ASSESSMENT OF SOFTWARE QUALITY USING MACHINE LEARNING TECHNIQUE

*Saumendra Pattnaik
**Binod Kumar Pattanayak

ABSTRACT

It is important to quantify the parameters which put impact on the quality of software. A broad literature survey is done by taking the literature survey is done by taking the leading quality models. This survey provides two different aspects like, predicting the software quality through machine learning techniques and quantifying the parameters which give impact on the software quality. The simplicity of the models is the common deficiency of this survey.

Keywords: *Software Quality, Quality Parameters, Fuzzy logic, Prediction Logic*

1. INTRODUCTION

Software Quality is defined as the scale of a system or a process which suits the acknowledge requirements [1]. Basically the quality of software expresses or calculates a system performance and explains the systems formulation and functioning. Software attributes can be classified in two different groups as internal and external [2]. The first type of attributes can be calculated in the course of the different stages of software development life cycle and also after releasing the product. The second type of quality attributes is multi dimensional in nature as different quality attributes are associated with other quality attributes and related with some internal quality attributes [3].

The software quality models are used to identify different external attributes beneficial for stake holders. The model also represents the functional relationship in between the external features which can be calculated. Instead of using the quality models for measuring the external features; after the fact, generally we use these models for predicting the features as fast as possible. Otherwise, we can say the model permits one for predicting the external features as a task for some variables that can compute the internal features. Here the internal features work as predictors for predicting the future external features, at the beginning stage of the SDLC [4]. Suppose if an external feature like maintainability is our concern, then the designer will be more concerned for predicting the maintainability of the design from coupling, cohesion point of view that can be calculated in the process of designing rather to waiting for evaluating the maintenance cost and work after getting the fact. While reviewing a particular external feature, the feature is knotted with two different tasks like:

1. Recognizing the attributes or the internal features supplying help to the proposed external features.
2. Distinguishing the functional attachments in between the external factors and the respective internal factors.

*Department of Computer Science and Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, Odisha, India
**Department of Computer Science and Engineering, Institute of Technical Education and Research, SOA University, Bhubaneswar, Odisha, India

There are some software quality models presented by the renounced researchers like Boehm, Dromey and McCall etc. These proposed models were used for calculating and describing the software quality in various points of views by explaining or estimating various software features. Yet these models present a more of skeleton to evaluate and predict the software quality rather than the solid models for application. The models don't supply the exact functional attachments in between the two different types of features. Using these models many trials have been projected in the survey for addressing the functional relations. Here a survey is presented which address two different views of these trials [5].

2. QUANTIFYING THE QUALITY PARAMETERS

Now a day's quantifying the constraints or bound influencing the quality of the software is a vital research topic. The factors are enumerated in different fields because of its outcomes on the quality of software. The models are distinguished in different forms:

A. **Fixed Model Approach**

Here this models gives a provision of some qualities in which the factors recognized by the customers in the sub division of the given model. For checking and restricting a specific factor, it needs a sub factors, the checking and the link with the given model [6].

B. **Specific Approach**

This approach is given to the model with the logic that if any of the quality factors are not properly identified. In the other way, the associated quality factors are distinguished for some exact system in the co-ordination with end users. The defined factors are crumbled into the quality attributes which can be checked and its measurements [7]. The crumbling will be checked with a given quality model. The link in between the factors and attributes can be given in two ways:

1. The indirect proposed model that is inherited by default.
2. A direct proposal model which can be checked through the stakeholders.

3. PREDICTION THROUGH PROPOSED TECHNIQUE

Machine learning is a unique technique which is used for solving many problems and helps in constructing the new algorithms for enhancement in the functioning. Utilizing this technique in software engineering gives the quality outcomes. So many different techniques were applied in software engineering for predicting. For example: Support Vector Machine technique is utilized for predicting some quality parameters. Some networking predicting models like naïve bayes network is utilized to predict in software engineering for integration of real time data and skilled person opinions. So many authors have proposed prediction models using Fuzzy rules for checking designing defects or problems in software. As the link between the different quality factors is tightened by uncertainty or different rules, here some steps has been taken for utilizing the usability of Fuzzy logic for predicting software qualities [8]. Another author grades the quality of software by a Fuzzy application method. Some different approaches has been done for estimating the maintenance capacity of a software, but out of these approaches no effort has been done towards the transparency of the model as a target. The previous models basically are not so clear enough for any layman to get there required knowledge.

4. METHODOLOGY

Clearness of a model indicates the characteristics in which the functional linkage can be recognized by the end user and it give rise to two outcomes like,

1. Skilled Knowledge

2. Unseen Knowledge of the data

Here the professional should be able to change the structure of the model if necessary. Openness is an essential factor in making the models valuable. It is because of the skilled person inputs to the data or knowledge in model building. Past data gives some valuable information from the previous projects. The skilled person utilizes the knowledge for giving valuable relations in between the different quality factors. The skilled person utilizes the different imprecise values or calculations. Fuzzy logic gives the valuable characteristics in comparison to other methods as it can represented through imprecise data and some other technique also. For clearness the different Fuzzy logic ways could be transformed, but it can be wrong also. The Fuzzy skeleton used for predicting some factors in need of correct example for clearness [9].

For getting clarity in the views we can take an actual image. In that image we can insert different Fuzzy values for defining the individual inputs. The Fuzzy rule gives an idea for each input. The Fuzzy rule gives an idea for each input which while combined gives different outcomes. An input X of Fuzzy type can be utilized in the same rules with some skilled operations [10]. Here a natural course for constructing the prediction model using Fuzzy logic is done. The maintenance of these models is checked through the different parameters for feature consideration.

5. CONCLUSION

Here, a review is done on some predicting models and its views. The intension is for checking the utilization of the information for creating different models. The proper association of information gives rise to clearness of the construction of the model and also this review gives an idea that checked models lacks in clearness.

REFERENCES

1. *Gillies and A. C, 1992, Software Quality: Theory and management, London: Chapman and Hall.*
2. *Standard, I.I., ISO-9126,1991, Software Product Evaluation – Quality Characteristics and Guidelines for Their Use.*
3. *Fenton, N.E. and S.L. Pfleeger,1998, Software Metrics: A Rigorous and Practical Approach, Boston, MA, USA PWS Publishing Co.*
4. *Boehm, B.W., et al. ,1978, Characteristics of Software Quality,: North Holland Publishing Company.*
5. *McCall, J.A., P.K. Richards, and G.F. Walters, 1977, Factors in Software Quality I. Vol. I, and III, US Rome Air Development Center Reports - NTIS AD/A-049 014, NTIS AD/A-049 015 and NTIS AD/A-049 016,U. S. Department of Commerce., Editor.*
6. *Standard, I.I.I., 2001, Software Engineering - Product quality - Part 1: Quality model.*
7. *Dromey, R.G., 1995, A Model for Software Product Quality. IEEE Transactions on Software Engineering, 21, pp 146-162.*
8. *Trendowicz, A. and T. Punter, 2003, Quality Modeling for Software Product Lines. In: 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE'03).*
9. *Lamouchi, O., A.R. Cherif, and N. Levy, 2008, A framework based measurements for evaluating an IS quality. in Proceedings of the fifth on Asia-Pacific conference on conceptual modeling, Wollongong, NSW, Australia.*
10. *Zhang, D. and J.J.P. Tsai, 2003, Machine Learning and Software Engineering. Software Quality Journal, Vol 11, Issue 2: pp 87-119.*