"Vasile Alecsandri" University of Bacău
Faculty of Sciences
Scientific Studies and Research
Series Mathematics and Informatics
Vol. 23 (2013), No. 1, 49 - 53

SOFTWARE QUALITY EVALUATION USING FUZZY THEORY

MARA HAJDU-MACELARU

Abstract. The software quality evaluation is still not a fully resolved problem in Software Engineering. It is known that software quality depends upon several quality attributes such as reliability, modifiability, understandability, testability, usability, portability, and so on. Several authors and international standards have tried to reach an agreement on their definition over the years and provide a standard and accurate one. Due to the complexity of software systems, it is often difficult to evaluate the overall quality of their underlying software components. A way for evaluation is to use software metrics as quantitative predictors when evaluating the quality attributes. There are several metrics based on statistical methods, similar to those used for product inspection and quality assurance in manufacturing, that have been investigated during time by Software Engineering. Some authors have started to use Fuzzy Theory in software assessment and evaluation. And it seems that its usage improves the evaluation of the software quality.

1. INTRODUCTION

Several quality models have been proposed during time, most of them containing same common attributes. Taking the ISO 9126 quality model we have the following characteristics/subcharacteristics that define the quality [1]:

1. Functionality characteristic:

Suitability- This is the essential Functionality characteristic and refers to the appropriateness (to specification) of the functions of the software. 'Can the software perform the tasks required?';

Keywords and phrases: Software Engineering, Fuzzy Theory.
(2010) Mathematics Subject Classification: 03E72, 68N99.

Accurateness - 'Is the result as expected?';
Interoperability - 'Can the system interact with another system?';
Compliance - 'Is the system compliant with standards?'
Security - 'Does the system prevent unauthorized access?']
2. Reliability characteristic:
Maturity - This subcharacteristic concerns frequency of failure of the software. Fault tolerance -The ability of software to withstand (and recover) from component, or environmental, failure.
Recoverability - Ability to bring back a failed system to full operation, including data and network connections.]
3. Usability characteristic:
Understandability - determines the ease of which the systems functions can be understood, relates to user mental models in Human Computer Interaction methods.
Learnability - Learning effort for different users, i.e. novice, expert, casual etc.
Operability - Ability of the software to be easily operated by a given user in a given environment.
4. Efficiency characteristic:
Time behavior - Characterizes response times for a given thru put, i.e. transaction rate.
Resource behavior - Characterizes resources used, i.e. memory, cpu, disk and network usage.
5. Maintainability characteristic:
Analyzability - characterizes the ability to identify the root cause of a failure within the software.
Changeability - Characterizes the amount of effort to change a system.
Stability - Characterizes the sensitivity to change of a given system that is the negative impact that may be caused by system changes.
Testability - Characterizes the effort needed to verify (test) a system change.
6. **Portability characteristic**:
Adaptability - Characterizes the ability of the system to change to new specifications or operating environments.
Installability - Characterizes the effort required to install the software.
Conformance - Similar to compliance for functionality, but this characteristic relates to portability.
Replaceability - characterizes the plug and play aspect of software components, that is how easy is it to exchange a given software component within a specified environment.

All this characteristics and sub-characteristics will be taken in consideration while evaluating the software quality. Because the vagueness that occur while evaluating this characteristics we will introduce the fuzzy approach to help in evaluating the quality.

Fuzzy Set Theory was formalized by Professor Lofti Zadeh at the University of California in 1965. What Zadeh proposed is very much a paradigm shift that first gained acceptance in the Far East and its successful application has ensured its adoption around the world.

A paradigm is a set of rules and regulations which defines boundaries and tells us what to do to be successful in solving problems within these boundaries. For example the use of transistors instead of vacuum tubes is a paradigm shift - likewise the development of Fuzzy Set Theory from conventional bivalent set theory is a paradigm shift.

2. FUZZY THEORY IN EVALUATION OF SOFTWARE QUALITY

Fuzzy set theory is an extension of the classical theory where elements have varying degrees of membership. Sometimes, the logic based on True and False is not adequate when it comes to describe the human reasoning. The fuzzy logic uses the whole interval between 0 (false) and 1(true) to describe the human reasoning.[2]

If we look at the characteristics and their subcharacteristics described in the ISO 9126 model, we will noticed that some sub subcharacteristics should count more on the users evaluation output and some on the experts evaluation output. If we take a general definition of the quality that says the quality stands in the user satisfaction, we tend to agree partially with it. Some of the product quality subcharacteristics should be evaluated by the users, and the degree of importance for their evaluation should be higher than the one done by experts.

Other subcharacteristics related to the quality can be evaluated only by experts. Maintainability for instance, is a characteristic that can be only evaluated by experts. Usability subcharacteristics should count more on the users evaluation output. We will use triangular fuzzy sets to make the difference between the groups of evaluators based on their evaluation impact over the subcharacteristic evaluated:

User importance criteria:	**Fuzzy degree:**
Very Low	(0.0,0.0,0.3)
Low	(0.0,0.3,0.6)
Medium	(0.3,0.6,0.9)
High	(0.6,0.9,1.0)
Very High	(0.9,1.0,1.0)

Table 2.1: Fuzzy criteria for user evaluation importance:

Same as above we can define the expert importance evaluation degree:

Expert importance criteria:	**Fuzzy degree:**
Very Low	(0.0,0.0,0.3)
Low	(0.0,0.3,0.6)
Medium	(0.3,0.6,0.9)
High	(0.6,0.9,1.0)
Very High	(0.9,1.0,1.0)

Table 2.2: Fuzzy criteria for expert evaluation importance:

Now that we pointed that for evaluating the quality we need to have the evaluation output coming from users as well as from the experts, and we have define their evaluation importance degree, looking at the characteristics and their subcharacteristics we can noticed that some subcharacteristics evaluation weight much more than the other while evaluating that specific characteristic.

For this purpose, we will also use the triangular fuzzy sets:

Subcharacteristic importance criteria:	**Fuzzy degree:**
Very Low	(0.0,0.0,0.2)
Low	(0.0,0.2,0.4)
Medium	(0.2,0.4,0.8)
High	(0.4,0.8,1.0)
Very High	(0.8,1.0,1.0)

Table 2.3: Fuzzy triangular number for user subcharacteristic importance in evaluating the characteristic:

There are several metrics that can be used to evaluate a subcharacteristic. For each metric used, we will define a triangular set to define the importance weight and rating.

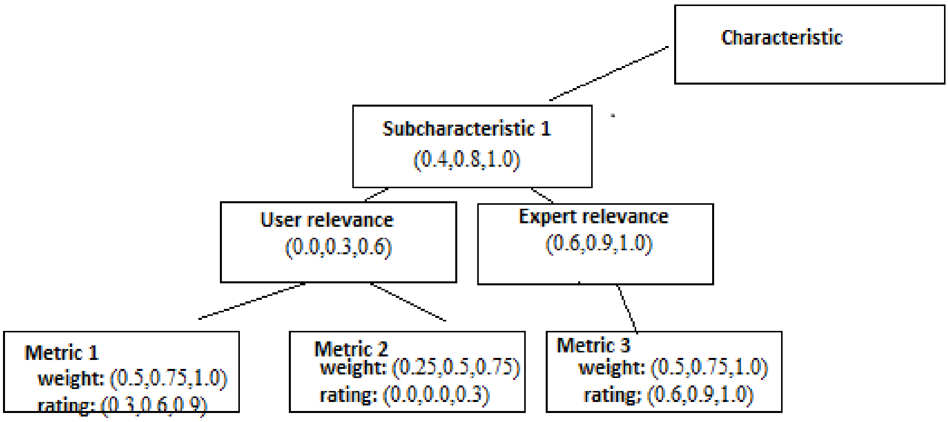
User importance criteria:	**Fuzzy degree:**
Very Low	(0.0,0.0,0.25)
Low	(0.0,0.25,0.5)
Medium	(0.25,0.5,0.75)
High	(0.5,0.75,1.0)
Very High	(0.75,1.0,1.0)

Table 2.4: Fuzzy triangular number for weight of a metric:

Expert importance criteria:	Fuzzy degree:
Very Low	(0.0,0.0,0.3)
Low	(0.0,0.3,0.6)
Medium	(0.3,0.6,0.9)
High	(0.6,0.9,1.0)
Very High	(0.9,1.0,1.0)

Table 2.5: Fuzzy triangular number for rating a metric:

Example:



In doing the final evaluation we will multiply the corresponding triangle sets in order to evaluate.

REFERENCES

[1] I. Padayachee, P. Kotze, A. van Der Merwe, **ISO 9126 external systems quality characteristics, subcharacteristics and domain specific criteria for evaluating e-Learning systems**, The Southern African Computer Lecturers' Association, University of Pretoria, South Africa. (2010).

[2] H. – J. Zimmermann, **Fuzzy set theory**, Springer (2001).

Technical University of Cluj-Napoca, North University Centre of Baia Mare
Faculty of Sciences
Department of Mathematics and Informatics
76 Victoriei, Baia-Mare, 430122, ROMANIA
e-mail: mara.macelaru@ubm.ro