Analysis of Quality Evaluation Based on ISO/IEC SQuaRE Series Standards and Its Considerations

Yi Zhao[1,2], Jiayu Gong[1,3,*], Yun Hu[1], Zhenyu Liu[1], Lizhi Cai[1,*]
[1]Shanghai Key Laboratory of Computer Software Testing &Evaluating
Shanghai Development Center of Computer Software Technology
[2]Shanghai Ruanzhong Information Technology Co.,Ltd
[3]Shanghai Industrial Technology Institute
Shanghai, China
{zy,gjy,huy, lzy, clz}@ssc.stn.sh.cn

***Abstract*—System and software quality evaluation is an important method of quality assurance, and its standardization provides quality requirements and evaluation supported by quality measurement. ISO/IEC organizations have also published a series of standards on system and software quality requirements and evaluation, which is significant for assuring the quality of system and software. This paper introduces quality evaluation based on SQuaRE series standards, and analyzes the relationships among quality model, measurement, requirements, evaluation. Besides, considerations and suggestions are given when applying the measurement function to quality evaluation. Finally, comparison with different methods of quantifying quality characteristics and sub-characteristics with the measured values is analyzed.**

Keywords—SQuaRE Series Standards; System and Software Quality Model; Quality Measurement; Quality Evaluation; Quantifying evaluation

I. INTRODUCTION

As the development of information technology, software is much more popular in our daily life, for example mobile payments, intelligence transportation. Software quality is a vital factor that related to business success and human safety, which describes the degree of conformance to explicit or implicit requirements and expectations, and it has been attracted more and more attention. There are two approaches that can be followed to ensure software quality. One is focused on a direct specification and evaluation of the quality of software product, while the other is to assure the process quality when the product is under developed. Here we take the first approach to ensure the software quality, and quality evaluation aims at systematic examination of the extent to which an entity is capable of fulfilling specified requirements.

In recent years, ISO/IEC organizations publish a series of standards about System and software Quality Requirement and Evaluation ("SQuaRE", also "25000 series standards"). This series of standards specifies quality model, measurement, requirement, and evaluation process applied to system and software quality assessment, and it can help to develop and capture systems and software products using quality requirements and evaluations [1].

SQuaRE includes International Standards on quality model and measurement, as well as on quality requirements and evaluation, consisting of Quality Management Division (ISO/IEC 2500n), Quality Model Division(ISO/IEC 2501n), Quality Measurement Division(ISO/IEC 2502n), Quality Requirements Division(ISO/IEC 2503n), Quality Evaluation Division(ISO/IEC 2504n)and Extension Division (ISO/IEC 25050 to 25099) five divisions respectively.

ISO/IEC 2500n defines common terms, definitions and models used in SQuaRE series standards, along with requirements and guidance for management of system and software product quality [1]. ISO/IEC 2501n presents quality models and guidance on model applications [2, 3]. ISO/IEC 2502n describes how to measurement a system/software product quality, in which measurement reference model, mathematical definitions of quality measures and their application guidance are given to in detail[1,4,5,6,7].ISO/IEC 2503n specifies quality requirements which can be applied to evaluation process as inputs on the basis of quality models and quality measures[8]. ISO/IEC 2504n defines evaluation process which is a synthetic application of quality model, measurement and requirements along with its guidance for independent evaluators, acquirers or developers [1, 9, 10]. Besides, Extension Division addresses its application in specific domains and as complements to SQuaRE series standards.

This paper presents how to use the SQuaRE series of standards for system and software quality evaluation, also provides analysis of different quantifying techniques. Section II depicts the relationships among the series standards along with how to evaluate the system and software quality by SQuaRE series of standards. Section III provides the considerations of measurement functions applied to quality evaluation and the analysis of different quantization methods of evaluation results.

978-1-5090-5507-4/17/$31.00 ©2017 IEEE
ICIS 2017, May 24-26, 2017, Wuhan, China

II. Quality Evaluation Scheme Based On SQuaRE Series Standards

Figure1 in the below depicts the relationships among SQuaRE series standards.

ISO/IEC 25010, 25012, 25020 and 2502n as basic resources provide quality model and measurement for quality requirements and evaluation, in which ISO/IEC 2502n provides quality measurement based on the quality model defined in ISO/IEC 25010. ISO/IEC 25030 specifies quality requirements based on the stakeholder needs, which can map the stakeholder needs to system/software requirements by the requirement definition and analysis phases [1, 8]. Stakeholder needs may be stated, implied and unaware. However, quality requirements and evaluation might be influenced by outer constraints such as the cost, environment, tools and methodology [9]. After quality model, measurement and requirements have been determined, system/software quality evaluation can be performed according to ISO/IEC 25040 and 25041.A system/software quality evaluation report can be completed after evaluation, which documents evaluation activities and results. It should be noticed that a generic quality evaluation process is provided in ISO/IEC 25040, while the process defined in ISO/IEC 25041 is suitable for developers, acquirers, independent evaluators. As is shown in the Figure 1, it is inseparable for each SQuaRE standard to perform system/software quality evaluation.

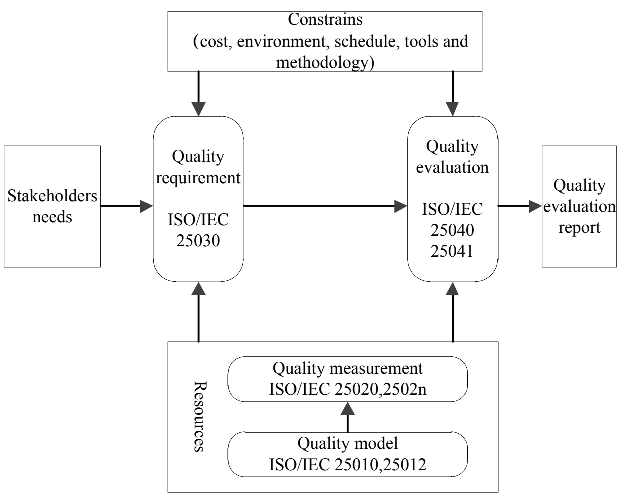

Fig. 1. Relationships among quality model, measurement,requirement and evaluation[11]

A. Quality Model

The latest quality model defined in ISO/IEC 25010:2011 is evolved from the model in ISO/IEC 9126, in which the new model consists of two sub-models from the view of the system, that are quality in use model with five characteristics, and product quality model with eight characteristics. Besides, data quality given in ISO/IEC 25012 describes data quality characteristics in view of inherent and system-dependent aspects. Each characteristic is broad and therefore it is divided into a set of sub-characteristics or attributes.

Quality in use model describes the effectiveness, efficiency, satisfaction, freedom from risk and context coverage five characteristics in the aspects of uses, and each characteristic can be used for different stakeholders activities, for example maintenance activities for developers.

The product quality model applied to a software product or a computer system consists eight quality characteristics: functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability and portability.

Quality in use model and product quality model are useful for specifying requirements, establishing measures, and quality evaluations. It should be noticed that characteristics and sub-characteristics in the quality model can be tailored based on the high-level goals and objectives for the project, with the consideration of unpractical actives of specifying all the characteristics and sub-characteristics. Therefore it is suggested focusing on relative important quality characteristics and sub-characteristics according to the project goals.

Data quality model in ISO/IEC 25012 is complementary to product quality model and is categorized into inherent and system-dependent data quality with 15charcteristics. Inherent data quality can be applied to metadata, data domain values and possible restrictions [3]. However, system-dependent data quality depends on the technological domain in which data are used, and it is it related with computer systems' components such as: hardware devices, computer system software and other software [3].

B. Quality Measurement

Quality measurement is defined in ISO/IEC 2502n series standards with quality measurement reference model (SPQM-RM), quantifying quality measurement by mathematic functions and its guidelines based on the models specified in ISO/IEC 25010.

The SPQM-RM model provided in ISO/IEC 25020 gives the description of the relationships among a quality model, its associated quality characteristics (and sub-characteristics), software product attributes with the corresponding software quality measures, measurement functions, quality measure elements, and measurement methods [4]. Quality measures element specified in the ISO/IEC 25021 provides the format specifications and examples for the measurement elements used for quality measurements construction. Measurement functions in ISO/IEC 25022, 25023, 25024 describe how the quality measure elements are combined to produce the quality measure. For example, clause 8.2.1 in the ISO/IEC 25023 defines measurement function of function completeness measures with functional coverage:

$$X=A/B \quad (1)$$

Where, X describes the results of measurement. "A" stands for the number of functions missing that is detected when the system/software doesn't reach the required function, and "B" refers to the number of functions specified which can be obtained in requirement specification, design specification or user manual [7].The formulas in ISO/IEC 25022, 25023, 25024 are liner functions like the formula (1), however, this is not suitable for all the measurement situations. For example when measurement results are close to accept range by normalization to the interval [0, 1], it is difficult to distinguish the values resulting in the worse rating, and more details will be discussed in Section III.

C. *Quality Requirement*

Quality requirement in ISO/IEC 25030 represents how to specify system/ software quality requirements on the basis of quality model defined in ISO/IEC 25010.Quality requirements are eventually obtained by selection of stakeholders' needs and transformation into system/software quality requirements through definition and analysis phases. The requirements in ISO/IEC 25030 focus on system requirements generally including software requirements and other software requirements (such as hardware, data, business requirements) [8]. Also software requirements may consist of software product and software development requirements.

D. *Quality Evaluation*

Quality evaluation is a sequence of activities described the extent to which the system/software meets the criteria [9]. ISO/IEC 25040 provides requirements, recommendations and guidelines for system/software product quality evaluation based on the specified quality requirements with 5 sub-procedures as shown in the Figure2. It is a generic process that can be used for evaluation of pre-developed software or custom software under-developed by the quality model defined in ISO/IEC 25010. The followings are the details of each activity in the Figure 2.

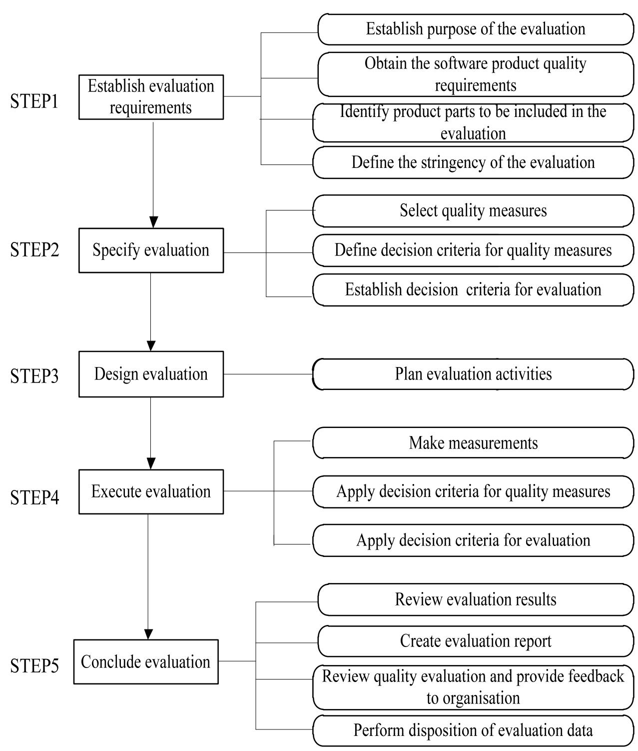

Fig. 2. Quality evaluation process in ISO/IEC 25040

1) *Establish evaluation requirements*

The activity consists four tasks, in which the quality evaluation needs, requirements specification and system/ software product can be as the inputs, and the outputs are the a set of specifications related to quality evaluation purposes, quality evaluation requirements and quality evaluation plan in high level correspondingly.

The first task in this activity is to establish the evaluation purpose for the future activities, and the evaluated system/software could be a final or an intermediate product; for example the purpose might be estimating the quality of the final product. Specifying the quality requirement could be done using a quality model combined with the ISO/IEC 25030 after the determining the evaluation purpose. Next is to identify and document the parts to be evaluated with the considerations of the stage of life cycle and evaluation purpose. When the first three tasks are finished, evaluation stringency could be defined , which are attached to a set of characteristics and sub-characteristics that are given by the requester [9]. The stringency defines the depth of the quality evaluation reflecting the evaluation confidence, and the techniques depended on the expected evaluation levels. The stringency definition could reference to ISO/IEC 15026, for example the stringency is leveled with A, B, C and D, in which A is the highest level with the most stringent evaluation techniques and D is the lowest level [9].

2) *Specify evaluation*

The purpose of this activity with three task is to specify the evaluation modules and the decision criteria of quality measure.The outputs of the previous activity (Establish the evaluation requirems) could be the inputs of this activity;and a set of specifications including selected quality measures,decision criteria for software product quality , the revised high level evaluation plan could be as the outputs in this activity. The quality measures selected should cover all the evaluation requirements ,and also system/software quality characteristics (and/or sub-characteristics) should be measured rigorously in accordance with the standards in ISO/IEC 2502n division.Decision criteria for quality measure and evaluation could be made , which describes the level of confidence in a given result with numerical thresholds. Also it is set in the consideration of quality requirements. Decision criteria for evaluation is used for further summarization as the support of managerial decision[9].

3) *Design evaluation*

This activity is to schedule the evaluation activities with the considerations of budget, methods,tools, adopted standards, personnel and etc[9].The high level evaluation plan defined in the early stage should be revised and adjusted as the evaluation activities evolve and the more detail information provided.

4) *Execute evaluation*

The evaluation execution is the application of the first three activities, which is to produce the results of software product quality measures and evaluation.Firstly, the quality measurement is performed with the selected quality measures based on the evaluation plan .Next is the process of applying the decision criteria for quality measures to the measured values.Finally, the decision criteria for evaluation is applied to characteristics (and/or subcharacteristics) to get the results that descirbe the degree to which the system/software product meets the requirements.

5) Conclude evaluation

This activity is the process of evaluation conclusion, focusing on a joint review of evaluation results,creating evaluation report,review and feedback, and disposition of evaluation data.It is essential to note that in the review and feedback task, not only the evaluation results, but also the validity of evaluation process, measures applied should be reviewed ; and the feedback from the reviewers could be as a good way to improve the evaluation process and techniques[9].Besides, after the evaluation finished, the data and documents used should be appropriate archieved and safely kept for a specified duration or destroyed in a secure way.

III. Considerations of Quality Evaluation

A. Considerations of Measurement functions

Functions in ISO/IEC 25022, 25023, 25024 are defined in liner functions, such as X=A/B or X=1-A/B, but in some cases, the liner functions couldn't deal with the quality measurement. Non-liner functions should be considered in view of values discrimination. Liner function describes the uniform steps changing as the A varies, however the result X by non-liner function changes inhomogeneous. The advantage of non-liner function is to map the independent variable into a recognizable range/level. Here, we don't provide the specific function formula, but only to present a possible way to solve the problem that the measurement results are close to the acceptable range.

Besides, some measurement functions in ISO/IEC 25022, 25023, 25024 may not confine the values in the interval [0, 1]. For example, the time behavior measures belonging to performance efficiency characteristic define the Mean response time attribute (PTb-1-G) that describes the mean time taken by the system to respond to a user task or system task in the following measurement function [7]:

$$X = \sum_{i=1\ to\ n} \left(A_i\right)/n \tag{2}$$

Where, A_i stands for the time taken by the system to respond to a specific user task or system task at *i*-th measurement, and *n* is the number of responses measured. When the measured values are combined to quantify the evaluation, different numeric ranges such as Response time adequacy attribute (PTb-2-G) in the same sub-characteristic (time behavior measures) may result in the wrong evaluation if there is no normalization function. Therefore normalization function could be as the supplement to adjust the numeric range, for example Min-Max Normalization:

$$X = \frac{x - \min}{\max - \min} \tag{3}$$

B. Methods of Quantifying Evaluation with measured values

In this section, several typical methods of quantifying quality characteristics and sub-characteristics with the measured values are discussed in detail. As referred in section II, the selected system/software product quality measures should be applied to the system/software product based on the standards in ISO/IEC 2502n, resulting in the measured values. Also decision criterion for quality measures and evaluation are defined according to ISO/IEC 2504n, therefore how to quantify characteristics and sub-characteristics with the measured values mapping to the decision criteria is the problem. A set of measured results in each sub-characteristic (and/or characteristic) could be obtained according to the measurement functions defined in ISO/IEC 2502n, and what we will do is synthesizing each measured result belonged to a sub-characteristic (or a characteristic) to get the sub-characteristic (or a characteristic) quantified values. Similarly, supposing all the quantified values of specified characteristics are ready, thus final evaluation values could be got. The common quantifying techniques used for quality evaluation generally includes experts score method, AHP (also "Analytic Hierarchy Process") method, fuzzy AHP method and etc.

The experts score method is the simplest available method based on the experts reviewing and grading the characteristics (and/or sub-characteristics) of the system/ software products separately. After experts grading procedure, the scores for each characteristic (and/or sub-characteristic) could be handled for statistics, and a set of weights about characteristics (and/or sub-characteristics) could be obtained. However, this method is strict with the selected experts, especially familiarizing with the system/software product. If experts are not familiar with the software, it will result in the much distortion and inaccuracy. The following formulas could be used to get the weights, in which s_{ij} is the scores for *i*-th characteristic (or sub-characteristic, attribute) by *j*-th expert; *K* is the number of experts, and T_i is the mean score. Therefore the weights could be calculated by the Eq. (5)

$$T_i = \frac{\sum_{j=1}^{K} s_{ij}}{K} \tag{4}$$

$$W_i = \frac{T_i}{\sum T_i} \tag{5}$$

Here we provide an example to get the weights, shown in the Table I.

TABLE I. AN EXAMPLE OF EXPERTS SCORE

Characteristic	Experts										S	T_i	W_i
	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*			
Functional Suitability	7	6	7	6	5	7	6	7	6	7	64	6.4	0.23
Performance efficiency	4	3	4	4	4	5	4	5	4	5	42	4.2	0.15
Compatibility	2	2	2	2	2	3	1	2	2	2	20	2	0.07
Usability	2	2	1	1	2	1	3	1	1	1	15	1.5	0.05
Reliability	6	7	6	6	7	6	7	6	7	6	64	6.4	0.23
Security	5	5	5	6	6	4	5	4	5	4	49	4.9	0.18
Maintainability	1	1	2	2	1	1	1	2	2	2	15	1.5	0.05
Portability	1	2	1	1	1	1	1	1	1	1	11	1.1	0.04
Total	28	28	28	28	28	28	28	28	28	28	280	28	1.00

Similarly, the weights of each sub-characteristic in a characteristic (or each attribute in a sub-characteristic) would be obtained in this method. Thus, the quantized evaluation results could be calculated by weighting the measured values.

AHP method for multi–criteria decision making developed by Thomas L. Saaty in the 1970s and has been widely used in the field of software engineering, which is based on disaggregation of more complex problem on several levels of hierarchy with three basic levels including objective (top level), criteria and alternatives, but it is possible to further disaggregate this structure. For system/software product quality model, levels are classified by characteristics, sub-characteristics and attributes corresponding to first level criteria, second level criteria and alternatives respectively, and the top level is the system/software quality. Once the hierarchy is established, the quantifying works could be starts by comparing its various elements to each other two at a time about their impact on an element above them in the hierarchy. The result of AHP method is a list of relevant significance with numerical weight for each element of the hierarchy. Finally weighted linear combination could be performed to get the quantified evaluation results with the obtained numerical weights. As to AHP method, CR (also "Consistency Ratio ") parameter is important for improving accuracy and reducing the subject factors, which is a comparison between Consistency Index and Random Consistency Index, or in Eq.(6)

$$CR = CI / RI \quad (6)$$

Where, CI is calculated by the Eq.(7)

$$CI = \frac{\lambda_{\max} - n}{n-1} \quad (7)$$

In which, n is the dimension of the matrix, and λ_{max} represents maximal eigenvalue.

Also random consistency index calculated by Saaty is given in Table II.

TABLE II. RANDOM CONSISTENCY INDEX

n	3	4	5	6	7	8	9	10
RI	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

If CR is smaller or equal to 10%, an acceptable consistency could be considered; otherwise, if CR is greater than 10%, the subjective judgment would be revised.

AHP method has the advantage of transforming the quantifying evaluation to numerical values that could be easily processed, and allowing the incommensurable elements to a pairwise comparison in a rational approach.

Another popular approach developed by Laarhoven is the extension of AHP, based on fuzzy logic, which is used to deal with the fuzziness or uncertainty. The critical problem is how to infer definite conclusions from highly imprecise, vague, and ambiguous information. Classical logic requires a high understanding of the system/software product, whereas fuzzy logic provides a way to model a complicated system/software using a higher level of abstraction without going deep into the system/software. The procedure of quantifying software quality characteristics and sub-characteristics generally contains fuzzification and defuzzification stage, and the detailed steps could be referred [12]. Its procedure is similar to the classic AHP except for the two aspects that include:

a) Judgment matrix: Fuzzy pairwise comparison matrix is constructed with respect to fuzzy scale in this method compared with the pairwise comparison matrix in AHP method by comparing two elements, which is an extension of pairwise comparison matrix.

b) Relative importance: The method of deriving the relative importance of each element from the fuzzy pairwise comparison matrix is different from AHP. Because, for fuzzy AHP, popular methods for pair wise comparisons is triangular and trapezoidal membership function, that is different from the AHP method by 1-9 scale.

Researches have been proved that AHP method applied to qualitative evaluations is more precise compared to the direct evaluation; and it is a better choice when qualitative judgments are involved[13]. AHP method tends to deal with less vagueness multi-criteria problem (with few influence factors) in an easy way However, for complex, vague and uncertain problem, fuzzy AHP could gain more advantages than AHP method.

IV. Conclusion

This paper provides a framework of system/software quality evaluation based on the ISO/IEC SQuaRE series standards, and the relationships among quality model, quality measurement, quality requirements and evaluation process are depicted in detail. Also the analysis of basic quality evaluation process is given according to ISO/IEC 25040. Besides, analysis and suggestions of measurement function applied to quality evaluation are given. Finally comparison of typical techniques about quantifying quality characteristics (and/or sub-characteristics) with the measured values is discussed. Properly applying SQuaRE series standards would improve systems/software quality and act as an effective means of quality management.

Acknowledgment

The authors would like to thank the reviewers for their suggestions. The work in this paper is supported by National Natural Science Foundation of China [grant 61502299], the Shanghai Committee of Science and Technology [grants 15511107003 and 16511101202]. Jiayu Gong is also sponsored by Shanghai Sailing Program [grant 14YF1412700].

References

[1] ISO/IEC 25000 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Guide to SQuaRE[S].2014.

[2] ISO/IEC 25010 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-System and software quality models[S].2011.

[3] ISO/IEC 25012Software engineering-Software product Quality Requirements and Evaluation (SQuaRE) - Data quality model[S].2008.

[4] ISO/IEC 25020 Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)-Measurement reference model and guide[S].2007

[5] ISO/IEC 25021Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE) - Quality measure elements[S].2012.

[6] ISO/IEC 25022 Systems and software engineering-Systems and software quality requirements and evaluation (SQuaRE) -Measurement of quality in use[S]. 2016.

[7] ISO/IEC 25023Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Measurement of system and software product quality[S]. 2016.

[8] ISO/IEC 25030 Software engineering-Software product Quality Requirements and Evaluation (SQuaRE)- Quality requirements[S].2007.

[9] ISO/IEC 25040 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Evaluation process[S]. 2011.

[10] ISO/IEC 25041 Systems and software engineering-Systems and software Quality Requirements and Evaluation (SQuaRE)-Evaluation guide for developers, acquirers and independent evaluators[S].2012

[11] Kazuhiro ESAKI, System Quality Requirement and Evaluation-Importance of application of the ISO/IEC25000 series, Global Perspectives on Engineering Management, vol. 2 Iss. 2,pp. 52-59, 2013.

[12] P. J. M. Laarhoven and W. Pedrycz, A fuzzy extension of Saaty's priority theory, Fuzzy Sets Syst., vol. 11, pp. 229–241, 1983.

[13] Alessio Ishizaka, Comparison of Fuzzy logic, AHP, FAHP and Hybrid Fuzzy AHP for new supplier selection and its performance analysis, International Journal of Integrated Supply Management, 9(1/2), 1-22, 2014.