Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 6, June 2014, pg.426 – 431

RESEARCH ARTICLE

An Effective Model for Software Risk Estimation

Ms. Alankrita Aggarwal
Assistant Professor, Department of Computer Science & Engineering, HCTM Technical Campus, Kaithal, India

Anu
Research Scholar, M.Tech, Department of Computer Science & Engineering, HCTM Technical Campus, Kaithal, India

Abstract- One of the major area for software risk estimation is the analysis of faults over the software modules. These modules are analyzed individually and aggregatively to under different vector. In this work, a metric based estimation approach is defined to estimate the software reliability. The metrics considered in this work includes the cohesion metric, coupling metric, module interaction metrics and system metrics. Based on these metric based analysis overall software risk is estimated.

Keywords: Software Risk Estimation, Metric Based, Software Quality, Module Interaction

I. INTRODUCTION

Software development and software quality analysis are the parallel processes that are used to develop a quality software product. The software measurement is required to perform the software analysis under different metrics so that the software reliability will be improved. Software analysis is required to predict the software system under different quality metrics so that the effective software generation will be done. There are number of parameters to develop a software system. These parameters include the prediction and process improvement parameters. These helps to analyze the software system under theoretical, conceptual and practical aspects. Based on the earlier analysis, the real time software systems can be analyzed and predicted for the effective analysis. There are number of existing models and parameters based on which the software systems can be analyzed. These models or approaches are adopted by different organizations to deliver quality product.

There are different points of views under which the software quality analysis can be done. These parameters include the software fault analysis, software quality analysis, software reliability analysis etc. Software engineering the effective branch of software testing system used to analyze the software system under statistical analysis. There are number of methods and models available to measure the system under different aspects.

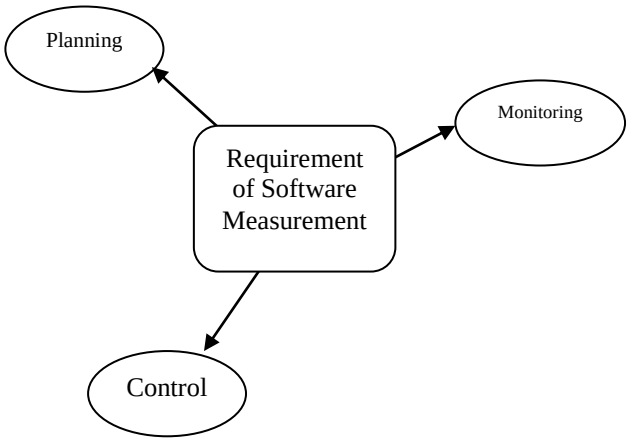

Figure 1 : Software Measurement Constraints

As shown in figure, software measurement is effective to plan the software process or software system. Software plan is not a single term; it defined a complete scientific process that represents the development procedure blue print. It means a software plan is able to define the software system under different aspects so that effective software development will be performed. Another aspect of software system is the monitoring of software system. Monitoring is here defined to analyze the ongoing process improvement and the objects meet till now. Another aspect of software metrics is the software control. The control is about to develop the software system under certain limits so that the software risk over the system will be minimized.

A) Software Measurement

Software measurement represents the rule based development process that includes the software evaluation, measurement and control. This kind of measurement includes the cost based analysis along with software system prediction. The prediction is here done to analyze the development process under the plan rules. These kind of software system are defined in effective rule generation. There are two types of approaches adopted to measure the software system.

i) Direct Measure

These kind of software systems are defined under the cost analysis so that the effective software process model can be defined. These kind of software systems are defined under different measure such as Line Of Code, Execution Speed, Complexity Analysis and Defect Rate Analysis.

ii) Indirect Measure

These kind of software systems are defined to analyzed the software system under different aspects. These aspects are shown in figure 2

Figure 2 : Software Measurement Aspects

Software measurement is defined as the process analysis model used to measure the software system under some defined hypothesis. These hypotheses are able to identify the project goal with actual outcome. The knowledge based system analysis is defined to measure the software product under analysis at different levels. These levels defined the software quality measurement along with specification of software system under the software quality analysis. These kind of software system can improve the software product and quality by controlling the software process.

In this paper, a model for software risk estimation is defined under software metrics analysis. The analysis is here defined under four different metrics called, module interaction analysis, coupling and cohesion metrics. In this section, the basic definitions of software risk and software measurement are defined. In section II, the work defined by earlier researchers is discussed. In section III, the proposed model is defined and explored. In section IV, the conclusion of this paper is defined.

II. RELATED WORK

In this section, the work defined by the earlier researchers is discussed. Lot of work is already defined by different researchers in the area of software quality estimation based on software metrices. Some of the work defined by earlier researchers is discussed in this section. This chapter includes the estimation of software product under different parameters such as cost, size, effort and quality estimation. There are number of reliability models used by the researchers to estimate the software quality. The software quality and reliability are estimated to analyze the software system effectively. These reliability metrices depends on reliability model based on which the performance computation and the complexity estimation is done. The software estimation based analysis is performed by the researchers using metrices. These metrics gives the analysis in quantitative form so that the easy decision can be taken for software quality. The quality metrics are defined under the component and process based estimation to schedule the software system for future projects. The productivity of the software system is defined under the impact analysis so that the software maintenance and estimation can be performed. In this chapter, the work defined by earlier researchers based on these quantitative measures is explore in detail. These estimation is performed for procedural as well as object oriented metrices to analyze the software system under the structural programming concept so that data structure based analysis can be performed effectively. Lalji Prasad,Aditi Nagar has defined a work on the analysis of software system for different structural and object oriented metrices. This metric based component estimation along with relationship analysis is performed under the procedure based analysis. Author discussed the metrics such as LOC, cyclomatic complexity, cohesion and coupling metrics. Author defined the analysis under the class level analysis so that the software product error detection and correction can be performed over the system. The estimation of software system under the operational measures is also performed to analyze the software quality. The operational features of the project and product is also discussed along with coupling analysis so that the estimation of software related metrices will be obtained. The paper has also discussed the software coupling measurement under the structural analysis. Zeeshan Ali Rana has defined an estimation on software products to analyze the software system under defect analysis for object oriented software system. Author defined the work to perform the defect prediction in the software system and analyze the software system effectiveness under module level defect analysis. The defects are analyzed under different vectors such as number of defects. Author defined the coupling based analysis, association analysis, dependency analysis and the interface analysis. The class and inheritance analysis is performed to analyze the software product effectively. The UML diagrams are constructed to establish the relationship between classes. These kind of software system also able to present the static view of system so that the decision regarding the software quality can be done. Software system analysis is performed under the complexity metrics based analysis. Barry W. Boehm has presented resource based software estimation scheme for software quality analysis. Author defined budget analysis approach to improve the software product analysis . Author performed analysis under different testing aspects. E Da-wei, Xiamen has defined an improved metrics based complexity model for object oriented programming. Author defined the complexity analysis under multiple aspects so that effective software development under method analysis will be performed. Author defined the analysis under the software complexity and method analysis so that the software design model will be improved. Author defined the development process with the specification of cost model under the size and volume analysis so that the prediction to the software system will be done effectively. Author defined the structural complexity model under integral factor so that the development effort will be reduced. Author defined a size and complexity based model for development of software system under cost estimation. Author defined an object oriented program so that effective software development will be done. Author defined a predictive software analysis under quality modeling so that the software system analysis will be done. Author defined an effective software evaluation and measurement for software system under traditional metrics analysis. Author defined a metrics suit with coupling, cohesion so that the software cost estimation will be done effectively.

A computational system for the software system under the software development rules was defined by the author under cost, timeline and quality analysis. Author defined the software development under software modeling and analysis. Author has

defined a professional practice analysis and model for development process so that effective development model will be presented.

A computational work to perform the software development and software product analysis under cost and time analysis was done by the researcher. Author defined a parametric analysis on software system under software quality and software history analysis. Author defined the software process and software design mechanism to analyze the software system under software evolution and software management analysis. Author defined the specific development model so that the development process will be improved. Author defined a set of management approaches to analyze the software system under software quality analysis. Author defined the management analysis program under the specification of research analysis and development by validating the software management under software improvement. Author defined the software measurement analysis under the defined framework so that the software measurement validation is performed. Author defined the structural model for software development so that the attribute relation analysis will be performed. Author defined the entity analysis so that the software development and validation will be performed. Author defined the software measurement based program to analyze the software system under rule specification and to analyze the program under different metrics. Author defined the effective path generation so that effective software measurement and testing will be applied over it.

III. Metrics Based Model

Metric model is used to evaluate the metric results. Figure 3 describes the metric model used for the selection of metrics and the selection of the metric further depends on the development phase of software product. In the starting stage of the development, process metrics is used and if it is the final stage of development (before the customer approval) then, product metrics is used. The results thus obtained from the metrics are later compared with the standard or the selection of the software product. The evaluation which is done is human.

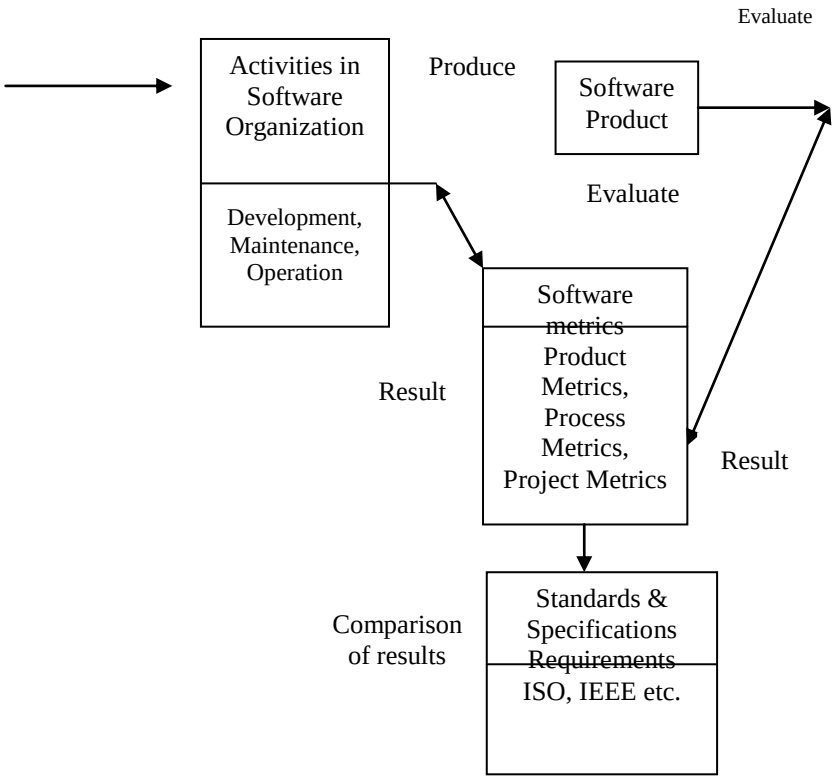

Figure 2 : Software Metrics Model

In this present work, a metrics analysis approach is been presented to perform the software risk estimation. The presented approach is defined under three main vectors. In the first stage, the individual module analysis over the software system is identified and analyzed. In this analysis state, the variable and methods correlation is been analyzed. The module interaction analysis is respective to the modules. The complete software system is dived in terms of software modules and each module is

defined along with integrated complexity. After the module complexity analysis, the software interactivity analysis between modules is done. The interactivity can be of based on access frequency or based or on interaction criticality.

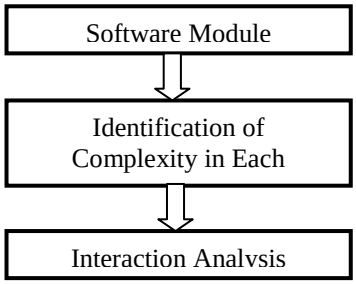

Figure 4 : Software Module Analysis

Once the module complexity is identified, the next work is to find the association between software modules. The association between the software complexities is called software system analysis dependency.

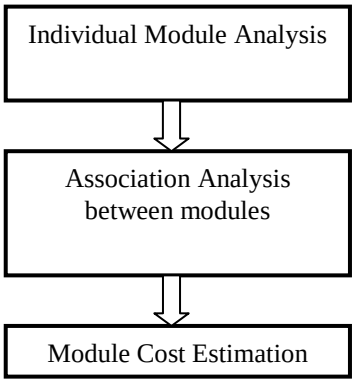

Figure 4 : Module Cost Estimation

The independent modules are comparative more effective than dependent modules. The cost estimation on each software module interactivity is applied to identify the association between these module methods. Based on this, the criticality of each software module will be identified individually. The cost estimation process is shown in figure 4.

At the final stage, the estimation of the software cost will be done by performing an aggregative cost analysis. While estimating the aggregative cost, at first the weightage is assigned to different software modules. Based on this weightage assignment. The aggregation on these vectors will be performed to identify the overall software cost or the software cost criticality.

IV. CONCLUSION

The presented work is about to perform a software reliability estimation under module metrics analysis. In this work, two level approaches is defined to perform the criticality analysis as well as cost analysis. The first level is implemented on individual modules and second level is implemented on aggregative cost. The obtained results show the clear module of criticality under fault vector.

References

[1] LiGuo Huang," Text Mining in Supporting Software Systems Risk Assurance", ASE'10, September 20-24, 2010, Antwerp, Belgium. ACM 978-1-4503-0116-9/10/09

[2] Mary Sumner," Risk Factors in Enterprise Wide Information Management Systems Projects", SIGCPR 2000 Evanston Illinois USA 1 -58113-212-x/00/04

[3] 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

[4] Guillaume Langelier," Visualization based Analysis of Quality for Large scale Software Systems", ASE'05, November 7–11, 2005, Long Beach, California, USA. ACM1-58113-993-4/05/0011.

[5] Mira Kajko-Mattsson," Laying out the Scope of Developers' Risk Management Responsibilities", ICIS 2009, November 24-26, 2009 Seoul, Korea ACM 978-1-60558-710-3/09/11

[6] Ossi Taipale," Improving Software Testing by Observing Practice", ISESE'06, September 21–22, 2006, Rio de Janeiro, Brazil. ACM 1-59593-218-6/06/0009

[7] Michael Grace," RiskRanker: Scalable and Accurate Zero-day Android Malware Detection", MobiSys'12, June 25–29, 2012, Low Wood Bay, Lake District, UK. ACM 978-1-4503-1301-8/12/06

[8] Adailton Mag alhaes Lima," Risk Assessment on Distributed Software Projects", ICSE '10, May 2-8, 2010, Cape Town, South Africa ACM 978-1-60558-719-6/10/05

[9] Margaret-Anne Storey," The Impact of Social Media on Software Engineering Practices and Tools", FoSER 2010, November 7–8, 2010, Santa Fe, New Mexico, USA. ACM 978-1-4503-0427-6/10/11

[10] Peter Hearty," Automated Population of Causal Models for Improved Software Risk Assessment", ASE'05, November 7–11, 2005, Long Beach, California, USA. ACM 1-58113-993-4/05/0011

[11] C. R. Rene Robin," Development of Educational Ontology for Software Risk Analysis", ICCCS'11, February 12–14, 2011, Rourkela, Odisha, India. ACM 978-1-4503-0464-1/11/02

[12] Lucas Layman," A Case Study of Measuring Process Risk for Early Insights into Software Safety", ICSE'11, May 21–28, 2011, Waikiki, Honolulu, HI, USA ACM 978-1-4503-0445-0/11/05

[13] C R Rene Robin," An Ontology Based Linguistic Infrastructure to Represent Software Risk Identification Knowledge", ICWET'11, February 25–26, 2011, Mumbai, Maharashtra, India. ACM 978-1-4503-0449-8/11/02

