A Study of Software Reliability under Different Risk Evaluation Models

Neelam[1], Mukesh Singla[2]

[1]Student, M. Tech (CSE), Satpriya Group of Institution, Rohtak, Haryana
[2]Associate Prof, CSE Dept, Satpriya Group of Institution, Rohtak, Haryana

ABSTRACT

A software system is defined under different product level and process level measures to define the software quality and reliability. The software system evaluation is here provided with specification of various risk and aspect analysis. In this paper, some of the model and the risk evaluation measures are defined. At first the paper has explored different risk measure and the quality assessment vectors. Later on, the paper has identified some of the risk estimation methods and models. These models are able to provide the risk evaluation under various criteria and aspect specification.

Keywords: Software Risk, Risk Modeling, Software Quality, Software Measures

I. INTRODUCTION

A software product is required to validate various aspects of software quality including the efficiency, scalability, reliability, complexity etc. These all vectors are evaluated at different stages of software development life cycle and predicted respective to the different measures. The prediction of expected quality constraint before any processing the stage is called the planning. This planning is able to identify the expected quality constraints. These constraints are later on mapped to the actual quality constraints and collectively considered as the accuracy of the estimation stage. The software quality is able to perform the cost estimation, development time estimation and fault prediction in the software system. One such planning stage analysis under quality measure is the risk vector. A software system suffers from various kinds of software risks. Such as the process risk it identified the chances of process completion, chances of error occurrence etc. These risks can be software product itself, the dependent software tools or products, the stakeholders specific risk and the environmental vector specification. Some of the common risks associated to the software system are listed here in figure 1.

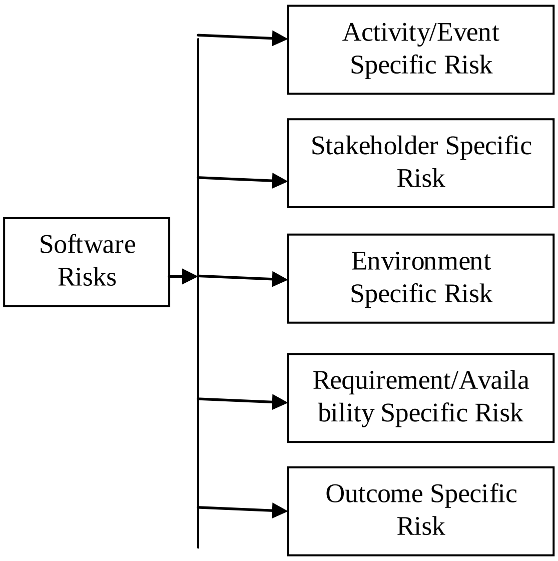
Software Risks
Activity/Event Specific Risk
Stakeholder Specific Risk
Environment Specific Risk
Requirement/Availa bility Specific Risk
Outcome Specific Risk

Figure 1: Types of Software Risk

Here figure 1 is showing the expected software risk that can occur in different stages of software development. The foremost risk type is the activity specific risk that is identified to evaluate different process or the run time specific observation. The risk is bounded the applied event, process or the function execution. The risk is about the occurrence of some failure, fault or error while executing some task. The executions of the process in scheduled sequence as well as before the deadline are also the integrated risk factors.

Another risk form associated to software projects is stakeholder specific risk. Each of the people involved in the development process comes under this risk stage. The programmer, project management, the technical person or the tester comes under this vector. The quality of the individual, experience, capability to deal with some problem or project, their role, dependency can be considered as the individual risk. Environment risk is about the analysis of the platform or the tool on which the software will execute. If the designed software is not supported to the environment or platform or the software system is having the fault in the platform itself, it can be identified as the software risk. The risk vector can be identified the support to the application, stakeholder and the resources. The weaker the support to any of these objects, higher the risk will be in software development.

Another risk form associated to the software system is the requirement and availability map based risk. If the software system does not meet the hardware, software or the environment specific requirements, it cannot provide optimized results. Because of this, there is the requirement of some method that can apply the mapping between the requirement and availability and verify the software system dependency. The memory availability, processor support, bandwidth requirement check can be performed to regulate and evaluate this risk factor. Another risk for defined to identify the deficiencies in the software system is called outcome specific risk. The end product, its features constraints, assumption, functions are required to match the planned requirement. The degree of mismap is here identified as the error. This chance of this degree of variation is identified as the outcome specific risk.

In this paper, an exploration to the software risk is provided along with the exploration of different risk models. The risk estimation and evaluation are discussed in this paper. In section I, the software risk with basic constructs is explored. The section has defined and explored different type of risk. In section II, the work defined by earlier researchers is discussed. In section III, some of the common software risk models are defined. In section IV, the conclusion of work is presented.

II. RELATED WORK

Lot of work is already provided by different researchers to identify different kind of risks in software system. The risk constraints, evaluation and characterization were discussed and improved by different researchers. In this section some of the contribution of earlier researchers for risk evaluation is presented. Author [1] has defined a work model to identify the impact of software risk in software management. Author identified the risk of integration of these associated tools and theories to the real environment. The community level and project level estimation provided by the author with specification of relative tools. Author also provided the activity driven observation at module, component and individual level so that the risk will be identified at the earlier stage of software development. Author[2] has provided a work on risk assessment for system with specification of casual modeling. The Bayesian integrated probabilistic model was used by the author to evaluate the risk and to generate the quantized mapping results.

A decision driven estimation was provided by the author to model the risk under various associated constraints. Author also defined decision support system for the project managers and the programmer to provide the effective system specific modeling. The tool was controlled by setting up the commercial constraints and to provide the relative execution in global environment. Author[3] has provided a work on software risk assessment under the knowledge driven estimation and basic structure specification. The sub ontology specific estimation and the relative repository driven learning method was provided by the author identify the software risk. The component generation and its mapping to the knowledge repository were discussed by the author. Author also provided the ontology specific risk observation factor to generate the development constraint. The visualization method along with ontology control map was provided to improve the reliability of developed software system.

Author[4] work on case specific measurement and process risk derivation was provided by the author to generate the software risk with safety evaluation. The risk estimation and evaluation was provided by the author with specification of spaceflight program. Author defined the hazard analysis was defined with process quantification so that the safe design phase evaluation will be obtained. Author[5] has defined a risk evaluation and assessment method to generate the conceptual measure based evaluation. The estimation of system with semantic feature generation and mapping was provided. A risk specific ontology generation and processing was provided by the author to control the interrelationship between the components. The management specific control method was defined to control the resources and to provide the support for linear software system. The support to the real time software system was provided under the effort and time estimation. Author[6] has defined a work on Cocomo II model with integration of portfolio management. The capability driven analysis with unrealistic linearly was provided by the author. The effort, efficiency and reliability of software system are observed to improve the software quality. Author[7] has defined process specific modeling and risk evaluation in cost sensitive learning method. The holistic analysis and predictive measures are defined in the problem generation and exploration.

The legal risk based estimation was provided to reuse the product and mitigate the risk vector. The software artifact based estimation and evaluation was provided by the author. The problem discovery, risk estimation and relative software system evaluation was provided by the author. Author[8] has defined an open cocomo system with extended cost and constructive observation to provide the evaluation at the deep level. The clean feature based evaluation and the relative process modeling in detailed environment was provided by the author. Author identified the product adaptive, hardware adaptive, personal adaptive and project adaptive results using cocomo model[8]. Author has defined a cocomo model for education and training system. Author identified the influence of different feature vectors with domain specification under cocomo model. This constructive cost based model has used the black box testing model and aspect to identify the concern of software system[9]. Author has provided a special application specific estimation of software system cost using Cocomo model. Author provided the estimation in real time and defined cost and unit integrated testing model. The statistical estimation and support was considered to estimate the cost and expenditure of software product. Author processed o small scale project and designed a new software system with special integrated application. The project estimation was done to reduce the error rate[10]

III. RISK FACTORS AND EVALUATON MODELS

To develop a reliable software system, an organized development process is required with time constraint specific measurement and evaluation. These evaluations are requirement to measure the quality of software product. Software development itself defined with integrated concerns, issues and challenges. Because of this, there is the requirement of some précised measurement process to identify the quality of software system. The software system is itself composted of various components and characterization. The domain dependency, user environment, degree of acceptability are some of the common factors based on which the software measurement and development can be regulated. Some of these factors, issues or the required observations are shown here in figure 1.

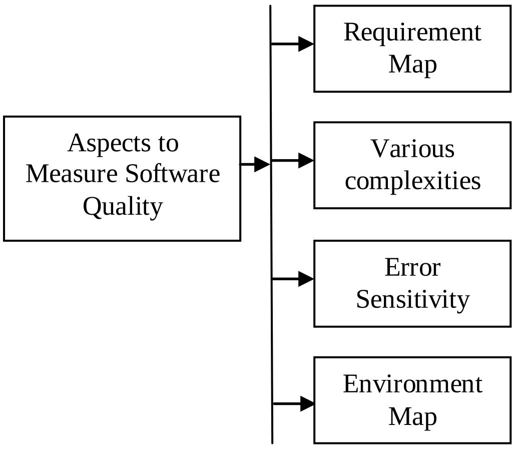

Figure 2: Measures of Software System Quality

Here figure 2 is showing some of the measures based which the software system quality can be measures or estimated. The figure is showing the different constraints respective to the client support are defined based on which the software quality can be measured. The first parameter considered here is the requirement map. It means the verification of the system between the generated software system and the relative measures. The expected and the actual values are compared to identify the degree of error in development process. The requirement specific analysis can be applied to perform the requirement specific mapping. Another parameter is the complexity driven mapping. Such mapping can be defined in terms of time taken, space requirement etc. The degree of satisfiability against these all vectors collectively defines the measures of software quality. The third parameter is the error sensitivity. The first requirement here is to avoid the chances of error occurrence, even them if some occasional or the event specific fault or error occur, it is required to measure the impact of the error and the impact on environment, user data and the application. Based on this the sensitivity of the error can be identified. The error sensitivity also defined as the criteria respective to which the requirement map can be defined. If the software system is highly sensitive to these vectors, then high quality aspects are required. The final measure to the software system quality is the environment map. The effect of the new software product on other products, operating systems are required to estimate. Based on this the quality or the risk dependency can be evaluated. To perform the reliability analysis, there is the requirement to evaluate the risk under different vectors. Here some of the common risk assessment models are defined

A) Weibull Failure Model

This particular model observes the risk for a software system under the error or the fault level observation. The model analyzed the software system defects and evaluates the impact on software system as well as on different components of software system. The fault specific analysis is here performed to identify the platform support, environment support and the various integrated estimations. These estimations can be applied based on the average input and the execution of the software system with error. The real time systems, embedded system are the main concern area of this model.

B) Musa Model

The model deals with the software quality or reliability estimation under the execution time. The real time software systems are very sensitive about the execution and completion of some task. This model estimates the process time, wait time, arrival time and execution time and generates various measures to define the quality of software system. The decision vectors are applied to observe the system under time constraint specification so that the reliability of software system will be improved. The model also evaluates the software fault and bound them with specification of time constraint. Some of assumptions are applied to provide the valid and error specific execution. The model uses the statistical evaluations to generate more concerned and adaptive results. The confidence vector based evaluation is also provided to generate more effective and optimized results.

C) Monte Carlo Model

This particular model is defined to estimate the software system reliability for real time projects of finance and physics stream. Different researchers provided the work contribution to handle the uncertainty conditions of this software system. The model applies the random test based on the run time evaluation. The fault, time, memory requirements are check under the brute force attack.. This model is comparatively takes higher execution time and the cost. But the estimation provided by the model is reliable.

CONCLUSION

In this paper, an assessment to software system is provided under different risk assessment measures. The paper has first identified the software system evaluation with specification of different measures. Later on different models are defined to utilize these measures to estimate the software quality, reliability and the risk.

REFERENCES

[1]. Victor R. Basili," Obtaining Valid Safety Data for Software Safety Measurement and Process Improvement", ESEM’10, September 16–17, 2010, Bolzano-Bozen, Italy. ACM 978-1-4503-0039-01/10/09

[2]. G. Antoniol," Requiem for software evolution research: a few steps toward the creative age", IWPSE'07, September 3-4, 2007, Dubrovnik, Croatia. Copyright 2007 ACM ISBN 978-1-59593-722-3/07/09

[3]. Piotr Tomaszewski," Comparing the Fault-Proneness of New and Modified Code — An Industrial Case Study", ISESE’06, September 21–22, 2006, Rio de Janeiro, Brazil. ACM 1-59593-218-6/06/0009

[4]. Shareeful Islam," Software Development Risk Management Model – A Goal Driven Approach", ESEC/FSE Doctoral Symposium’09, Aug. 25, 2009, Amsterdam, The Netherlands. ACM 978-1-60558-731-8/09/08.

[5]. Tokuro Matsuo," Diversification of Risk based on Divided Tasks in Large-scale Software System Manufacture", June 12, 2007. San Diego, California, US ACM 978-1-59593-856-5/07/06

[6]. Stig Larsson," Assessing the Influence on Processes when Evolving the Software Architecture", IWPSE'07, September 3-4, 2007, Dubrovnik, Croatia ACM ISBN 978-1-59593-722-3/07/09

[7]. Andreas Schmietendorf," Process models for the software development and performance engineering tasks", WOSP '02, July 24-26, 2002 Rome, Italy ACM ISBN 1-1-58113-563-7 02/07

[8]. H. K. Sharma, R. Tomar, A. Dumka and M. S. Aswal, "Open ECOCOMO: The algorithms and implementation of Extended Cost Constructive Model (E-COCOMO)," Next Generation Computing Technologies (NGCT), 2015 1st International Conference on, Dehradun, 2015, pp. 773-778.

[9]. R. E. Fairley, "The Influence of COCOMO on Software Engineering Education and Training," 19th Conference on Software Engineering Education & Training (CSEET'06), 2006, pp. 193-200.

[10]. J. E. Helm, "The viability of using COCOMO in the special application software bidding and estimating process," in IEEE Transactions on Engineering Management, vol. 39, no. 1, pp. 42-58, Feb 1992.