

A review on software project reliability estimation

Sanjana and P.K. Suri

Department of Computer Science and Applications, Kurukshetra University, Kurukshetra, Haryana, India.

ARTICLE INFO

Article history:

Received: 8 September 2011;

Received in revised form:

15 November 2011;

Accepted: 24 November 2011;

Keywords

Reliability estimation,

Software Quality,

Software Reliability,

Software Reliability Engineering.

ABSTRACT

Despite the increased importance of the role that reliability plays in commercial product development, most companies are still unable to produce reliable software. The mission of organizations is to provide innovative products at a competitive price to fulfill the needs of very high expectations highly reliable software products for its customers and to do so ahead of its competitors. This requires focus on customer's satisfaction and software reliability to ensure that desired factor is built into the software product and enable reliable software to be produced. By optimizing best practices for defect removal, organizations can produce high reliability software. In this paper we focus on some research efforts directed at gaining a deeper understanding of the reliability estimation using different algorithms, approaches, simulation methods for software.

© 2011 Elixir All rights reserved.

Introduction**Software Reliability**

To achieve improvements, reliability must be defined and measured. Reliability is probably the most important factor to claim for any engineering discipline, as it quantitatively measured quality and quality can be properly engineered. Product development team view reliability as the domain to address mechanical and electrical and software issues. Customers view reliability as a system level issue with minimal concerned placed on the distinction into sub domains. Since the primary measure of reliability is made by customers, engineering team must make a balance of both view system and sub domain in order to produce a reliable product.

Software Reliability is an important facet of software quality. Software reliability is defined as the probability of failure-free software operation for a specified period of time in a specified environment. While Software quality measures how well software is designed (quality of design), and how well the software conforms to that design (quality of conformance). It is often described as the 'fitness for purpose of a piece of software. Software reliability is an attribute and key factor in software quality. It is also a system dependability concept. Software Reliability Engineering (SRE) is defined as the quantitative study of the operational behavior of software-based systems with respect to user requirements concerning reliability. SRE employs proven best practice to ensure that product reliability meets user needs, to speed products to market faster, reduce product cost, improve customer satisfaction, and increase tester and developer productivity. One of reliability's distinguishing characteristics is that it is objective, measurable, and can be estimated, whereas much of software quality is subjective criteria. This distinction is especially important in the discipline of Software Quality Assurance.

Software Reliability Engineering

Software Reliability Engineering is an emerging discipline that is concerned with improving an approach to software Reliability. The function of reliability engineering is to develop the reliability requirements for the product, establish an adequate reliability program, and perform appropriate analyses and tasks

to ensure the product will meet its requirements. The relationship between software defects and failures is not one-to-one. Some defects remain undiscovered and never cause a failure, but a single defect can cause many some failures. Many researchers have offered solutions to this problem, but their approaches typically reflect developers view point of reliability. Work done by various researchers in the field of software reliability estimation is surveyed below:

Mrs. J Emi Retna et. al. [1] provided insights in defining semantically rich annotations to source code using Tags for Software Engineering Activities tool and to improve navigation and management of annotations while estimating the reliability of the tool. Reliability is one of the illusive targets to achieve in the software development for the successful software projects. It is one of the most important parameter or attribute of software to be achieved for the software quality. There are different techniques and models used for estimating the reliability of the software. Architecture-based approach for estimating the reliability is used.

Marko Palviainen et. al. [2] addressed software reliability evaluation during the design and implementation phases; it provides a coherent approach by combining both predicted and measured reliability values with heuristic estimates in order to facilitate a smooth reliability evaluation process. The approach contributes by integrating the component-level reliability evaluation activities and the system-level reliability prediction activity to support the incremental and iterative development of reliable component-based software systems.

Shelbi Joseph et. al. [3] proposed an algorithm for determining reliability of software based systems by integrating both hardware reliability and software reliability. The open source software data is made use of and the methodology for evaluating the software reliability involves identifying a fixed number of packages at the start of the time and defining the failure rate based on the failure data for these preset number of packages. The defined function of the failure rate is used to arrive at the software reliability model. The hardware reliability is obtained using constant hazard model.

Mou Dasgupta et. al. [4] proposed two linear time complexity algorithms for approximate assessment and the enhancement of the reliability of the given networks. The proposed techniques basically identify the node-pairs having lower reliability, insert communication links in them and calculate the increase in reliability on insertion iteratively until the satisfactory reliability is achieved. The simulated experimentation of the proposed algorithms have been done and compared with the existing methods, which show satisfactory performance.

Wei-Chang Yeh et. al. [5] analyzed three Monte Carlo Simulation (MCS) methods for estimating the two-terminal network reliability of a binary-state network. MCS1 simulates the network reliability in terms of known Minimal paths. MCS2 estimates the network reliability in terms of known minimal cuts. CAMCS based on cellular automata estimates the network reliability directly without knowing any information of Minimal Paths or Minimal Cuts. simulation results show that the direct estimation without knowing any information of Minimal Paths or Context can speedup about 185 times when compared with other traditional approaches which require minimal paths or cuts information.

Wei-Chang Yeh et. al. [6] proposed Multi-state Node Networks that are composed of multi-state nodes with different states determined by a set of nodes receiving the signal directly from these nodes without satisfying the conservation law. Current methods for evaluating multi state node reliability are all derived from Universal Generating Function Methods. The proposed method allows cycles with the same time complexity as the best-known method. The correctness and computational complexity of the proposed UGFM are analyzed and proven.

S Chatterjee et. al. [7] proposed a sequential Bayesian approach for estimating reliability growth or decay of software. It also showed the variation of parameter over a time, as new failure data become available. The usefulness of method is demonstrated with some real life data. P.K. Suri et. al. [8] proposed a simulator that has been attempted to compute the reliability of the system as a function of reliabilities of its components along a path called course-of-execution are executed during each simulation run. if the reliability of the component being purchased off the shelf, is known in prior, its effect on the overall application reliability can be computed and a decision can be made whether to incorporate that component in the system or search for a different alternative. As regard to the number of components in a system, if they are increased, the overall system reliability starts decreasing. It may help in deciding how much reliable software is desired.

Morris, R. et. al. [9] proposed the elements that affect software reliability and compares the differences when trying to estimate reliability of today's systems. Software reliability is characterized by data gathered during systems integration and test. Software reliability is more than the processes advocated by CMMI (Capability Maturity Modeled Integration) and is susceptible to esoteric and infinitely harder parameters to measure. Yu-Shen Su et. Al. [10] proposed an artificial neural-network-based approach for software reliability estimation and modeling and applied neural network to predict software reliability by designing different elements of neural networks. Furthermore, they used the neural network approach to build a dynamic weighted combinational model. The applicability of proposed model is demonstrated through real software failure

data sets. The results obtained show that the proposed model has a fairly accurate prediction capability.

Tumminello, M. et. al. [11] introduced a new technique to associate a spanning tree to the average linkage cluster analysis and also introduced a technique to associate a value of reliability to links of correlation based graphs by using bootstrap replicas of data. It showed that the average reliability of links in the Minimum Spanning Tree is slightly greater than the average reliability of links in the Average Linkage Minimum Spanning Tree. Jintao Xiong et. al. [12] proposed a rational approximation algorithm for network reliability estimation can be applied to estimate both the all-terminal reliability measure and the reliability of networks with various protection algorithms. Algorithm is very efficient and it can obtain the whole reliability function curve. Stephen Clark et. al. [13] proposed a technique for estimating the probability distribution of total network travel time. A solution method is proposed, based on a single run of a standard traffic assignment model, which operates in two stages and also discussed how the resulting distribution may be used to characterize unreliability.

Linda Rosenberg et. al. [14] described that metrics used early can aid in detection and correction of requirement faults that will lead to prevention of errors later in the life cycle. At each phase of the development life cycle, metrics can identify potential areas of problems that may lead to errors and the areas in the phase they are developed decreases the cost and prevents potential ripple effects from the changes later in the development life cycle.

Tongdan Jin et. al. [15] proposed an algorithm to approximate the terminal-pair network reliability based on minimal cut theory. The new model estimates the reliability by summing the linear and quadratic unreliability of each minimal cut set. Given component test data, the new model provides tight moment bounds for the network reliability estimate. Those moment bounds can be used to quantify the network estimation uncertainty propagating from component level estimates. Jones, E.L et. al. [16] proposed a simulator that fixes the fault whenever it is encountered during the simulation and resumes reliability testing. It provides a low-cost environment for learning reliability testing and modeling, and for exploring factors that contribute to the effectiveness of reliability testing. Whittaker et. al. [17] proposed reliability as a function of application complexity, test effectiveness and operating environment. The reward for doing so resulted in the adoption of software reliability prediction by majority.

Chat srivaree ratana et. al. [18] proposed artificial neural network predictive models for the estimation of all terminal network reliability, hierarchical approach are used i.e. a general neural network screens all network designers for reliability followed by a specialized neural network for highly reliable network designers result on a ten node problem are given using a group crossed decomposition approach. Nicholas Lynn et. al. [19] proposed joint prior distributions for the reliabilities of individual components. To simulate the reliability of network by using samples obtained as input to a network reliability calculation. Tausworthe et. al. [20] proposed a general simulation technique that relaxes or removes many of the usual reliability-modeling assumptions and expands the reliability process to encompass the entire software life cycle and hence emerged as the most effective predictor.

Mastran D et. al. [21] estimated the moments of coherent system reliability estimates based on attribute test data from

component level. Musa J.D. [22] discussed the needs and potential uses of software reliability measurement, leading software reliability models and the combination of software components and availability. This also concluded with an analysis of the current state of art in software reliability measurement.

Oliveira G.C. et. al. [23] proposed a novel technique for reducing the number of samplings in Monte-Carlo-based composite power system reliability evaluation. The proposed technique uses analytical information from simple models as regression variables to reduce the variance of loss-of-load probability and expected power not supplied estimates with the complete composite model. The higher speedups prevail in the overall convergence if a uniform criterion is adopted. Timothy B. Brecht et. al. [24] devised two strategies for obtaining lower bounds on the connection probability for two terminals. The first improves on the Kruskal-Katona bound by using efficient computations of small path sets and the second employs efficient algorithms for finding edge-disjoint paths. The resulting bounds are compared; while the edge-disjoint path bounds typically outperform the Kruskal-Katona bounds, they do not always do so. Finally a method is outlined for developing a uniform bound which combines both strategies. Natvig B et. al. [25] proposed a six different lower and upper moment bounds for two terminal network reliability estimate based on minimal cuts and minimal paths theory. These moment bounds assume all component reliability estimates, among and within the cut and path sets are statistically independent.

HECHT H et. Al. [26] proposed software reliability measurement was carried out during the development of data base software for a multi-sensor tracking system. Every run made during this project was scored as success or failure, and supporting data were collected for further analysis. The failure ratio and failure rate were found to be consistent measures and therefore considered valid indicators of reliability in this environment. Over one-half of the observed failures were due to factors associated with the specific run submission rather than with the code proper. Pitu B. Mirchandani [27] proposed an algorithm to compute the expected shortest travel time between two nodes in the network when the travel time on each link has a given independent discrete probability distribution. The algorithm assumes the knowledge of all the paths between two nodes and methods to determine the referenced paths. In reliability computations, associated with each link are a probability of "failure" and a probability of "success". Since "failure" implies infinite travel time, the algorithm simultaneously computes reliability.

Cavers, J et. al. [28] described the problem of calculating the reliability of a communication network of given topology. The definition of failure is generalized from simple disconnection to the event that terminal capacity falls below some threshold, and simple methods are developed for bounding the probability of such failures and for approximating the mean time between failures. It is shown that the dependence of network reliability on individual link reliability is substantially different when the generalized definition of failure is used.

Conclusion

This chapter has primarily focused on the work done by various researchers in the field of software reliability. Besides, various standardized reliability techniques along with their nomenclature also formed a part of the study. The crux of the matter which emerged is that every process has its own

particular environment and no reliability criterion is absolute in the sense that it may not be applicable universally to all the systems. Thus, shifts to new researchers to study every system or subsystem in its own entity and take the task to devise reliability controlling techniques specific to the system under study and not forgetting about its limitations to function equally efficiently under a different set of environment. This is a harsh truth which the researchers have to accept though with a heavy heart and must not jump to exceedingly high expectations from any single technique

References

- [1]Mrs. J Emi Retna, Sumy Joseph, Merlin Soosaiya "An Architecture based Approach for Reliability Estimation", International Journal of Computer sciences, January 2011.
- [2]Marko Palviainen, Antti Evesti and Eila Ovaska "The reliability estimation, prediction and measuring of component-based software", Journal, Volume 84, Issue 6, Jan 2011, Pages 1054-1070.
- [3]Shelby Joseph, Shouri P.V, Jagathy Raj V. P "A Model for Reliability Estimation of Software based Systems by Integrating Hardware and Software" Special Issue on Computational Science - New Dimensions & Perspectives (1):26-29, 2011.
- [4]Mou Dasgupta, G. P. Biswa "Reliability Measurement and Enhancement of the Communication Networks", International Journal of Computer Applications 1(9):18-25, February 2010, Number 9 - Article 3, 2010.
- [5] Wei-Chang Yeh, Yi-Cheng Lin and Yuk Ying Chung "Performance analysis of cellular automata Monte Carlo Simulation for estimating network reliability ", Expert Systems with Applications Volume 37, Issue 5, May 2010, Pages 3537-3544.
- [6] Weii-Chang Yeh "A simple universal generating function method for estimating the reliability of general multi-state node networks", IIE Transactions ,volume 41,issue 1,2009, pages 3-11.
- [7]S Chatterjee, S Alam and R B Mishra "sequential Bayesian Technique: an alternative approach for software reliability estimation", published in sadhana vol. 34, part2, April 2009.
- [8]P.K. Suri, Sandeep Kumar "design of simulator for reliability estimation of component based software system", International Journal of Computer Science and Network Security, vol.9, no.9, Sep 2009.
- [9]Morris R. "Software Reliability - 40 Years of Avoiding the Question", Software Reliability Engineering, 2008. ISSRE 2008, 19th International Symposium on 10-14 Nov. 2008.
- [10]Yu-Shen Su and Chin-Yu Huang "Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models", Journal of systems and software volume 80, Issue 4, April 2007, Pages 606-615.
- [11]Tumminello, M.Coronello, C.;Lillo, F.;Micciche', S.;Mantegn,R.N. "Spanning Trees and bootstrap reliability estimation in correlation Based networks", arxiv: physics/0605116 on May 2006 .
- [12]Jintao Xiong and Weibo Gong "A Novel algorithm on network reliability estimation" Mathematical and Computer Modeling Volume 41, Issues 2-3, January-February 2005, Pages 119-133.
- [13]Stephen Clark¹ and David Watling "Modeling network travel time reliability under stochastic demand", Transportation Research Part B: Methodological ,Volume 39, Issue 2, February 2005, Pages 119-140.

- [14]Dr. Linda Rosenberg, Ted Hammer, Jack Shaw Unisys/NASA GSFC Bid 6 Code 300.1 greenbelt,MD 2771 USA, Ted Hammer NASA GSFC “Software metrics and reliability” ,2004.
- [15]Tongdan Jin and David W. Coit “Approximating network reliability estimates using linear and quadratic unreliability of minimal cuts”, *Reliability Engineering and system safety* volume 82,Issue 1,October 2003, Pages 41-48.
- [16]Jones, E.L. Rembert, A.J. Comput. & Inf. Sci. Dept., Florida A&M Univ., Tallahassee, FL, USA “A simulation based trainer for software reliability modeling”, *Software Reliability Engineering*, 2001. ISSRE Proceedings. 12th International Symposium on 27-30 Nov. 2001 on page 160.
- [17]Whiittaker, J.A. Voas “towards a more reliable theory of software reliability” dec 2000.
- [18]Chat srivaree ratana and Alice E. Smith “Estimating All-Terminal Network Reliability Using artificial Neural Networks”, *International Journal of Computer Science*, 1999.
- [19]Nicholas Lynn, nozer singpurwalla and Adrian smith “Bayesian assessment of network reliability”, vol. 40, No.2, in june1998 pages 202-227.
- [20]Tausworthe, R.C. Lyu, M.R.; Jet “A generalized technique for simulating software reliability” *software IEEE* on Mar 1996.
- [21]Mastran D, Singpurwalla N.” a Bayesian estimation of the reliability of coherent structure” *Oper Res*, 26(4), pp.663-72, 1994.
- [22]Musa J.D. “the measurement and management of software reliability” proceedings of IEEE on sep. 1990.
- [23]Oliiveira, G.C.; Pereira, M.V.F.; Cunha,.H.F.; Centro de Pesquisas de Energia Electr., Rio di Janeiro “A technique for reducing computational effort in Monte-Carlo based composite reliability evaluation”, *power Systems, IEEE Transactions on* Nov 1989,Volume 4,Issue 4 Page 1309-1315.
- [24]Timothy B. Brecht and Charles J. Colbourn “Lower bounds on two-terminal network reliability”, *Discrete* Volume 21, Issue 3, October 1988, Pages 185-198.
- [25]Natvig B, Eide H.” Bayesian estimation of system reliability”, *Sc and J Stat*, 14, pages 319-27, 1987.
- [26]HECHT, H | STURM, W A | TRATTNER, S “Reliability Measurement during Software Development”, *computers in Aerospace Conference*, Los Angeles, Calif; United States; 31 Oct.-2 Nov. 1977. pp. 404-412. 1977.
- [27]Pitu B. Mirchandani”Shortest distance and reliability of probabilistic networks”, *Computers & Operations Research*, Volume 3, Issue 4, December 1976, Pages 347-355.
- [28]Cavers, J.; Carleton Univ., Ottawa, Ont., Canada “Cutest Manipulations for Communication Network Reliability Estimation”, *Communications, IEEE Transactions on* June 1975, and Issue: 6 page 569-575.