

Software Quality Appraisal Using Multi-Criteria Decision Approach

Dr. Sanjay Kumar Dubey

Amity University/ CSE Dept., Noida, India
Email: skdubey1@amity.edu

Disha Sharma

Amity University/ CSE Dept., Noida, India
Email: disha.24sharma@gmail.com

Abstract—Software quality measurement is the key factor in the development of any software system. Various software quality models are devised to measure the performance of a software system, which consists of numerous quality parameters on the basis of which software are quantified. Different types of software quality models are already present like an ISO/IEC9126 Quality model, Boehm's model, McCall's model, etc. In this paper, an attempt has been made to increase the quality of a software system by introducing some new quality parameters in ISO/IEC9126 model. Since the quality parameters are very unpredictable in nature, so as to evaluate the performance of quality parameters, the fuzzy multi criteria approach has been used.

Index Terms—Software Quality Attributes, ISO/IEC 9216, Fuzzy Multi Criteria Approach, Software Quality Evaluation, Fuzzy Membership Function.

I. INTRODUCTION

Software engineering is the application of a systematic, disciplined, and quantifiable approach to the development, operation and maintenance of software [1]. Software engineering mainly aims to produce software of good quality, deliver on time and on budget. It is a continuously growing field. Due to the evolution in the applications of software engineering, software quality becomes an important field for quantifying the performance of software attributes.

Software quality can be measured by using various parameters or metrics. These parameters or metrics are then combined to evaluate the software system. Various researchers have worked so far in developing models that defines parameters to judge software quality like ISO/IEC 9126 model [2], Boehm's model [3], Dromey's model [4], McCall's model and the FURPS model [5]. These models are proposed for a general software application. Out of the above mentioned model, ISO/IEC 9126 is the most important model, which covers almost all the attributes defined in other models. It is a highly accepted and recognized model in industry, research community and various further organizations.

ISO/IEC 9126 is a very successful model in quantifying the quality of software systems. But we have made an attempt to further increase the software quality by introducing some more quality measuring parameters in the actual model only i.e. ISO/IEC 9126. These newly added parameters will help to increase the software quality results from what we are getting into the actual model.

Quality attributes are very unpredictable in nature; to evaluate them properly is not an easy task. To deal with these qualities, attributes or to evaluate them we need a technique which can easily evaluate the values of these attributes. One such technique which we will use is a fuzzy multi criteria approach.

Based on the evaluation of software systems in actual model and the proposed model, a comparison is made to clearly show how the introduction of new attributes makes the software quality appraisal high.

II. LITERATURE SURVEY

Software quality is recently the most important field of software engineering. Many researchers have made various attempts to measure the software quality criteria [6-8]. Sharma et al. [8] had considered the Component Based Software Development Model to evaluate the software quality criteria. Srivastava P. R. et al. has also considered ISO/IEC 9126 model as a base model to quantify the software quality parameters by taking the weighted average of all those factors in developer's, user's and project manager's perspective to get the net software quality [6]. Slaughter S. A. et al. has also made an attempt to evaluate the cost of software quality [9]. Agarwal M. and Chari k. had considered quality, effort and cycle time to evaluate software quality [10].

Kanellopoulos Y. et al. evaluated the quality of code with help of Analytical Hierarchy process model [11]. They evaluate it on the basis of some characteristics which include functionality, efficiency, maintainability and portability.

Previously, Lin L. et al. presented a new evaluation method to calculate the net software quality by using the fuzzy set theory based on the ISO 9126 Sample Quality

Model with a single evaluator [12]. Various prioritizations and synthesis have been done to arrive at final software quality in terms of triangular fuzzy number, which can be defuzzified to get the original software quality.

Present paper, presents a method to evaluate the net software quality using ISO/IEC 9126 as a model with some new characteristics added to it. Fuzzy multi-criteria approach will be used to evaluate the net software quality. This approach takes several real time values for metrics as input and gives quantified software quality as output.

III. SOFTWARE QUALITY MODEL

A. ISO/IEC 9126

The fundamental objective of the ISO/IEC 9126 standard is to address some of the well known human biases that can adversely affect the delivery and the perception of a software development project [17]. These biases include changing priorities after the start of a project or not having any clear definitions of "success." By clarifying, then agreeing to the project priorities and subsequently converting abstract priorities (compliance) to measurable values (output data can be validated against the schema X with zero intervention), ISO/IEC 9126 tries to develop a universal accepting of the project's objectives and goals.

The ISO/IEC 9126 standard makes a distinction between internal quality and external quality, and introduce the so-called quality in use [2]. These model categories software qualities attributes into characteristics. The ISO/IEC 9126 classifies software quality in a structured set of characteristics and sub-characteristics as follows [2]:

1. **Functionality:** A set of attributes that bear on the existence of a set of functions and their specific properties. The functions are those that assure avowed or implied needs. It contains: Suitability, Accuracy, Interoperability, Security and Functionality Compliance.
2. **Efficiency:** A set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under

stated conditions. It contains: Time Behavior, Resource Behavior and Efficiency Compliance.

3. **Maintainability:** A set of attributes that bear on the effort needed to make specified modifications. It contains: Analyzability, Changeability, Testability, Stability and Maintainability Compliance.
4. **Portability:** A set of attributes that bear on the ability of software to be transferred from one environment to another. It contains: Replace-ability, Adaptability, Install-ability, Co-existence and Portability Compliance.
5. **Usability:** A set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. It contains: Understandability, Learn-ability, Operability, Attractiveness and Usability Compliance.
6. **Reliability:** A set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time. It contains: Maturity, Recoverability, Fault Tolerance and Reliability Compliance.

B. Proposed Model

New Sub-Characteristics: The following attributes are added to the ISO/IEC 9126 Model to enhance the quality of software systems:

- a. **Flexibility:** The degree to which the user may introduce extensions or modifications to the information system without changing the software itself. This sub-characteristic is added under Functionality characteristic.
- b. **Robustness:** The degree to which the information system proceeds as usual even after an interruption. This sub-characteristic is added under Functionality characteristic.
- c. **Supportability:** Supportability is the ability of the system to provide information helpful for identifying and resolving issues when it fails to work correctly. This sub-characteristic is added under Usability characteristic.

The Table 1 below shows the various sub-characteristics of the proposed attributes along with the above defined new sub-characteristics:

Table 1 Proposed Model

Proposed Model					
Functionality	Efficiency	Maintainability	Portability	Usability	Reliability
Suitability	Time Behaviour	Analyzability	Replace-ability	Understand-ability	Maturity
Accuracy	Resource Behaviour	Changeability	Adaptability	Learn-ability	Recoverability
Interoperability	Efficiency Compliance	Testability	Install-ability	Operability	Fault Tolerance
Security		Maintainability Compliance	Co-Existence	Attractiveness	Reliability Compliance
Functionality Compliance			Portability Compliance	Usability Compliance	
Flexibility				Supportability	
Robustness					

IV. FUZZY MULTI-CRITERIA APPROACH

Fuzzy logic is a very powerful technique that can be used to get solutions of those problems who are uncertain, imprecise, vague and ambiguous in nature [15].

In this paper we are using a triangular fuzzy approach. This approach helps to evaluate software quality by evaluating their sub-attributes, then attribute, then characteristics and finally the software quality (as shown in the Figure 1).

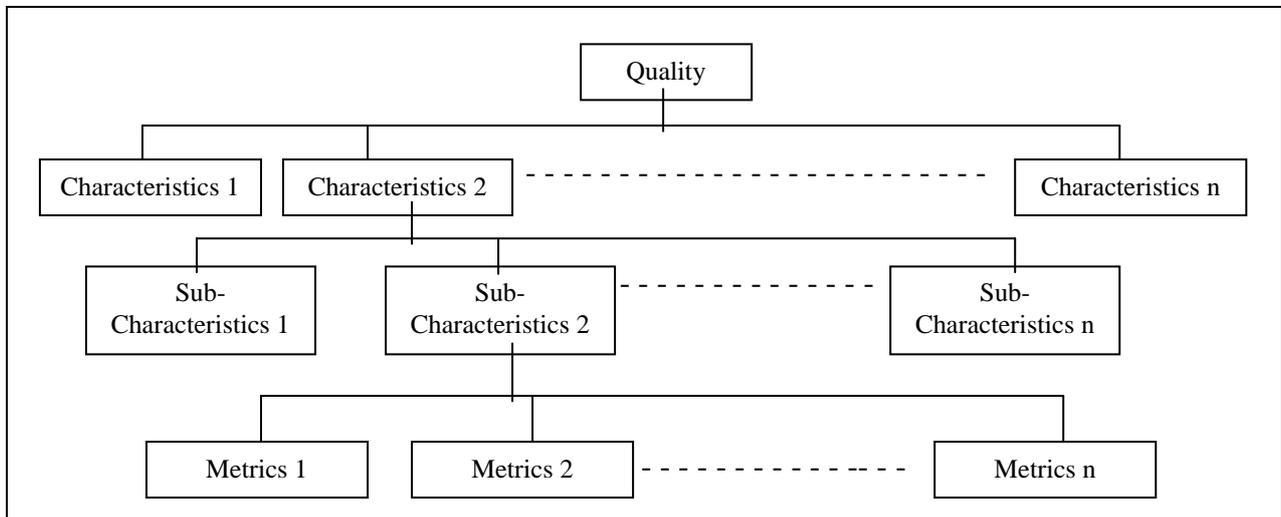


Fig. 1 Hierarchy of Software Quality Parameters

A. Procedure

We will follow the following procedure to evaluate the net software quality-

1. Assign fuzzy ratings (r_i) to all the metrics in the hierarchy structure.
2. Assign fuzzy weights (w_i) to all the nodes (characteristics, sub-characteristics and metrics) present in the hierarchy structure.
3.
 - a. Firstly, take the fuzzy weighted average of the metrics (level 3) to evaluate the rating of the sub-characteristic.
 - b. Secondly, fuzzy weighted average of the sub-characteristics (level 2) is taken to evaluate the rating of characteristics.
 - c. Thirdly, take the fuzzy weighted average of the characteristics to evaluate the net fuzzy rating.
4. Now with the help of net fuzzy rating, a triangular fuzzy membership function is build [14].
5. Based on the triangular fuzzy membership function a crisp value is calculated from the defuzzification process [37] as shown below:

$$z^* = \frac{\int \mu(z) z dz}{\int \mu(z) dz}$$

Where z^* is the crisp value.

V. CASE STUDY

Here for the evaluation & quality appraisal of the proposed quality model with respect to actual quality

model, we will choose a sample case study of Ms Word 2003. A group of 5 users was made to fill a questionnaire for the fuzzification criteria for all the parameters given in the hierarchical structure. These questionnaires will give the fuzzy set values by taking the real time values as Very High (VH), High (H), Medium (M), Low (L) and Very Low (L). Consider the following triangular fuzzy sets for fuzzy ratings & fuzzy weights in Figure 2 and 3 [16].

Importance of Criteria	Fuzzy Ratings
Very Low	(0.0,0.1,0.3)
Low	(0.1,0.3,0.5)
Medium	(0.3,0.5,0.7)
High	(0.5,0.7,0.9)
Very High	(0.7,0.9,1.0)

Fig. 2 Fuzzy Ratings

Importance of Criteria	Fuzzy Weightings
Very Low	(0.0,0.0,0.25)
Low	(0.0,0.25,0.5)
Medium	(0.25,0.5,0.75)
High	(0.5,0.75,1.0)
Very High	(0.75,1.0,1.0)

Fig. 3 Fuzzy Weights

Table 2. Metrics weights calculation of Functionality Sub-characteristics

Metrics (Functionality)	Metrics (Ratings)	U1	U2	U3	U4	U5	Average Ratings
Suitability	1- (no. of operations not suitable/total no. of operations)	H	M	M	H	H	(0.48, 0.62, 0.82)
Accuracy	Degree of importance for the number of operations meeting required accuracy	H	H	H	H	H	(0.5, 0.7, 0.9)
	Degree of importance for precision	M	M	H	H	M	(0.38, 0.58, 0.78)
Interoperability	Degree of importance for databases	H	M	M	M	H	(0.38, 0.58, 0.78)
	Degree of importance for multimedia & graphics	H	M	M	L	H	(0.4, 0.54, 0.74)
	Degree of importance for file system support	H	H	H	H	H	(0.5, 0.7, 0.9)
	Degree of importance for Internet support	VH	H	H	H	H	(0.54, 0.74, 0.92)
Security	Degree of importance for access controllability	H	H	VH	H	H	(0.5, 0.7, 0.9)
	Degree of importance for software that enables restricted user access	M	H	M	M	M	(0.34, 0.54, 0.74)
Functionality Compliance	Whether software has adhered to functionality compliance standards or not	H	M	M	M	M	(0.34, 0.54, 0.74)
Flexibility	Degree of extending functionality	H	M	H	M	H	(0.48, 0.62, 0.82)
	Degree of adoptability of any new feature	M	H	H	H	H	(0.46, 0.66, 0.86)
	Degree of integrity among old & new versions	H	H	M	H	H	(0.46, 0.66, 0.86)
Robustness	recovery file made or not	H	H	H	H	H	(0.5, 0.7, 0.9)
	lost event information provided or not	H	H	H	H	H	(0.5, 0.7, 0.9)

Table 3. Metrics weights calculation of Functionality Characteristics

Metrics (Functionality)	Metrics (Weights)	U1	U2	U3	U4	U5	Average Weights
Suitability	1- (no. of operations not suitable/total no. of operations)	H	VH	VH	H	H	(0.6, 0.85, 1)
Accuracy	Degree of importance for the number of operations meeting required accuracy	H	H	H	VH	H	(0.55, 0.8, 1)
	Degree of importance for precision	H	H	H	H	H	(0.5, 0.75, 1.0)
Interoperability	Degree of importance for databases	H	VH	H	H	VH	(0.6, 0.85, 1)
	Degree of importance for multimedia & graphics	H	M	M	M	H	(0.35, 0.6, 0.85)
	Degree of importance for file system support	H	H	H	H	H	(0.5, 0.75, 1.0)
	Degree of importance for Internet support	VH	H	VH	H	VH	(0.65, 0.9, 1)
Security	Degree of importance for access controllability	H	H	M	H	H	(0.48, 0.52, 1)
	Degree of importance for software that enables restricted user access	VH	H	H	H	VH	(0.6, 0.85, 1)
Functionality Compliance	Whether software has adhered to functionality compliance standards or not	H	H	H	H	H	(0.5, 0.75, 1.0)
Flexibility	Degree of extending functionality	H	H	H	H	H	(0.5, 0.75, 1.0)
	Degree of adoptability of any new feature	VH	H	H	H	H	(0.55, 0.8, 1)
	Degree of integrity among old & new versions	H	H	VH	H	H	(0.55, 0.8, 1)
Robustness	recovery file made or not	H	VH	H	H	H	(0.55, 0.8, 1)
	lost event information provided or not	H	VH	VH	H	H	(0.6, 0.85, 1)

Similarly the average ratings and weights of other metrics, sub-characteristics and characteristics are calculated on the basis of real time values gathered from the questionnaire. After evaluating the fuzzy ratings of the characteristics the final net software quality given in terms of triangular fuzzy membership function is:

$$\Gamma_{\text{net_quality_proposed}} = (0.13, 0.50, 0.92)$$

The above is evaluated the net quality is for the proposed model and when we calculate the net quality of the actual ISO/IEC 9126 model is given below:

$$\Gamma_{\text{net_quality_actual}} = (0.1, 0.39, 0.9)$$

The fuzzy membership function for both actual and proposed model is shown in figure 4 and 5:

$$\text{Equation of Line 1: } \mu = 3z - 0.34$$

$$\text{Equation of Line 2: } \mu = 1.76 - 1.96z$$

$$z^* = \left(\int (3z - 0.34)z \, dz \, (z = 0.1 \text{ to } 0.39) + \int (1.76 - 1.96z)z \, dz \, (z = 0.39 \text{ to } 0.9) \right) / \left(\int (3z - 0.34) \, dz \, (z = 0.1 \text{ to } 0.39) + \int (1.76 - 1.96z) \, dz \, (z = 0.39 \text{ to } 0.9) \right)$$

$$z^*_{\text{actual_model}} = 0.475$$

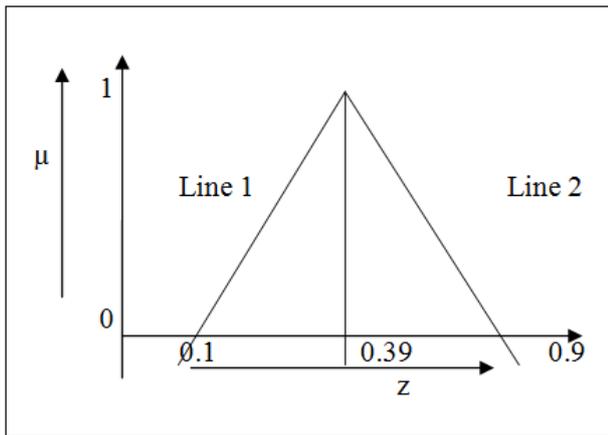


Fig. 4 Fuzzy Membership Function for Actual Model

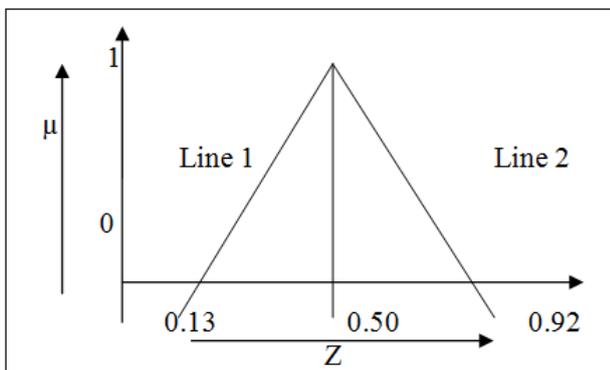


Fig. 5 Fuzzy Membership Function for Proposed Model

$$\text{Equation of Line 1: } \mu = 2.7z - 0.35$$

$$\text{Equation of Line 2: } \mu = 2.19 - 2.38z$$

$$z^* = \left(\int (2.7z - 0.35)z \, dz \, (z = 0.13 \text{ to } 0.50) + \int (2.19 - 2.38z)z \, dz \, (z = 0.50 \text{ to } 0.92) \right) / \left(\int (2.7z - 0.35) \, dz \, (z = 0.13 \text{ to } 0.50) + \int (2.19 - 2.38z) \, dz \, (z = 0.50 \text{ to } 0.92) \right)$$

$$z^*_{\text{proposed_model}} = 0.512$$

VI. RESULTS

As seen from the crisp value of both the model, we analyze that by adding proposed characteristics to the actual model we can make the software quality better. Based on the some real value data, the characteristics show the increase performance whenever new quality attributes are added to the original set of attributes.

VII. CONCLUSION

In the past few years, various quality models were proposed which are used to evaluate the quality of software systems. In this paper, we have used ISO/IEC 9126 as the base model and added some more characteristics to it and then evaluate the quality of software by using the proposed model. Using the algorithm given in this paper, we can evaluate similar software's performance with the proposed model on larger scale. Crisp value will give the idea about the

quality of software for a given set of users in a certain environment.

REFERENCES

- [1] P. Bourque and R. Dupuis, Guide to the Software Engineering Body of Knowledge, 2004 Edition, Vol. 1, IEEE Press Piscataway, NJ, USA, 2004, pp. 1-1.
- [2] ISO/IEC 9126-1:2001, "Software Engineering-Product Quality Part 1: Quality Model", Int'l Organization for Standardization, 2001.
- [3] B. W. Boehm, J. R. Brown and M. L. Lipow, "Quantitative Evaluation of Software Quality," Proceedings of the 2nd International Conference on Software Engineering, San Francisco, CA, USA, October, 1976, pp. 592-605.
- [4] J. A. McCall, P. K. Richards, and G. F. Walters, Factors in Software Quality, 1977, Vol. I, II, 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.
- [5] R. G. Dromey, "A model for software product quality," IEEE Transactions on Software Engineering, Vol. 21, No. 2, February, 1995, pp. 146-162.
- [6] P. R. Srivastava and K. Kumar, "An Approach towards Software Quality Assessment," Communications in Computer and Information System Series (CCIS Springer Verlag), Vol. 31, No. 6, 2009, pp. 345-346.
- [7] P. R. Srivastava, A. P. Singh, K. V. Vageesh, "Assessment of Software Quality: A Fuzzy Multi-Criteria Approach," Evolution of Computational Optimization Algorithms in Software Engineering: Applications and Techniques, IGI Global USA, 2010, chapter - 11, pp. 200-219.
- [8] A. Sharma, R. Kumar and P.S. Grover, "Estimation of Quality for Software Components- an Empirical Approach," ACM SIGSOFT Software Engineering Notes, Vol. 3, No. 5, November, 2008, pp. 1-10.
- [9] S. A. Slaughter, D. E. Harter, & M. S. Krishnan, "Evaluating the Cost of Software Quality," Communications of the ACM, Vol. 41, No. 8, August, 1998, pp. 67-73.
- [10] M. Agarwal, & K. Chari, "Software Effort, Quality, and Cycle Time: A Study of CMM Level 5 Projects," IEEE Transactions on Software Engineering, Vol. 33, No. 3, March, 2007, pp. 145-156.
- [11] Y. Kanellopoulos, P. Antonellis, D. Antoniou, C. Makris, E. Theodoridis, C. Tjortjis and N. Tsirakis, "Code Quality Evaluation Methodology Using The ISO/IEC 9126 Standard," International Journal of Software Engineering & Applications (IJSEA), Vol. 1, No. 3, July, 2010, pp. 17-36.
- [12] L. Lin and H. M. Lec, "A Fuzzy Software Quality Assessment Model to Evaluate User Satisfaction," Proceedings of the Second International Conference on Innovative Computing, Information and Control, Washington DC, USA, September, 2007, pp. 760-764.
- [13] T. J. Ross, Fuzzy Logic with Engineering Applications, 2nd Ed, Wiley India Pvt. Ltd, New Delhi, India, 2004.
- [14] A. P. Singh and A. K. Vidyarthi, "Optimal allocation of landfill disposal site: A fuzzy multi criteria approach," Iranian Journal of Environmental Health Science & Engineering, Vol. 5, No. 1, 2008, pp. 25-34.
- [15] G. Buyukozkan, C. Kahramanand D. Ruan, "A fuzzy multi-criteria decision approach for software development strategy selection," International Journal of General Systems, Vol. 33, No. (2-3), 2004, pp. 259-280.

- [16] J. S. Challa, A. Paul, Y. Dada, V. Nerella, P. R. Srivastava and A. P. Singh, "Integrated Software Quality Evaluation: A Fuzzy Multi-Criteria Approach," Journal of Information Processing Systems, Vol. 7, No. 3, Sep 2011, pp. 473-518.

Mining. He has published more than 55 research papers in indexed National & International Journals including ACM and in Proceedings of the reputed International/ National Conferences (including IEEE Explore). These publications have good citation records. He has authored 2 books also. In addition, he has also served as a Technical Program Committee Member of several reputed conferences.

Author's Profile



Dr. Sanjay Kumar Dubey is Assistant Professor in Department of Computer Science and Engineering in Amity School of Engineering and Technology, Amity University Uttar Pradesh, India. He has 13 year of teaching experience. He is member of IET and ACM digital library. He has a rich academics & research experience in various areas of Computer Science.

His research areas include Object Oriented Software Engineering, Soft Computing, HCI, Cloud Computing and Data



Disha Sharma has done her B. Tech (IT) from UPTU, Lucknow, (U.P.), India. She is pursuing her M. Tech in CSE from Amity University. Her research interest is Software Engineering and Artificial Intelligence. She has published two papers in International Journals.

How to cite this paper: Sanjay Kumar Dubey, Disha Sharma, "Software Quality Appraisal Using Multi-Criteria Decision Approach", IJIEEB, vol.7, no.2, pp.8-13, 2015. DOI: 10.5815/ijieeb.2015.02.02