding models
- Input domain based models

The time between failure models is assumed that the time between (i-1) and ith failures is a random variable, following a distribution whose parameters depends on the number of faults remaining in the program during this interval. In the fault count models, the random variable of interest is the number of faults occurring during specified time intervals. It is assumed that failures counts follow the same known stochastic process, usually a Poisson distribution with a time dependent discrete or continuous failure rate. In fault seeding model, a known number of faults are seeded. The program is tested and observed numbers of seeded and indigenous fault are counted. Typical models of this class are Mills seeding models, Lipow model and Basin model. In input domain based models, a set of test cases is generated from the input, covering the operational profile of the input. Usually the input domain is partitioned into a set of equivalent classes, each of which is usually associated with a program path.

On the basis of data requirements, the software reliability model can be grouped into two main groups; empirical models and Analytical models. The structure of empirical model requires identification of the appropriate software reliability metrics and the development of the right type and form of relationships between the metrics and reliability measures. Models of this type are Moanda model, Holstead model and Schneider model. Analytical models require some form of data gathered from software failures. It is based on fitting of a suitable distribution with required assumption for simplicity on a set of data gathered during software testing and prediction of software reliability parameters from the fitted distribution. It involves four steps;

- Defining the assumptions associated with a software test procedure.
- Developing an analytical model based on the assumption and the test procedure.
- Using the model for performance prediction
- Obtaining parameter for the model using collected data.

Analytical model subdivided into static models and dynamic models based on time depend of collected data. The software reliability models proposed range from simple empirical ones to the complex ones like Littlewood and Verrall Model, where extensive use of Bayesian distribution and Gamma distributions have been made. The failure process, itself is very complex involving interaction of human factors, program logic and input and output made in the design of software reliability model to

approach to the real operational environment as possible. There are about more than 20 reliability model proposed by software experts, but all the models have its own structure. There is no common prototype followed in the models.

## Conclusion

In this paper we have discussed the structure of different types of software reliability model. All the models has been designed based on a specified application. No model has dealt with the distributed environment of the software systems. Since, now a day all the software applications are based on web and heterogeneous environment. So in the reliability model system environment is considered as an important parameter. In our work we have mainly focus on the various structure followed to construct a software reliability model.

## References

- [1] ANSI/IEEE, Standard Glossary of Software Engineering Terminology, *STD-729-1991*, ANSI/IEEE, 1991.
- [2] J.D. Musa, *Software Reliability Engineering: More Reliable Software Faster and Cheaper (2nd Edition)*, AuthorHouse, 2004
- [3] B. Littlewood and L. Strigini, "Software Reliability and Dependability: A Roadmap," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE'2000)*, Limerick, June 2000, pp. 177-188.
- [4] M.L. Shooman, *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis and Design*, Wiley, New York, 2002
- [5] M. Chen, M.R. Lyu, and E. Wong, "Effect of Code Coverage on Software Reliability Measurement," *IEEE Transactions on Reliability*, vol. 50, no. 2, June 2001, pp.165-170.
- [6] Y.K. Malaiya, N. Li, J.M. Bieman, and R. Karcich, "Software Reliability Growth with Test Coverage," *IEEE Transactions on Reliability*, vol. 51, no. 4, December 2002, pp. 420-426
- [7] M.A. Vouk, "Using Reliability Models During Testing With Nonoperational Profiles," in *Proceedings of 2nd Bellcore/Purdue Workshop on Issues in Software Reliability Estimation*, October 1992, pp. 103-111.
- [8] Rome Laboratory (RL), *Methodology for Software Reliability Prediction and Assessment*, Technical Report RL- TR-92-52, volumes 1 and 2, 1992.
- [9] C. Liu, L. Fei, X. Yan, J. Han, and S. Midkiff, "Statistical Debugging: A Hypothesis Testing-based Approach," *IEEE Transaction on Software Engineering*, vol. 32, no. 10, October, 2006, pp. 831-848.
- [10] A.X. Zheng, M.I. Jordan, B. Libit, M. Naik, and A. Aiken, "Statistical Debugging: Simultaneous Identification of Multiple Bugs," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006, pp. 1105-1112.