Estimation of Software Quality by using fuzzy (FIS)

Pinky Chandwal, A. S. Zadgaonkar, Abhinav Shukla

Abstract—As software are being used in more and more critical areas, therefore quality of software becomes a very important factor for business and human safety. Estimation of software quality is the key for achieving a high quality product. Quality of software is associated with number of quality attributes and estimation of software quality involves broad views and various perspectives which might involve natural description, in linguistic terms. Linguistic terms are more convenient to use when human express the subjectivity and imprecision of their evolution but these linguistic variables involve ambiguity and vagueness. Since fuzzy logic deals with the ambiguity, imprecision and vagueness therefore this study proposes the applicability of fuzzy along with ISO 9126 quality model for developing quality estimation framework.

Index Terms— Quality, Quality attributes, FIS, Rule base, linguistic variable.

I. INTRODUCTION

Software engineering is driven by three major factors that are Cost, Schedule and Quality. Today quality is main mantra and business strategies are designed around quality. There are many definitions of quality some of them are:-

Acc. to Heritage Dictionary Quality is “A Characteristics or attributes of something” [1]. Acc. to Pressman Software Quality is “ensuring that software organization does the right things at the right time in a right way” [1]. Acc. to McCall Software Quality is “a general term applicable to any trait or characteristics whether individual or generic, a distinguishing attribute which indicate a degree of excellence or identifies the basic nature of something” [2]. ISO 8402 define Software Quality is “totality of Characteristics of an entity that bear as its ability to satisfy stated and expected needs” [3]. Developing high quality software is one of the fundamental goals of software engineering. Because now-a-days Software are used in almost every field like Telecommunication, Genetic Engineering, Education, Medical, Transportation, Entertainment, Industrial Process, Office Products and many more. Software Defects removal are very costly in terms of money and any compromise with the quality may result in the loss of reputation and life; Ariane 5 crash June 4, 1996: Maiden Flight of European Ariane 5 launcher crashed about 40 sec. after takeoff due to software error and lost was about half a million dollars[4]. Software that is delivered on time and within budget and also perform its stated function can’t be considered as a high quality product because of many reasons. Some of the reasons are:-

1) Software product may not be user friendly so difficult to understand and use.
2) Software product may not provide any security to data and programs and may be misused.
3) Software product may be difficult or impossible to modify or maintain.
4) Software product may not be compatible with other programs and software in the system so, fail to interact with them.
5) Software product may not be machine independent (portable).
6) Software product may not be as efficient as expected by the client.

From the above discussion we can conclude that Software Quality does not depend upon only budget & schedule but also on so many other factors or attributes. And software quality is the degree to which software possess a desired combination of quality attributes.

II. REVIEW LITERATURE

In the context of software quality estimation different researcher have proposed different methods, techniques and mechanism, time to time, to estimate the software quality. To our knowledge there have been almost no or very little effort devoted to fuzzy logic for the estimation of software quality.

Jyothi G and Ch.Verra babu presents a technique for modeling code readability based on the judgment of human annotators and investigates its relationship to software quality for measuring software quality [5].

An approach for quality estimation is a suite of in-process metrics(strew metric suite) that leverages the software testing effort to provide an estimation of potential software field quality in early software development phases and the identification of low quality software program [6].

Krazysztof Sach provides a mechanism for software quality evaluation. The evaluation is based on set of criteria that are decided by stating questions and finding answer to those questions [7].

Another approach for quality estimation is based on clustering. Clustering technique along with the help of software engineering human expert is used for building a software quality estimation system. The system first clusters hundred of software modules in to small number of coherent group and presents the representative of each group to a software quality expert,
Estimation of Software Quality by using Fuzzy (FIS)

who labels each cluster as either fault-prone or not fault prone based on his domain knowledge as well as some data
statics[8].

Yaun et al also use clustering but with fuzzy to predict the
number of faults in software module [9].

A method based on unsupervised learning and clustering is
also proposed for quality estimation. An unsupervised
approach for extracting principal components from software
measurement data use Artificial Neural Network [10]. This
study focuses on extracting the principal components for
software measurement data. The extracted component are
then used to train a supervised neural network based software
quality classification model for predicting the quality of
software modules as fault prone or not fault prone.

ISO 9126 Quality Model

A series of attempts to define the software quality has been
made from time to time; say McCall Quality Model [2],
Boehm’s Quality Model [11] and many more. For this study,
we choose ISO 9126 Quality Model [12].The International
Organization for Standardization release ISO 9126 , which is
a widely accepted standard for measuring quality of software.
It is most widely accepted because it includes both internal
and external quality characteristics of a software product.

ISO 9126 standard defines six criteria called functionality,
reliability, usability, efficiency, maintainability and
portability. Each of these criteria consists of several sub
criteria as shown in figure A:

- Each quality factors is defined as follows:
  1) Functionality: A set of attributes that relate to the existence of
     a set of functions and their specified properties. The functions
     are those that satisfy stated or implied needs.
  2) Reliability: A set of attributes that relate to the capability of
     software to maintain its level of performance under stated
     conditions for a stated period of time.
  3) Usability: A set of attributes that relate to the effort needed
     for use, and on the individual assessment of such use, by a
     stated or implied set of users.
  4) Efficiency: A set of attributes that relate to the relationship
     between the level of performance of the software and the
     amount of resources used, under stated conditions.
  5) Maintainability: A set of attributes that relate to the effort
     needed to make specified modifications.
  6) Portability: A set of attributes that relate to the ability of
     software to be transferred from one environment to another.

III. METHODOLOGY

Fuzzy Logic

The term “fuzzy logic” was introduced with the proposal of
fuzzy set theory by Lotfi A Zadeh in 1965. Fuzzy logic is based
on the concept of fuzzy sets. Fuzzy logic deals with the
vagueness and uncertainty. Fuzzy set provides a mathematical
framework in which vague conceptual phenomena can be
handled efficiently. Fuzzy set theory is a suitable tool to
reinforcement the comprehensiveness and correctness to
decision making stages: Fuzzy set theory is an important
approach to measure the uncertainty of concepts that are
associated with human beings subjective judgements in terms
of linguistic variables that are often vague. Conceptual theory
distinguishes between those element that are the members of
the group and those that are not, there being very clear or crisp
boundaries. It has two-value logic either true (1) or false
(0). In contrast with the traditional set theory, fuzzy set can
have varying values that ranges between 0 to 1. The central
concept of the fuzzy set theory is the membership function .The membership function is a graphical representation of the magnitude of participation of each input.

An element of a fuzzy set can be full member (100%
membership) or a partial member (between 0% to 100%
membership) that is, membership value of an element is no
longer restricted to just two values but can be any value
between 0 to 1; So that we can draw definite conclusion from
vague and imprecise information[13].

A. Membership functions

There is different membership function, some of them are:-

1) Triangular membership function

Consider the temperature and plot the membership function for
high, if we plot triangular membership function; we get a
graph shown in figure below:
2) **Trapezoidal membership function**

Consider the temperature and plot the membership function for low, if we plot trapezoidal membership function; we get a graph shown in figure below:

![Trapezoidal membership function](image1)

3) **Gaussian membership function**

Consider the temperature and plot the membership function for medium, if we plot Gaussian membership function; we get a graph shown in figure below:

![Gaussian membership function](image2)

B. **Fuzzy set operations**

1. **Intersection of two fuzzy sets.**

\[ \mu_{A \cap B}(x) = \mu_A(x) \land \mu_B(x) \]

Or

Often \[ \min[\mu_A(x), \mu_B(x)] \]

![Intersection of two fuzzy sets](image3)

2. **Union of two fuzzy sets.**

\[ \mu_{A \cup B}(x) = \mu_A(x) \lor \mu_B(x) \]

Or

\[ \max[\mu_A(x), \mu_B(x)] \]

![Union of two fuzzy sets](image4)

3. **Complement.**

\[ \mu_{\neg A}(x) = 1 - \mu_A(x) \]

![Complement](image5)

C. **Fuzzy inference system**

It is the process of formulating the mapping from a given input to an output using rule base.
Estimation of Software Quality by using Fuzzy (FIS)

1) Fuzzification
   To convert the crisp set in to fuzzy set.
2) Fuzzy Rule base
   It consists of a set of antecedent-consequent linguistic rule in the form of:
   If (temperature is high) and (fan is slow) then cooling is started.
   If (temperature is low) and (fan is fast) then cooling is stopped.
3) Defuzzification
   To extract the crisp value from fuzzy set
   Mat lab Fuzzy Logic Toolbox
   For implementation we use Mat lab environment with fuzzy logic tool box [14].

   Figure9. Mat lab Fuzzy Logic Toolbox

1) Fuzzy Editor
   It handles issues like name and number of input, output variables.
2) Rule Editor
   It is used for developing and editing the list of rules that describes the behavior of the system.
3) Membership Function
   It is used for defining and editing membership value and their shape associated with each variable.
4) Rule and Surface viewer
   They are strictly read-only tools and are used for looking at, as opposed to editing to FIS.

IV. FRAMEWORK DEVELOPMENT FOR QUALITY ESTIMATION
   For this study input parameters are the different attributes of quality that are defined by the ISO 9126 model.
   INPUT PARAMETER          OUTPUT PARAMETER
1. Functionality            1. Quality
2. Reliability
3. Usability
4. Efficiency
5. Maintainability
6. Portability

| Table1. Show input parameters and their range with respect to linguistic Variables |
|-----------------|-----------------|-----------------|-----------------|
| INPUT           | INPUT NAME      | LINGUISTIC      | RANGE           |
| INPUT 1         | FUNCTIONALITY   | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |
| INPUT 2         | RELIABILITY     | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |
| INPUT 3         | USABILITY       | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |
| INPUT 4         | EFFICIENCY      | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |
| INPUT 5         | MAINTAINABILITY | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |
| INPUT 6         | PORTABILITY     | LOW             | 0-40            |
|                 |                 | MEDIUM          | 20-60           |
|                 |                 | HIGH            | 50-100          |

| Table2. Show output parameter and its range with respect to linguistic Variables |
|-----------------|-----------------|-----------------|-----------------|
| OUTPUT          | OUTPUT NAME     | LINGUISTIC      | RANGE           |
| OUTPUT 1        | QUALITY         | POOR            | 0-50            |
|                 |                 | GOOD            | 35-95           |
|                 |                 | BEST            | 90-100          |

A. Fuzzification
   It is the process of converting crisp value into a range of membership for linguistic terms of fuzzy set. The membership function is used to set the range for each linguistic variable.
The above figure shows that there are six input parameters and one output parameter that estimate the quality of the software product by developing and applying different rules to the FIS.

**B. Rule base**

According to the fuzzified input and output parameter rule base has been generated by applying our own reasoning as experts for making decision and drawing conclusion in order to estimate the quality of the software product. We develop 103 rules by using AND operator. Some of them are:

1. **Rule No.1** If (functionality is low) then (quality is poor)
2. **Rule No.12** If (functionality is medium) and (reliability is medium) and (usability is medium) then (quality is good)
3. **Rule No.22** If (functionality is medium) and (reliability is medium) and (usability is medium) and (efficiency is medium) and (maintainability is medium) then (quality is good)
4. **Rule No.53** If (reliability is high) and (usability is high) and (efficiency is high) and (maintainability is high) and (portability is high) then (quality is best)
5. **Rule No.103** If (usability is medium) and (efficiency is medium) and (maintainability is medium) and (portability is medium) then (quality is good)

The above figure shows the Fuzzification of output parameter quality by defining name, shape and range of its membership functions that corresponds to its linguistic variables.
Figure 14. Inference process when functionality = 32.1, Reliability = 38.1, Usability = 32.1, Efficiency = 39.6 Maintainability = 35.1, Portability = 33.6 then Quality = 55.7

Figure 15. Inference process when functionality = 51.7, Reliability = 50, Usability = 56, Efficiency = 64.9 Maintainability = 51.5, Portability = 60.4 then Quality = 69.4

Figure 16. Inference process when functionality = 59, Reliability = 54.5, Usability = 56, Efficiency = 57.5 Maintainability = 53, Portability = 63.4 then Quality = 70.1

Figure 17. Inference process when functionality = 64.9, Reliability = 61.9, Usability = 67.9, Efficiency = 69.4 Maintainability = 59.2, Portability = 64 then Quality = 84.9
Table3. Show various inputs with their corresponding estimated quality

<table>
<thead>
<tr>
<th>INPUT</th>
<th>Functionality</th>
<th>Reliability</th>
<th>Usability</th>
<th>Efficiency</th>
<th>Maintainability</th>
<th>Portability</th>
<th>Quality</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. no</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INPUT 1</td>
<td>11.7</td>
<td>20.4</td>
<td>19.6</td>
<td>15.2</td>
<td>19.6</td>
<td>21.3</td>
<td>47.3</td>
</tr>
<tr>
<td>INPUT 2</td>
<td>32.1</td>
<td>38.1</td>
<td>32.1</td>
<td>39.6</td>
<td>35.1</td>
<td>33.6</td>
<td>55.7</td>
</tr>
<tr>
<td>INPUT 3</td>
<td>51.5</td>
<td>50</td>
<td>56</td>
<td>64.9</td>
<td>51.5</td>
<td>60.4</td>
<td>69.4</td>
</tr>
<tr>
<td>INPUT 4</td>
<td>59</td>
<td>54.5</td>
<td>56</td>
<td>57.5</td>
<td>53</td>
<td>63.4</td>
<td>70.1</td>
</tr>
<tr>
<td>INPUT 5</td>
<td>64.9</td>
<td>61.9</td>
<td>67.9</td>
<td>69.4</td>
<td>59.2</td>
<td>64</td>
<td>84.9</td>
</tr>
<tr>
<td>INPUT 6</td>
<td>81.3</td>
<td>75.4</td>
<td>72.4</td>
<td>78.4</td>
<td>75.4</td>
<td>81.3</td>
<td>95</td>
</tr>
</tbody>
</table>

Overall quality can be estimated by dividing the sum of the individual qualities (outputs) by no. of inputs.
Overall quality = \((47.3 + 55.7 + 69.4 + 70.1 + 84.9 + 95) / 6 = 70.4\)

V. CONCLUSION

The proposed work is summarized as the development of a framework based on fuzzy for the estimation of software quality. Since software quality is the totality of different quality attributes. Therefore, for choosing different quality attributes we select ISO 9126 model and define membership functions along with their shape, name and range that corresponds to the various linguistic variables for the chosen attributes and output quality. Then, we develop various rules for FIS to estimate the software quality.

REFERENCE

6. Nachippan, Laurie Williams, M laden vauk Jaso n Osborne-Early estimation of software quality using in process testing metrics; A controlled case study.
7. Krazysztof Sacha- Evaluation of software quality
8. Shi Zhong,Taghi M.Khosgoftaar and Naem Seliy- Unsupervised learning for expert based software quality estimation
9. X.Yuan,T.M.Khosgoftar,E. Allen and K.Ganesan-An application of fuzzy clustering to software quality prediction
11. Bohn, B.W. et al-Characteristics of Software Quality

AUTHORS PROFILE

Pinky Chandwal, did her MSC-IT from Baba Ghulam Shah Badshah University, Rajouri, J&K., India and Currently pursing M.Phil-IT From Dr. C. V. Raman University, Bilaspur, Chhattisgarh, India.

Dr. A. S. Zadgaonkar, Ph.d (Instru.), Ph.d (Materials), D. Lit (Speech Recog.) is vice chancellor of Dr. C. V. Raman University, Bilaspur Chhattisgarh. He has forty years of teaching and Administrative Experience. He has published more than 470 papers in International, National Journals/ Conferences. He has guided more than 10 Ph.d Candidates. He is author of 3 books. He has received more than 13 Award and 10 Research Grants. He is member of more than 15 societies.

Abhinav Shukla, Assistant Professor and HOD (IT) in Dr. C. V. Raman University, Bilaspur, Chhattisgarh. He did his MSc-IT from Guru Gasidass Central University Bilaspur, Chhattisgarh. M. Tech (IT) from KSOU. And M.Phil. from Dr. C. V. Raman University, Bilaspur, Chhattisgarh. He has more than 10 years of teaching experience. He has published more than 4 papers in National Journals.