On

Modeling for Early Reliability Prediction of Software Systems

Thesis Submitted for the Degree of

Doctor of Philosophy

By

Lalit Kumar Singh

(Enrolment No. 323431)

Supervisors:

Prof. A.K. Tripathi

Dr. Gopika Vinod

Department of Computer Science and Engineering

Indian Institute of Technology

(Banaras Hindu University)

Varanasi-221005, India

2014

Chapter 7

Conclusion and Future Research

Traditionally, SGRM are used for reliability modeling of a software system. The limitations of SGRM are discussed and shown that these limitations can be addressed by providing a methodology to support SRE from requirements to deployment, with adequate analysis, through appropriate mappings. One of the prime potential benefits of software reliability early prediction is in taking preventive action to meet the target reliability of a software system. Before implementation phase of the software development, there are two phases: requirements analysis and software design. Researchers are continuing to propose approaches to predict the reliability of a software system in these two phases. Many of the existing models, during software design phase, are based on Markov Chain, but the uncertainties associated with the accuracy of the model, parameters, phenomenon and assumptions limit the practical usage. In this dissertation, four issues have been addressed: (i) Uncertainty in Markov reliability models (ii) Uncertainty in input parameters in Markov reliability models and their limitations (iii) Impact analysis of component reliability on the reliability of a software system for preventive action (iv) Updation of components reliability during testing/operational phase for corrective action The specific conclusions for the above mentioned four issues are highlighted in the following sections.

7.1 Uncertainty in Markov reliability models

There are different approaches available in the literature to predict the software reliability using MC model. A good reliability prediction model should include all the requirements, functional and non-functional, of the software system. Keeping accuracy of the reliability model, with respect to the requirement integrity, as a principal objective, UML has been extended for its creation. The purpose of using UML is to construct a reliability model with the involvement of all the stakeholders, especially the clients. The reason to choose the UML is its simplicity and ease to understand for all the stakeholders. The concept used in our approach is to predict software system reliability from scenario specifications by extending a scenario specification to model (1) the probability of component failure, and (2) scenario transition probabilities derived from an operational profile of the system. Cheung's approach has been followed to predict the reliability from the created Markov reliability model. The proposed methodology has been illustrated and validated on a safety critical system of a NPP. The detail description of the case study is described. The operational profile data of 1 year of the NPP system has been taken for validation. The sensitivity analysis to determine the impact of change in component reliabilities and change in the transition probabilities in the constructed Markov reliability model on system reliability is also discussed.

7.2 Uncertainty in input parameters in Markov reliability models and their limitations

Chapter two contains an introduction of software reliability modeling. Some of the catastrophic accidents, due to the failure of software, have been summarized to understand the essence of software reliability. Then the limitations of SGRM have been given, and shown that these limitations can be addressed by predicting the software system reliability in early phases of SDLC. Some of such approaches are based on Markov model. We emphasized on the necessity of precise and ambiguous requirements for constructing a reliability model. The importance of involvement of all the stakeholders for constructing a reliability model is discussed. Based on it we make the use of UML for converting it into the Markov reliability model. Then the limitations of the current approaches of early prediction using Markov reliability model have been discussed, which needs to be addressed. A software system is made of many components. We also considered the importance of the impact analysis of the change in reliability of any component in both the early phase, to take preventive action; and in operational phase, to take preventive an corrective action. We discussed that these issues can be addressed in our contribution section. At the end of this chapter, we gave the outline of the thesis.

7.3 Impact analysis of component reliability on the reliability of a software system for preventive action

Complex software system is composed of several software components and the reliability of each can be predicted during the design phase using Markov reliability model. There is a requirement for impact analysis of reliability of a component on the other associated components and the overall system during the architectural or design phase of SDLC to take preventive action, is required. The preventive action may include: (i) change the design of a software component (ii) replace a component (iii) remove a component (iv) optimize the coupling among the software components. Any of the mentioned preventive action, or combination of these may be taken to meet the target reliability of a software system. BN and MCS have been used to compute the updated estimate of reliabilities of the components of the CBS or CBS itself, whenever any of its component reliabilities or system reliability changes. The experimental validation of this approach is also shown on the same case study.

7.4 Updation of components reliability during testing/ operational phase for corrective action

During the system operation, the environmental condition changes and hence the reliability of any component may get change. Hence there is a requirement to know the impact of reliability change of one component on other components and overall system, during the dynamics of the system to take preventive action, for example-by switching to the healthy redundant system; and to take corrective action, for example- repairing the faulty or degraded component. Further the reliability of the COTS component can be discovered during system dynamics only because of unavailability of its design. Hence decisions about COTS components in a CBS can also be taken. The same approach as given in section 7.3 is used to address this issue. The experimental validation is also shown on the same case study.

7.5 Future Work

In our approach of early prediction of software reliability using UML, we need to extend the model to include the time information, which is useful for many real time systems for performance analysis during the early phase. The rest of the limitations that are given in the chapter 2 are also subject to the future research. Also, the effects of CPU hopping and cache thrashing vastly distort the results and hence needs to be included in the modeling phase. Our future research also includes designing and conducting experiments to find out the limitations of our approaches for early prediction, parameter estimation and reliability re-estimation and solutions to address them. Also, for the usefulness of the Markov reliability model for different types of application, different parameters of the models need to be estimated. In the current trend of software system development, many COTS are being used, which are black-box and difficult to expose its architecture. The future research can be developing the architecture from the COTS test data to construct its reliability model for reliability assessment before actually deploying into the software system.