International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-3, July 2013

A Fault Analysis based Model for Software Reliability Estimation

Garima Chawla, Santosh Kr Thakur

Abstract: ***When a software system is designed, the major concern is the software quality. The quality of software depends on different factors such as software reliability, efficiency, cost etc. In this paper, we have defined the software reliability as the measure of software quality. There are different available models that estimate the reliability of software based on type of faults, fault density etc. In this paper, a study on different aspects related to software reliability are discussed..***

Keywords- Software Faults, Reliability, Quality.

I. INTRODUCTION

Software reliability is about to define the stability or the life of software system with different properties. These properties includes the trustfulness of software system, software cost, execution time, software stability etc. The aspects related to these software system includes the probability of software faults, frequency of fault occurrence, criticality of fault, associated module with respective fault etc. In a software development process, the pre estimation of software reliability is required to deliver the software product. According to the required level of software quality estimation of software cost, development time are also estimated. There are number of quality measures that approves the software reliability.

Each stage of software life cycle itself take some time quantum to deal with software reliability. Higher the software quality, lesser the software maintainability. If a software is a quality software, it requires minimum changes or negligible chances of faults after the software delivery. To estimate the software quality different software matrices are available. These metrics includes the process level as well as product level matrices. Software engineers are required to follow a systematic and organized plan to deal with the software faults as well as software reliability. The another vector that influence the software reliability is the dependency vector. It is always easy to analyze an independent module whereas as the dependency occur in a software system, it is never easy to estimate the quality of the software. In dependent modules, the fault in one module increase the criticality of other modules[2,4]. Reliability vector is important for all stakeholder of software products or for each stage of life cycle. All the parties that deal with software system on any stage having the equal importance vector respective to the software reliability. The parties can be manager, tester, developer etc. At the initial stage when the software plan is been generated the software reliability and criticality are the major concern.

According to the type of software system design the reliability vector of a software is been defined. Such as a real time software system is more critical enough then a gaming software system. So that it requires more efforts, cost and time.

Another concern, while defining the software reliability is the security vector. What level of security is defined by a software system also defines its reliability. The security vector defined here in terms of some data security approaches, authenticity etc. Such as a web based system requires more security efforts whereas the personal software system requires less security. In such system, the estimation and testing of reliability under security is also required. In this section of such measure of software reliability system are been defined [1,6]

A)Software Error

A software error can be an exception that can occur over the system because of some wrong input, because of some internal or external software fault or because of some hardware problem such as memory overflow. A software error can be fatal or non fatal error. It means it can be ignored and enable to process with error. But as some fatal error occur over the system, it can restart the application, tool or the whole system. Software error can be critical if it affect the system and not even allow to save the work done and perform the abnormal termination. The error criticality also depends on the associated module. At the initial stage, when a software is been implemented on client end, there are more chances of occurrence of such kind of errors over the system[5,6].

B) Software Faults

An error because a fault or the defect if the error is identified as the basic error that associates with some software function such as if some application restart occur as the print command is given. Once the fault identified, the next work is about to identify the criticality of the software fault. The fault criticality depends on the type of fault and the associated module. If the associated module is only some help or information module the criticality is less where as if it attached with main modules the criticality of the software fault will be higher. The action associated with the fault is also define its criticality. If the fault affect the system or the tool then the criticality of the fault will be higher[5,6].

C)Software Failures

As the fault is identified over the system, now the next work is to estimate the reasons and the results of the fault occurrence. Along with that the estimation of fault frequency is also analyzed. To perform this estimation the gap between two software faults as well as the number of faults in a time span is also identified.

Revised Manuscript Received on 30 July 2013.
* Correspondence Author
Garima Chawla, Department of CSE, BRCM Engg. College Bahal, India.
Santosh Kr Thakur, Department of CSE, BRCM Engg. College Bahal, India.

The another vector included here is the dependency of the fault with other faults as well as its association with different modules of software system. If the fault crashes the software system it is taken as the software failure. Software failure is the most critical stage of software reliability. Generally the failure can occur because of compatibility issue. If the hardware or some other software is not compatible to the software system it can result software crash.

In this paper, the software reliability is been discussed under different aspects of software faults and related aspects.

II. LITERATURE SURVEY

Sultan Aljahdali performed a work,"Predicting the Reliability of Software Systems Using Fuzzy Logic". In this paper, Author explore the use of fuzzy logic to build a SRGM. The proposed fuzzy model consists of a collection of linear sub-models joined together smoothly using fuzzy membership functions to represent the fuzzy model. Results and analysis based data set developed by John Musa of Bell Telephone Laboratories are provided to show the potential advantages of using fuzzy logic in solving this problem. Khalaf Khatatneh performed a work," Software Reliability Modeling Using Soft Computing Technique". In this paper, a model that can be used for software reliability prediction is explored. The proposed model is implemented using the fuzzy logic technique and has been applied on a custom set of test data. The model is characterized as a growth reliability model. This model focused on a particular dataset behavior in predicting reliability. Focusing on a particular dataset behavior is performed to develop an accurate model since the recent work focused on developing a model which can be more accurate. GUO JUNHONG performed a work," Software Reliability Nonlinear Modeling and Its Fuzzy Evaluation". This paper presents a nonlinear model of software reliability based on time series and gives corresponding algorithms. The simulation experiments show the accuracy and efficiency of this new model. The newly proposed model suits the requirement of software engineering better and its parameters can reflect the changing of software reliability. The new model can have more accurate analysis and forecast to software reliability issue without any strict assumption. Sultan H. Aljahdali performed a work," Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study". In this paper, Author explore connectionist artificial neural networks models as an alternative approach to derive these models by investigating the performance analysis of four different connectionist paradigms for modeling the software reliability prediction. The presented four paradigms are multi-layer perceptron neural network, radial-basis functions, Elman recurrent neural networks and a Takagi- Sugeno fuzzy inference system learned using a neural network algorithm (neuro-fuzzy model). Ajeet Kumar Pandey performed a work," Fault Prediction Model by Fuzzy Profile Development of Reliability Relevant Software Metrics". This paper presents a fault prediction model using reliability relevant software metrics and fuzzy inference system. For this a new approach is discussed to develop fuzzy profile of software metrics which are more relevant for software fault prediction. The proposed model predicts the fault density at the end of each phase of software development using relevant software metrics. K. Krishna Mohan performed a work," Selection of Fuzzy Logic Mechanism for Qualitative Software Reliability Prediction". The purpose of the this work is to demonstrate the same. Validation of this approach could be obtained by comparing the results with the ones obtained on realized prototypes at module level. In a work of the first of its kind involving studies at the PoC level, qualitative predictions for the metric 'number of defects' are obtained using a generic Fuzzy Logic based modeling. Jie Yang performed a work," Managing knowledge for quality assurance: an empirical study". The purpose of this paper is to examine the relationship between knowledge management and the quality of new product and identify different hidden patterns in which knowledge acquisition and dissemination affect the quality of new product. It employs Additivity and Variance Stabilization (AVAS) analysis. Findings – The quality of new product is related to knowledge management significantly. Michael R. Lyu performed a work," Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development". This paper considers "software component testing resource allocation" for a system with single or multiple applications, each with a pre-specified reliability requirement. The relation between failure rates of components and "cost to decrease this rate" is modeled by various types of reliability-growth curves. Closed-form solutions to the problem for systems with one single application are developed, and then "how to solve the multiple application problem using nonlinear programming techniques" are described. Also examined are the interactions between the system components, and inter-component failure dependencies are included in the modeling formula.

III. PROPOSED MODEL

Software reliability for a software system depends on following attributes

- Type of fault, Software faults are itself divided under different categories based on the influence of fault on software system. A fault can be warning, bug or the failure.
- Application Type, A fault severity depends on the type of application affected from the software fault. If the affected application is a game application then the criticality of system is less whereas if the application type is business or real time application the criticality of the software fault is higher.
- Associated Module, The module association with the fault is also the reason to identify the criticality of the fault. If the associated module in the system is having the higher criticality then overall criticality of the system will be affected.

In this paper, we have defined an analysis on software system under the fault based prioritization. As we know the reliability of a software depends on the effectiveness of software testing. To design a reliable software system it is required to identify and categorize these faults based on fault severity. To peform the analysis the first work is to divide the software system in terms of different modules associated with the software system. Once the software modules are defined, the next work is to define the associated faults with each module.

Once the faults are identified, the work is to list the associated test cases with each module and fault.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

These test cases are associated with the system, modules or the code blocks. In this paper, a metric based prioritization approach is been discussed. In table 1 some of these software faults and the associated prioritization is defined in this work.

Table 1 : Fault based Prioritization

Module Id	Function	Priority
1	Evaluation of Graphical Window	Low priority
2.	Database Backup	High Priority
3.	Data Recovery	High Prioirty

A)Algorithm

Algorithm

1. {
2. Define the software system in terms of associated software modules
3. Define the Faults associated with each software module
4. Categorize the software faults based on severity of software fault.
5. Assign the priority to different modules based on software fault and fault criticality
6. Reliability=1;
7. For i=1 to Length(Modules)
8. {
9. For j=1 to Length (Faults)
10. {
11. If (Associated(Module(i),Fault(j))=True)
12. {
13. K=Criticality(Fault(j);
14. Relibility=Relibility-K;
15. }
16. }
17. }
18. Return Reliability;
19. }

IV. CONCLUSION

In this paper, the study of the software reliability is been defined under the software fault analysis. The paper has also defined a fault based software cost estimation concept. The fault defined the software criticality under the fault study and based on it software reliability is been defined in this work.

REFERENCES

1. G Sultan Aljahdali,"Predicting the Reliability of Software Systems Using Fuzzy Logic".
2. Khalaf Khatatneh," Software Reliability Modeling Using Soft Computing Technique", European Journal of Scientific Research ISSN 1450-216X
3. GUO JUNHONG," Software Reliability Nonlinear Modeling and Its Fuzzy Evaluation", 4th WSEAS Int. Conf. on NON-LINEAR ANALYSIS, NON-LINEAR SYSTEMS and CHAOS
4. Sultan H. Aljahdali," Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study", ICGST-AIML Journal, ISSN: 1687-4846
5. Ajeet Kumar Pandey," Fault Prediction Model by Fuzzy Profile Development of Reliability Relevant Software Metrics", International Journal of Computer Applications (0975 – 8887)
6. K. Krishna Mohan," Selection of Fuzzy Logic Mechanism for Qualitative Software Reliability Prediction".
7. Jie Yang," Managing knowledge for quality assurance: an empirical study".
8. Michael R. Lyu," Optimal Allocation of Test Resources for Software Reliability Growth Modeling in Software Development", IEEE TRANSACTIONS ON RELIABILITY 0018-9529/02© 2002 IEEE
9. A. Yadav," Critical Review on Software Reliability Models", International Journal of Recent Trends in Engineering
10. Sultan H. Aljahdali," Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study", ICGST-AIML Journal, ISSN: 1687-4846
11. Michael R. Lyu," Optimization of Reliability Allocation and Testing Schedule for Software Systems".
12. J. O. Omolehin," Graphics to fuzzy elements in appraisal of an in-house software based on Inter-failure data analysis", African Journal of Mathematics and Computer Science Research
13. P.C. Jha," A fuzzy approach for optimal selection of COTS components for modular software system under consensus recovery block scheme incorporating execution time", TJFS: Turkish Journal of Fuzzy Systems (eISSN: 1309–1190)
14. Qiuying Li," A Software Reliability Evaluation Method", 2011 International Conference on Computer and Software Modeling IPCSIT

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)