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. 2, Issue. 6, June 2013, pg.116 – 122

RESEARCH ARTICLE

A Fuzzy Improved Association Mining Approach to Estimate Software Quality

Sonakshi Rana[1], Rahul Kumar Yadav[2]
[1]CSE, PDM Bahadurgarh, Haryana, India
[2]Assistant Professor, CSE, PDM Bahadurgarh, Haryana, India

[1] sonakshirana@yahoo.com; [2] er13rahulyadav@gmail.com

Abstract— Software Quality analysis is one of the major criteria required to analyze the software life as well as software reliability. Software quality is been defined under different parameters. Software risk analysis is one such criterion required to identify the software reliability. When software is planned or being developed according to the type of software as well as the efforts required to develop the software collectively defines the software risk. Such as the availability of the required software, hardware, manpower all are the predictive risk factors. In this work, these all risk factors are defined under the fuzzy rule. Based on this fuzzy estimation to the some weightage is been assigned to these all risk factors. Finally, the aggregative risk is been computed to predict the software reliability under the risk vector. The presented work is capable to identify the software risk under three levels. These three levels are individual vector risk, sub category risk and the aggregative risk.

Key Terms: - Software Risk; Reliability; Fuzzy Logic; Weighted Approach

I. INTRODUCTION

Software Reliability is the probability of failure free software which work for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability. Software reliability is different from hardware reliability because it defines the design perfection, rather than manufacturing perfection. The high complexity of software is the major contributing factor of Software Reliability problems.

Hardware Reliability: - It is the ability of hardware to perform its functions for some period of time without any failure. It is expressed as MTBF which means mean time between failures. The hardware reliability is described on the basis of bath tub curve.

Software Reliability: - Software Reliability is the probability of failure free software which work for a specified period of time in a specified environment. Software Reliability is also an important factor affecting system reliability. Software reliability is different from hardware reliability because it defines the design perfection, rather than manufacturing perfection. Below diagram describes the software reliability.

A) Reliability Model

There are number of software reliability models which are used for different purposes. These models are explained below:-

1. Jelinski Moranda Model

The Jelinski Moranda model is the earliest models in software reliability .it is given in 1972.It is time between failure models. It assumes N software faults at the start of testing, failure occur purely at random and all the faults contribute equally to cause a failure during testing. It also assumes the fix time is negligible and the fix for each failure is perfect. The software product's failure rate improves by the same amount at each fix.

2. **Littlewoods Model**

It is similar to Jelinski Moranda Model but it assumes that different faults have different size. Large sized defects tend to detected and fixed earlier. As the number of errors is driven down with the progress in the test, so is the average error size causing a law of diminishing return in debugging.

3. **Goel Okumoto Imperfect Model**

The J-M model assumes that the fix is negligible and that the fix for each failure is perfect. It assumes perfect debugging .In practice this is not always case. In the process of fixing a defect new defects may be injected. This model proposed imperfect debugging model to overcome the limitation of the assumption. In this model the hazard function during the interval between the (i-1) st and the failure is given

$$Z(ti) = [N-p(i-1)]\lambda$$

Where N is number of faults at start of testing
P is probability of imperfect debugging
λ failure rate per fault

4. **Goel Okumoto Non Homogeneous Poisson Process Model**

The NHHP model is given in 1979.It is concerned with modeling the number of failures observed in given testing intervals. It defines the cumulative number of failure observed at time t.N (t) can be modeled as a non-homogenous Poisson process with the time dependent failure rate.

5. **The Delayed S and Inflection S Models**

With the help of defect removal process Yamada said that a testing process consists of not only a defect detection process but also defect isolation process because of the time needed for failure analysis significant delay can occur between time of first failure observation and the time of reporting. They offer the delayed S-Shaped reliability growth model for such a process in which the observed growth curve of cumulative number of detected defects is S-shaped. The model is based on the no homogeneous poison process but with a different mean value function to reflect the delay in failure reporting.

II. REVIEW LITERATURE

Zigmund Bluvband (2011) describes that there are two advanced analytical models for obtaining accurate results for software reliability prediction. The First model can inhibit some specific features of software testing process and it is based on well-known S shaped Ohba model. This advanced model is applicable only for non-rare bug testing. For the rare bug rate prediction other model is proposed which is based on introduction of the additional control parameter last suspended time. He describes the easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

Advanced parametric models for assessment and prediction of software reliability that is to find the bugs in the software which is based on statistics of bugs at the initial stage of testing. The parametric model approach commonly associated with reliability issues which deal with the evaluation of the amount of bugs in the code.

Computed parameter from the allow estimating:

Number of bugs remaining in the product

Time required detecting the remaining bugs.

In his paper he describes different models are developed for doing same purpose Duane Reliability Growth Model, Goel Model, Weibull Model, Classical S-shaped Model, Ohba S-shaped Model, etc. By taking some details of these models and by doing practical aspects of the software testing process a few Advanced Models were developed and implemented. The model which is proposed is sensitive to the situations typical for the early stages of Software development. This one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. In his research paper he describes the advanced parametric models for assessment and prediction of software reliability which is based on statistics of bugs at the initial stage of testing. The parametric model approach, commonly associated with reliability issues, deals with the evaluation of the amount of bugs in the code. The proposed models are sensitive to the situations typical for the early stages of Software development. As a result, one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. To support the optimization of such complex models, the Cross-Entropy Global Optimization Method is proposed. Some authentic numerical examples are considered to demonstrate the efficiency of the proposed models. The model which is proposed is sensitive to the situations typical for the early stages of

Software development. This one deals with the essentially non-linear, multimodal goal function to define the optimal value as the estimation of the unknown control parameter. In his research paper he describes the advanced parametric models for assessment and prediction of software reliability which is based on statistics of bugs at the initial stage of testing. The parametric model approach, commonly associated with reliability issues, deals with the evaluation of the amount of bugs in the code.

Akira Hada (2011) describes in his research paper which presents various models to estimate reliability for a future profile with increased stress using the current observations to develop a model for future reliability.

For the foreseeable future the equipment itself will not change, and thus the increase of the load or stress, that the system is exposed to, may decrease reliability and increase maintenance. The model is required to determine the impact of these future profiles.

This paper was affected by an actual application modelling the current and anticipated future reliability of naval aircraft launch and recovery equipment .System reliability models are presented and demonstrated to anticipate system reliability and availability for changing and increasing applied loading distributions. In his research paper he described that these models are intended for systems whose components are exposed to predictable and quantifiable, but changing loading patterns. In the intended application, the models will be used to estimate the future reliability of naval aircraft catapult and arresting gear when subjected to different air wing compositions.

Each air wing usage profile is potentially different, forming a distribution of usage stresses, and this distribution is shifting with time, as aircraft weight and missions change. The current ALRE systems will continue to be used for the foreseeable future and the model is required to predict future performance and to identify the most unreliable components or problems within the existing design. In his research paper he defined that Weibull .distribution models are used in the typical fashion to model component failure times, but the initial Weibull distribution parameters are mathematical functions of the current, known applied stress distributions.

In his research paper he described that these models are intended for systems whose components are exposed to predictable and quantifiable, but changing loading patterns. Each air wing usage profile is potentially different, forming a distribution of usage stresses, and this distribution is shifting with time, as aircraft weight and missions change. The current ALRE systems will continue to be used for the foreseeable future and the model is required to predict future performance and to identify the most unreliable components or problems within the existing design.

Kuei-Chen Chiu (2011) described that over the last two decades, various software reliability growth models (SRGM) have been proposed, and there has been a gradual but marked shift in the balance between software reliability and software testing cost in recent years.

Shuanqi Wang (2011) describes that in order to incorporate the effect of test coverage, two novel software reliability growth models (SRGMs) are proposed in this paper using failure data and test coverage simultaneously. One is continuous using testing time, and the other is discrete with respect to the number of executed test cases instead of testing time. Since one of the most important factors of the coverage-based SRGMs is the test coverage function (TCF), we first discuss a discrete TCF based on Beta function.

Then **Shuanqi Wang** developed mean value functions (MVF) of the two models integrating test coverage and imperfect debugging. Finally the proposed TCF and MVFs are evaluated and validated on actual software reliability data collected from real software development projects. The results demonstrate clearly that both the proposed TCF and SRGMs provide better estimation and fitting for the data sets under comparisons.

She describes that recently an important trend in developing SRGMs is incorporating some additional information such as test coverage into the SRGMs. This is because that only the faults located in the covered constructs of the software can be exposed in the testing process. In this paper, to describe the relationship between reliability and test coverage, two novel NHPP SRGMs based on test coverage are proposed.

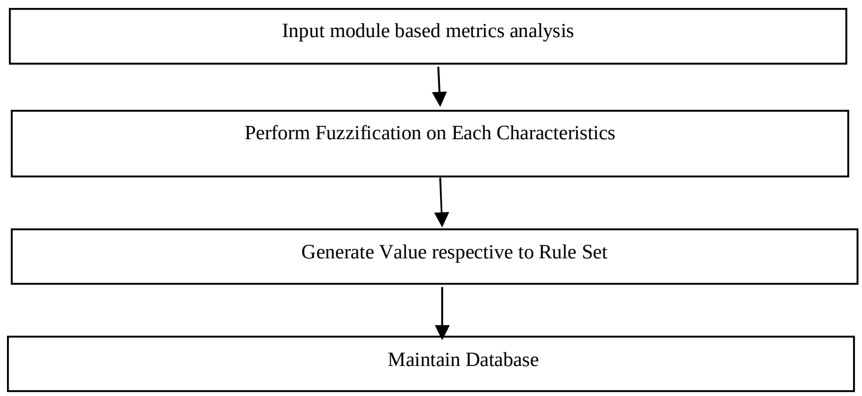
In this he discussed a discrete TCF with respect to the number of test cases based on Beta function. Then we develop discrete and continuous SRGMs integrating test coverage. The proposed TCF and SRGMs have been evaluated and validated on one published failure data set and a testing project. It is found that both the proposed TCF and SRGMs provide better reliability fitting and estimation power for the data sets under comparisons.

It shows that the proposed approach of software reliability modelling incorporating test coverage is an effective and novel attempt, which may be very promising for further researches and applications.

He developed mean value functions (MVF) of the two models integrating test coverage and imperfect debugging. Finally the proposed TCF and MVFs are evaluated and validated on actual software reliability data collected from real software development projects. The results demonstrate clearly that both the proposed TCF and SRGMs provide better estimation and fitting for the data sets under comparisons.

III. PROPOSED WORK

The use of fuzzy logic improved association mining to predict software reliability has been proposed. Software reliability assessment has been a vital factor to characterize the quality of any software product quantitatively during testing phase. The work is based on the software failures or the defects and on which the analytical decision will be drawn using fuzzy logic. It only takes failure history as input and Predicts future failures. The input to the proposed method is software execution time, whereas output of the system is predicted as number of failures. The failures or the errors will be defined with different weights. Therefore, here we explore the applicability of fuzzy logic for better prediction of reliability in a realistic environment and present an assessment method of software reliability growth using connectionist model. The main methodologies included in this work are the association mining and the fuzzy logic.



Ensuring the reliability of a software project is important to all parties involved including Managers, Marketing, Programmers, and Customers. Unreliable systems can impact software developers and consumers by simply being an annoyance, by costing time and money, or worst case scenario, by costing single or multiple lives. Everybody involved in the software process has reasons for desiring reliable software.

Some of the specific features of using fuzzy logic approaches in software reliability prediction are:

• It is easy to design and construct models for reliability growth of varying complexity at different points of time for a given data set.

• Fuzzy Logic based models are easily adaptable in complex operational phase for different failure datasets.

•Software reliability assessment has been a vital factor to characterize the quality of software product quantitatively during testing phase. Over the years many analytical model have been proposed for modelling software reliability growth trends with different predictive capabilities at different phases of testing. Yet we need to develop such single model that can be applied for accurate prediction in all circumstances. Here we explore the applicability of fuzzy logic models for better prediction of reliability in a realistic environment and present an assessment method of software reliability growth using connectionist model. we can select a particular model first by generalizing the applicability of application used and second by developing adaptive models.

Use of fuzzy logic improved association mining to predict software reliability has been proposed. Software reliability assessment has been a vital factor to characterize the quality of any software product quantitatively during testing phase. The work is based on the software failures or the defects and on which the analytical decision will be drawn using fuzzy logic. It only takes failure history as input and Predicts future failures. The input to the proposed method is software execution time, whereas output of the system is predicted as number of failures. The failures or the errors will be defined with different weights. Therefore, here we explore the applicability of fuzzy logic for better prediction of reliability in a realistic environment and present an assessment method of software reliability growth using connectionist model.

Ensuring the reliability of a software project is important to all parties involved including Managers, Marketing, Programmers, and Customers. Unreliable systems can impact software developers and consumers by simply being an annoyance, by costing time and money, or worst case scenario, by costing single or multiple lives. Everybody involved in the software process has reasons for desiring reliable software.

Some of the specific features of using fuzzy logic approaches in software reliability prediction are:

• It is easy to design and construct models for reliability growth of varying complexity at different points of time for a given data set.

• Fuzzy Logic based models are easily adaptable in complex operational phase for different failure datasets.

•Software reliability assessment has been a vital factor to characterize the quality of software product quantitatively during testing phase. Over the years many analytical model have been proposed for modeling software reliability growth trends with different predictive capabilities at different phases of testing. Yet we need to develop such single model that can be applied for accurate prediction in all circumstances. Here we explore the applicability of fuzzy logic models for better prediction of reliability in a realistic environment and present an assessment method of software reliability growth using connectionist model. we can select a particular model first by generalizing the applicability of application used and second by developing adaptive models.

IV. RESULT

The presented work is implemented in matlab 7.8 environment. The results obtained from the system are given as under

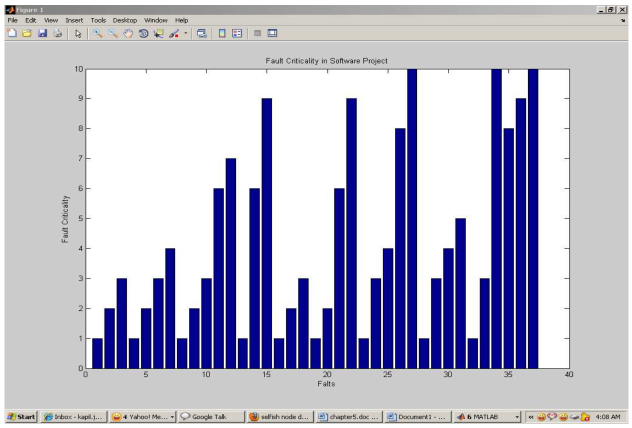


Figure 1 : Fault Occurrence in Software Project

Here figure 1 is showing the results of fault occurrence in a software project. Here the fault criticality is between 0 and 1. Higher the bar more critical the fault is.

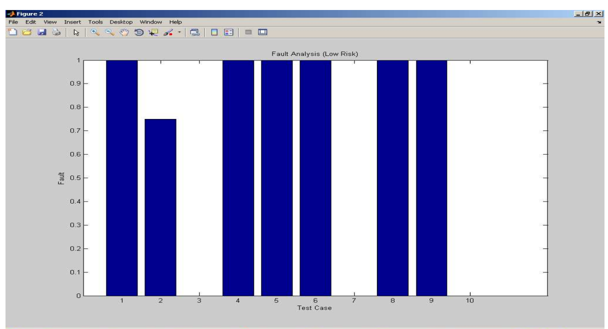


Figure 2: Fault Criticality in Software Project (Less Critical)

Here figure 2 is showing the results of low fault criticality in a software project.

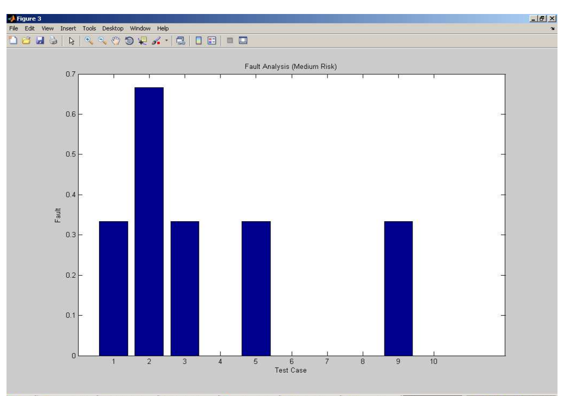


Figure 3: Fault Criticality in Software Project (Medium Critical)

Here figure 3 is showing the results of low fault criticality in a software project.

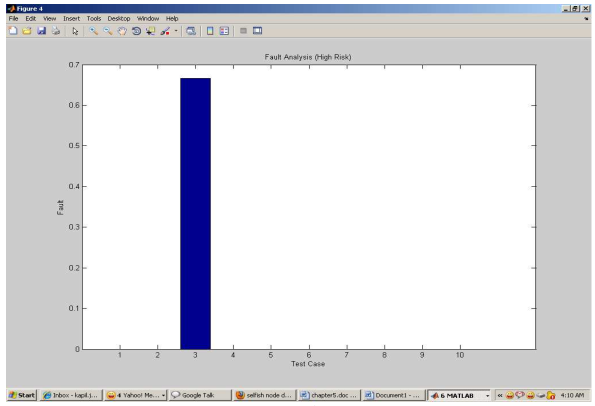


Figure 4: Fault Criticality in Software Project (High Critical)

Here figure 4 is showing the results of low fault criticality in a software project.

V. CONCLUSIONS

To perform all kind of measurements about the software such as quality, time, cost etc, there is the requirement of analysis of software metrics. Software metrics provides a mathematical approach to measure the software. The process of predicting the effort required to develop a software system is known as software cost estimation. Many models for estimation have been proposed over the last 30 years. In this present work we have defined a fuzzy based software quality estimation approach to analyze the software criticality. The proposed system is about the analysis of an object oriented software system.   In this work a layered approach is defined to identify the software criticality on individual factor and later on association mining is implemented to perform the combined analysis. The obtained results classify the available software modules under different criticality levels.

ACKNOWLEDGEMENT

I am very grateful to my guide/supervisor Mr. Rahul Kumar Yadav for his help and able guidance for the thesis. I am very thankful to my Institute, for providing me resources and facilities to help in this work.

REFERENCES

[1] Sultan Aljahdali," Improved Software Reliability Prediction through Fuzzy Logic Modeling". The ISCA 13th International Conference on Intelligent and Adaptive Systems and Software Engineering, Nice, France, July 1-3, 2004.pages 17-21, ISCA, 2004.

[2] Sultan Aljahdali, Alaa F. Sheta: Predicting the Reliability of Software Systems Using Fuzzy Logic. ITNG 2011: 36-40

[3] Khalaf Khatatneh," Software Reliability Modelling Using Soft Computing Technique", European Journal of Scientific Research ISSN 1450-216X

[4] GUO JUNHONG," Software Reliability Nonlinear Modelling and Its Fuzzy Evaluation", 4th WSEAS Int. Conf. on NON-LINEAR ANALYSIS, NON-LINEAR SYSTEMS and CHAOS

[5] Sultan H. Aljahdali," Employing four ANNs Paradigms for Software Reliability Prediction: an Analytical Study", ICGST-AIML Journal, ISSN: 1687-4846

[6] Ajeet Kumar Pandey," Fault Prediction Model by Fuzzy Profile Development of Reliability Relevant Software Metrics", International Journal of Computer Applications (0975 – 8887)

[7] Jie Yang, (2008) "Managing knowledge for quality assurance: an empirical study", International Journal of Quality & Reliability Management, Vol. 25 Issue: 2, pp.109 - 124

[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, (Nov 2009)" Critical Review on Software Reliability Models", International Journal of Recent Trends in Engineering ISSN 1797-9617 Volume 2, Number 3

[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" ISSRE '97 ISBN: 0-8186-8120-9

[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 (e ISSN: 1309–1190)
[14] Qiuying Li," A Software Reliability Evaluation Method", 2011 International Conference on Computer and Software Modeling IPCSIT
[15] N. Raj Kiran," Software reliability prediction by soft computing techniques", The Journal of Systems and Software

