Fuzzy Based Software Reliability Estimation

Feroza Haque, Dr. Sanjeev Bansal
1,2 Amity University, Noida, UP, India

Abstract
Software reliability is one of the important factors as described by all the Quality models. In many case we have seen that due to software failure many hazardous result has been faced by the world. So a step should be taken to avoid this type of failure and predict the reliability of the software. Many reliability models and integrated tools have been designed for the same purpose. In this paper, we have define a methodology to estimate the reliability on the basis of the external metrics. Software reliability and evaluation methods are different approaches for dealing with the problem. The paper will introduce the two models also. We will also introduce the fuzzy metrics that will be used to evaluate the reliability.

Keywords
Estimation Model, Fuzzy System, FRSQ, Software Quality

I. Introduction
Software reliability is a area of software engineering which deals with the failure free execution of a software. According to ASCI “Software Reliability is defined as probability of failure free software operation for a specified period of time in a specified environment” [4]. Software reliability is one of the major factor that affect the quality of any software. IEEE defines software quality as the degree to which a system component or process meets customer requirement [3]. Whenever we design a software we generally consider the following principle [7]:

- Customer requirement.
- Establish standards for designing good quality software.

Generally while defining the quality of any software we consider a direct and indirect approach. In direct approach we apply the internal metrics for example KLOC, Function point etc. Where as in indirect method we design external metric for usability, maintainability and reliability. So we can say that reliability is one of the major factor for evaluating software quality.

Software Reliability Engineering (SRE) is the discipline that deals with the quantitative study of the operational behavior of software based system with respect to the user requirement [2]. This includes:-

- Software reliability measurement, which includes estimation and prediction with the help of software reliability models established in the literature.
- The consideration of the reliability metrics and implication of those metrics in evaluating the software reliability.
- The application of this knowledge in specifying and guiding system software architecture, development testing, acquisition, use and maintenance.

Based on the above discussion we will first discuss the reliability model and then the metrics been considered. Software reliability models can be broadly classified as estimation model and prediction model [1].

A. Estimation Model
This method is used after the designing of the software. This is done on the basis of the data been collected from the present model. This is applied in the last phase of the SDLC life cycle. For example:- exponential distribution models, Weibull distribution model, Thompson and Chelson's model, etc.

B. Prediction Model
This method is used Before designing of the software. The evaluation is done on the basis of the past history collected from different software. This is applied in the early phase of the SDLC life cycle. Using prediction models, software reliability can be predicted early in the development phase and enhancements can be initiated to improve the reliability. For example:- Musa's Execution Time Model, Putnam's Model. and Rome Laboratory models TR-92-51 and TR-92-15, etc.

For the purpose of measurement we use metrics. Metrics are used to evaluate the quantity of any attribute and is considered as one of the efficient way for making any estimation. Software reliability metrics used by all the models for measurement are defined as below [1]:

1. Product Metrics
Software size is thought to be reflective of complexity, development effort and reliability. Lines Of Code (LOC), or LOC in thousands (KLOC), is an intuitive initial approach to measuring software size.

2. Project Management Metrics
Research has demonstrated that a relationship exists between the development process and the ability to complete projects on time and within the desired quality objectives. Costs increase when developers use inadequate processes. Higher reliability can be achieved by using better development process, risk management process, configuration management process, etc.

3. Process Metrics
Based on the assumption that the quality of the product is a direct function of the process, process metrics can be used to estimate, monitor and improve the reliability and quality of software. ISO-9000 certification, or “quality management standards”, is the generic reference for a family of standards developed by the International Standards Organization (ISO).

4. Fault and Failure Metrics
The goal of collecting fault and failure metrics is to be able to determine when the software is approaching failure-free execution. The failure data collected is therefore used to calculate failure density, Mean Time between Failures (MTBF) or other parameters to measure or predict software reliability.

In this paper our main concern will be the estimation model and how it can be designed. The metrics measurement we will use will be the external metrics related to process and fault failure metrics. This paper has quantified the metrics in fuzzy logic. Fuzzy logic is a inference rule based system which deals with complex problem in absence of complete knowledge. In fuzzy we have different type of linguistic variable like almost , many, some of etc. And this are the general term that we use when we are uncertain and is not able to evaluate the result exactly. Fuzzy is capable of estimating a single scalar output corresponding to all the above linguistic terms. Linguistic variable is a term used in logic programming to measure the quantity or quality of any
system. For example if we have a scenario like “Reliability of the software is not much”. Then by using fuzzy inference rule we can easily evaluate the output of the statement. The whole paper is designed in four sections. The section I of the paper has introduced the concept on which the paper is based. The section II of the paper will introduce the model and metrics that we will consider in the paper. The section III will describe in brief the proposed methodology and the last section will describe the fuzzy system.

II. Model and Metrics

A. Estimation Model
Estimation models are used to evaluate the reliability of the software on the basis of data been collected from the current software. This is a method that we implement at the later phase of the SDLC life cycle. A number of model has been already been design for this purpose. The main purposes of the estimation model are:-

1. To estimate the time needed to test the software to achieve a specified objective.
2. To estimate the expected reliability of the software when the testing is finished.

The estimation models are also known as Software Growth Model (SGM).

B. Metrics
Software metrics are used for the purpose of measurement of any attribute. Generally an metrics can either be an internal metric or external metrics. An internal metric deal with the internal representation of the software. For example:- KLOC , Function point etc. An external metric deal with the features that affect the external interface or performance of the software. For example:- failure count, documentation etc. In this paper we have considered the external metric for evaluating the reliability of the software. Given below are the attributes of reliability as defined by ISO/IEC 9126 model along with the external metrics used to evaluate them.

1. Maturity
This describes the frequency of failure of the software by faults. The metrics defined are mentioned below:-

(a). Number of successful version released
The factor will be used to calculate the rate of successful version released for particular software. It can be calculated as SVR (Successful version released)= No of successful version/ Total no of version released.

(b). Level of CMM
This factor deals with organization. We have to check that the organization belongs to which level of CMM (Capability maturity model)If a organization is of CMM level 4,5 then it will be considered to be more reliable.

2. Fault Tolerance
This evaluates the robustness of the software. It describes the software attributes that describe the ability of the software to maintain a specified level of performance in cases of software faults or the violation of its specified interface. The metrics defined are mentioned below:

3. Exception Handling
This a situation where program responds to an undesirable situation. For example: “ X/0”(A variable divided by zero). If a software is able to perform exception handling then it will be considered more reliable the other.

- Capability of detecting faults: A fault may be a logical or semantic error that exists in a program.
- Percentage of functionality successively met: FS(functionality successively met)=Total no of functionality met/ total no of functionality.

4. Recoverability
Describes the capability of the software to re-establish its level of performance and to recover the data directly affected in case of failure and the time and effort needed for it. The metrics defined are mentioned below:-

1. Capability to recover from failure: This is the ability of software to tolerant failure.
2. Back of data: This parameter tells us how the availability of the backup of data affects recoverability. If software has a data backup facility available, the recoverability of the software is very high, otherwise it is low.

5. Reliability Compliance
This determines whether the software adheres to the compliance standards of reliability or not.

1. Capability to achieve the requirement been defined: The parameter tells us how adherence of the software to reliability compliance standards affects reliability compliance. It can be calculated by using the equation below:

RC(Reliability Compliance)= No of requirement meet/total no of requirement.

III. Proposed Methodology
In this paper we have defined a very simple method for the measurement of any attributes. This attributes are defined by ISO/IEC 9241 in 1998 [5]. Generally if design any software then we consider two facts

A. Personal Satisfaction
This defines the comfort and acceptability of use. This factor is concern with user perception about the software. This reflects the likeliness of the user for particular software. Based on that we have identified two metrics as given below:-

1. Recommendation
This describes how many users has recommended the software for use. This can be calculated as

RC (Recommendation)=(No of user recommended/ Total no of user been communicated)*100.

2. Popularity

(i). Efficiency
This defines the effective usage of the software as per the requirement is defined. To evaluate this we have designed different attributes as discussed in the previous section in metrics. Based on that we have defined further sub attributes or metrics. An abstract view of all the metrics are shown as below.
Table 1: Reliability attributes and sub attributes

<table>
<thead>
<tr>
<th>SNO</th>
<th>SUB ATTRIBUTE OF RELIABILITY</th>
<th>REAL METRICS</th>
</tr>
</thead>
</table>
| 1   | MATURITY                    | 1) Number of successful versions [4]  
|     |                             | 2) Level of CMU. |
| 2   | FAULT TOLERANCE             | 1) Exception handling [6]  
|     |                             | 2) Capable of detecting the fault.  
|     |                             | 3) Percentage of functionality met. |
| 3   | RECOVERABILITY              | 1) Capability to recover from failure. |
|     |                             | 2) Basis of data [6]. |
| 4   | RELIABILITY COMPLIANCE      | 1) Capability to achieve the requirements been defined. |

After identifying the attributes we calculate the result by obtaining the net effective result which is calculated as

\[
\text{Reliability} = (\text{User Satisfaction} + \text{Efficiency})
\]

The result will be calculated as the sum of the user satisfaction and efficiency. The net result is calculated the weighted sum of the factors. The metrics are quantified on the basis of survey FRSQ (Fuzzy Reliability Survey Questionnaire). We have used different linguistic variable as will be discussed in the next section.

The fuzzy methodology is explained as below

Step 1. Identify the software quality model suited for the problem. In this case we have considered the ISO / IEC 9126 model. After that evaluate the Reliability factors defined by the model.

Step 2. Based on the factors identify the real time metrics. For this purpose a questionnaire is design and calculates the real metrics result.

Step 3. Collect the data.

Step 4. The calculate the result by calculation the aggregate result

\[
\sum^N_{i=1} \text{factori}
\]

IV. Fuzzy System

Fuzzy system is a rule based system where the result is calculated on the basis of the uncertain facts of the system. In world generally many situations arises with the output is uncertain like “Almost”, “Around”, “Not so clear” etc in that case fuzzy is used. Fuzzy logic is a soft technique which deals with uncertainty. The concept was first introduced by Lofti A Zadeh in the year 1965 in his seminar paper presentation. It was first introduced as a concept and now the world is using it to deal with complex problem.

In the past we have seen many adverse result of software failure like 2YK problem, Arina Space problem etc. So it is required to design a integrated tool to deal with this type of failure. Fuzzy logic and neural networks are been the major computing technique used in software engineering. The fuzzy logic is a inference rule based system where we uses “If-else” statement. The “if – else” statement are calculate on some metrics form and then the result is evaluated. A simple inference rule example to calculate reliability is “If the failure rate is Negligible and data recovery is High then the reliability is More”. In this example we have use three linguistic variable ie HIGH, NEGLIGIBLE and MORE for which we will design the fuzzy set. The overall fuzzy process done in this paper is represented in the figure given below.

V. Conclusion

This paper has revealed the fact that reliability is one of the important for judging the quality and performance of the software. The questionnaire system FRSQ is a very efficient in evaluating the performance since every parameter is considered precisely. The evaluation method is very simple and effective. But the system has some limitation also because been a survey based study the results vary from user to user. Fuzzy logic has proven itself as an very effective in identifying better solution in any situation.

References

Feroza Haque is senior lecturer in the department of computer science and engineering in NIET engineering and technology and pursuing M.Tech from Amity University, Uttar Pradesh. Her research area of work is evaluation of software reliability.

Dr. Sanjeev Bansal is director of MBA in Amity University, Uttar Pradesh. He is a highly respected faculty member of the college and head of the Phd division of school.