Research on International Standardization of Software Quality and Software Testing

Yi Zhao,Yun Hu,Jiayu Gong*
Shanghai Key Laboratory of Computer Software Testing &Evaluating
Shanghai Development Center of Computer Software Technology
{zy, huy, gjy}@sscenter.sh.cn

2021 IEEE/ACIS 20th International Fall Conference on Computer and Information Science (ICIS Fall) | 978-1-7281-7679-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICISFALL51598.2021.9627426

***Abstract*—Software industry is a key force leading a new round of technological revolution and occupies an important position in the development of the global economy. Software quality is of great significance to promote the healthy and orderly development of the software industry, and software testing is also a reliable means to ensure software quality. Standardization as a technical means to regulate software quality and software testing, can improve software product quality, reduce software development and testing costs. It can also provide guidance and support for the development of the software industry. This paper gives a detailed description of the international standardization of software quality and software testing and analyzes the relationship between software quality and software testing. Finally, it summarizes the problems and suggestions of software quality and testing standards. This paper is beneficial for demanders, developers, independent evaluation parties, quality assurance and control personnel to understand and use the relevant standards of software quality and testing.**

Keywords—Software quality; Software testing; Standardization

I. INTRODUCTION

Quality is the life of software. Low-quality software may bring about different levels of consequences, ranging from affecting the user experience to causing serious accidents. Therefore, people are paying more and more attention to software quality, and putting forward all-round requirements, such as more software functions, faster response speed, but also stable and safe operation. However, these all-round requirements can be tested by software testing, and its importance is self-evident. Software testing and software quality complement each other and can be used as one of the means to improve software quality.

In recent years, the standardization of software quality and software testing is developing rapidly and has become the trend of software quality development. ISO organization publishes a series of standards about system and software quality and testing. System and software quality standards mainly involves Systems and software Quality Requirements and Evaluation series of standards ("SQuaRE", also "25000 series of standards"), which specify quality model, measurement, requirement, and evaluation process applied to system and software quality assessment; and that can help to develop and acquire systems and software products.

Software testing standards are quite necessary, as it provides a method of validation for the developers, demanders to ensure software meet certain requirements or standards. Software testing standards are mostly dedicated to defining an agreed set of standards for software testing such as testing process, testing documentation, testing techniques, that can be used by any organization. The core of software testing standards is ISO/IEC/IEEE 29119 series of standards; other related standards consist work product reviews (ISO/IEC 20246:2017), process evaluation (ISO/IEC33063:2015), testing tool capability (ISO/IEC30130:2016), etc.

This paper presents international standardization of software quality and software testing, also provides the relationship between them. Section II depicts the software quality standardization, and analyses relationships among ISO/IEC 25000 series of standards. Section III provides software testing standards, details ISO/IEC/IEEE 29119 software testing standards. Section IV discusses the relationship between software quality standards and software testing, and Section V summarizes the problems and suggestions of software quality and testing standards.

II. SOFTWARE QUALITY STANDARDIZATION

From the perspective of the development of software quality international standards, in December 1991, ISO and IEC organizations published ISO/IEC 9126:1991 Software product evaluation—Quality characteristics and guidelines for their use. This standard defines six characteristics of software product quality and describes software product evaluation process model. As more and more attention are paid to quality, ISO and IEC organizations divide this standard into two related multipart standards: ISO/IEC 9126 (software products quality) and ISO/IEC 14598(software product evaluation).ISO/IEC 9126 consists four parts: quality model, external metrics, internal metrics, and quality in use metrics; ISO/IEC 14598 is divided into six parts, including general overview, planning and management, process for developers, process for acquires, process for evaluators and documentation of evaluation modules. With the development of software technology, software has gradually expanded to system, and traditional software engineering is also transferred to system and software engineering. At the same time, ISO/IEC JTC1 SC7 noted that there is a strong correlation between software

978-1-7281-7679-6/21/$31.00 ©2021 IEEE
ICIS2021-Fall, October 13-15, 2021, Xi'an, China

quality, software requirements, and software evaluation, while current standards are scattered and inconsistent. Based on these considerations, Systems and software Quality Requirements and Evaluation (SQuaRE) series of standards are proposed. Figure 1 shows a schematic diagram of software quality standards evolution.

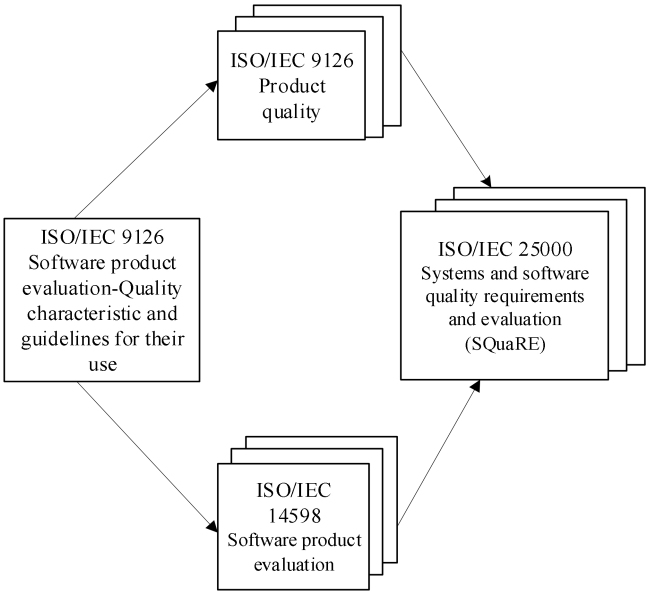

Fig. 1. ISO/IEC 25000 series of standards evolution

Software quality standardization is mainly based on ISO/IEC 25000 series of standards. This series of standards consists of the following divisions: Quality Management Division (ISO/IEC 2500n), Quality Model Division (ISO/IEC 2501n), Quality Measurement Division (ISO/IEC 2502n), and Quality Requirements Division (ISO/IEC 2503n), Quality Evaluation Division (ISO/IEC 2504n) and Extension Division (ISO/IEC 25050 to 25099). Figure 2 illustrates the organization of the ISO/IEC 25000 series representing families of standards.

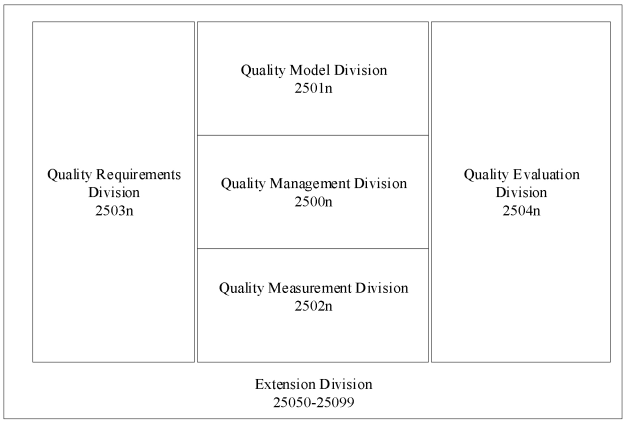

Fig. 2. Organization of ISO/IEC 25000 series of standard [1]

ISO/IEC 25000 series of standards establishes the criteria for the specification of system and software product quality requirements, measurement , and evaluation, aiming to assist in the developing and acquiring of systems and software products with the specification and evaluation of quality requirements[1]. Figure 3 presents the relationships among ISO/IEC 25000 series of standards. ISO/IEC 25010 and ISO/IEC 25012 as general guidelines give product quality and data quality models , which provide guidance for the proposal, implementation, testing, evaluation of quality requirements, and their standardization.ISO/IEC 25001, ISO/IEC 25020, and ISO/IEC 25040 are specific guidelines,which respectively give requirements and recommendations for managing quality requirements specification and evaluation activities, quality measurement framework, and the general process of quality evaluation. These guidelines provide basis and foundation for the definition of quality requirements, quality measurement and evaluation activities. In the process of specifying requirements, quality descriptions can be converted into quality requirements by ISO/IEC 25022, ISO/IEC 25023, and ISO/IEC 25024.Quality measures defined in ISO/IEC 25022, ISO/IEC 25023, and ISO/IEC 25024 can be used for evaluation process.ISO/IEC 25041 provides specific requirements and suggestions for quality evaluation by different roles(developers, demanders and independent evaluators). ISO/IEC 25045 provides the specification to evaluate the subcharacteristic of recoverability defined under the characteristic of reliability of the quality model.

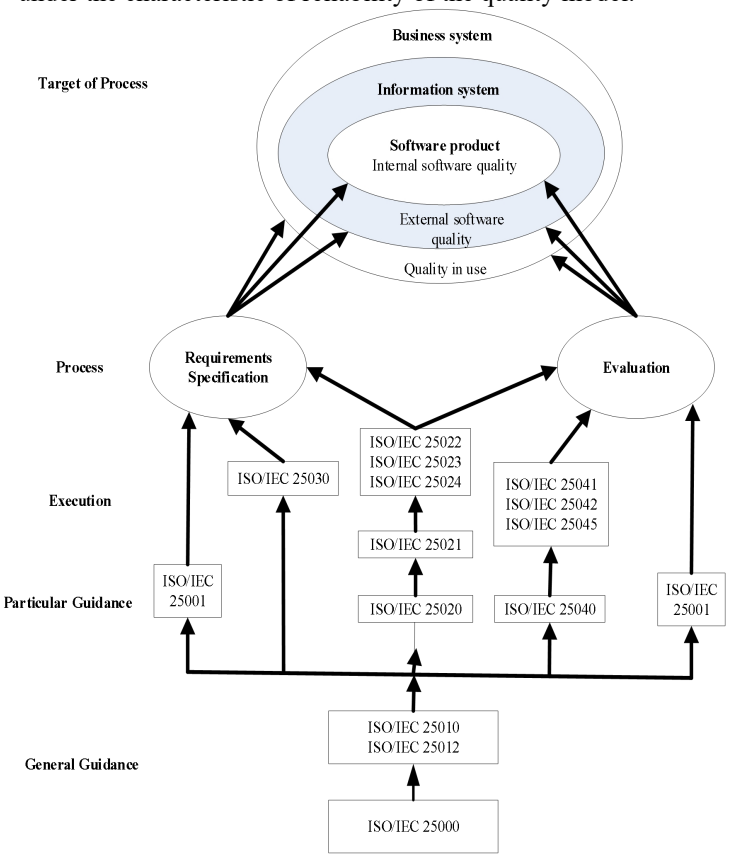

Fig. 3. Relationships among ISO/IEC 25000 series of standards [1]

In summary, ISO/IEC 25000 series of standards provide general quality model, measurement framework and evaluation methods. When using these standards, it is necessary to tailor standard adaptively according to specific fields, such as artificial intelligence and industrial app, and add customized modifications.

III. SOFTWARE TESTING STANDARDIZATION

ISO/IEC 29119 series of standards are formulated by ISO/IEC JTC1/SC7 WG26. Currently, six parts of ISO/IEC 29119 series of standards have been published, namely:

——ISO/IEC/IEEE 29119-1:2013 Software and systems engineering — Software testing — Part 1: Concepts and definitions

——ISO/IEC/IEEE 29119-2:2013 Software and systems engineering — Software testing — Part 2: Test processes

——ISO/IEC/IEEE 29119-3:2013 Software and systems engineering — Software testing — Part 3: Test documentation

——ISO/IEC/IEEE 29119-4:2015 Software and systems engineering — Software testing — Part 4: Test techniques

——ISO/IEC/IEEE 29119-5:2016 Software and systems engineering — Software testing — Part 5: Keyword-Driven Testing

——ISO/IEC TR 29119-11:2020 Software and systems engineering — Software testing — Part 11: Guidelines on the testing of AI-based systems

WG26 group is also actively developing other software testing standards, such as Guidelines for the use of ISO/IEC/IEEE 29119 in Agile projects (ISO/IEC TR 29119-6), Testing of software for automotive systems, model-based testing (ISO/IEC 29119-8), game testing, performance testing (ISO/IEC 29119-10). Figure 4 shows the framework of ISO/IEC/IEEE 29119 first four-part standards. These standards take the Part 2:test process as the principal line to carry out software testing, which cover testing at organizational, management, and dynamic test levels. Test documentation including templates and examples defined in Part 3 is produced during testing process. Test design techniques that can be used in testing process are given in Part 4. Common concepts and definitions used by other parts are defined in Part 1.

Part 1: Concepts and definitions. It specifies general concept of software testing, describes the role of software testing in an organization and project context, explains the general software life cycle, and introduces how to establish software testing process and sub-processes for specific test items or with specific test objectives. It also describes how software testing adapts to different life cycle models, clarifies how to use risk-based testing in each testing process, as well as how automation can be used to support testing [2].

Part 2: Test process. A multi-level test process model is established, including organizational test process, test management process and dynamic test process, and activity requirements of each test process are given. These processes can be combined with software development life cycle model [3].

Part 3: Test documentation. It provides templates and examples of software test documentation. That describes the output of each software testing process, including organizational test documentation, test management documentation, and dynamic test documentation (as shown in Figure 4) [4].

Part 4: Based on the test process in Part 2, three types of test design techniques are given, including specification-based test design techniques, structure-based test design techniques and experience-based test design techniques (see Figure 5). These test design techniques can provide specific guidance on the derivation of test conditions (TD2), test coverage items (TD3), and test cases (TD4) activities in the test design and implementation process. In specification-based testing, requirements, specifications, models, or user needs can be used as test basis to design test cases. In structure-based testing, structure of the test item, such as source code or the structure of a model, can be used as the main source of information to design test cases. For experience-based testing, knowledge and experience of the tester can be used as the basis to design test cases [5]. These three types of testing design techniques are complementary, rather than isolated. It is recommended that the combination of these techniques will make the testing more effective.

At present, there are few relevant standards for test verification, test methods, and test tools; and the guidance of software testing activities is limited. ISO/IEC organization is also actively planning relevant standards for software testing according to industry needs, and constantly enriching software testing standards.

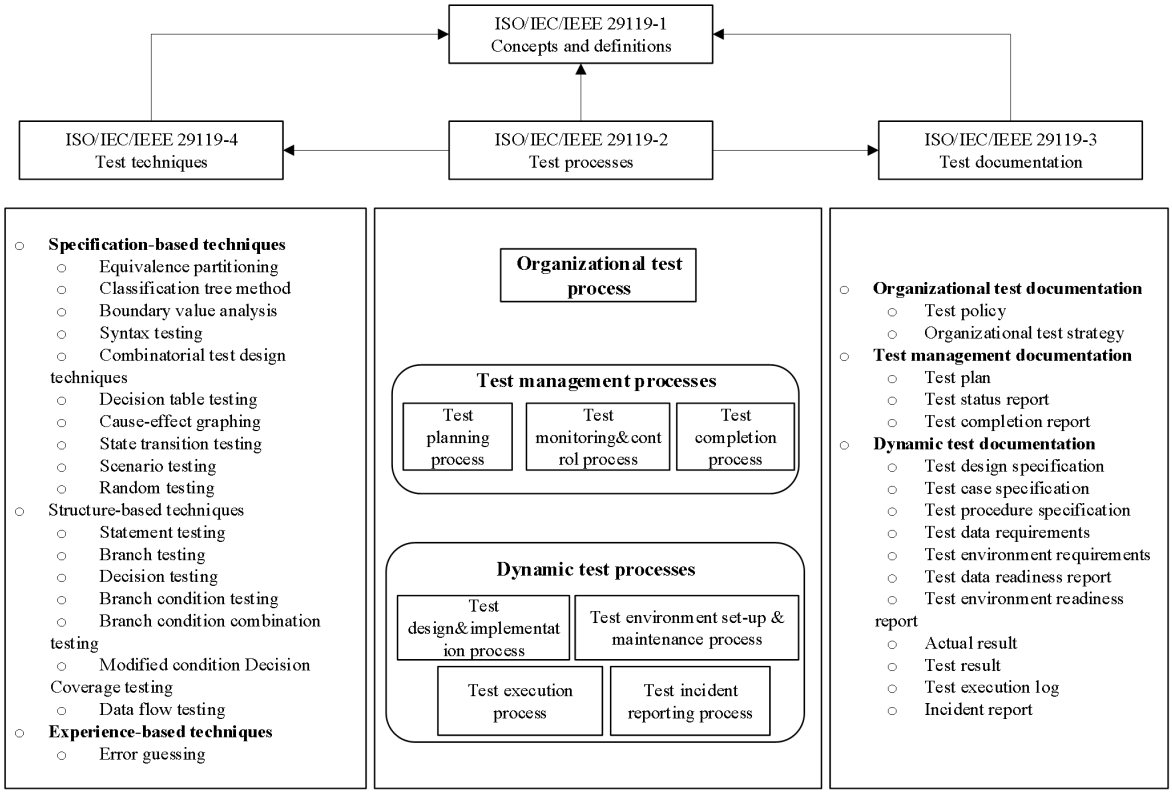

Fig. 4. ISO/IEC/IEEE 29119 framework

IV. RELATIONSHIP BETWEEN SOFTWARE QUALITY AND SOFTWARE TESTING

Software testing is a means to improve software quality, and software quality is purpose, that is, software testing is the necessary and insufficient condition for improving software quality. Software testing standards help improve the quality and efficiency of software testing, thereby promoting the improvement of product quality. Figure 5 shows the relationship between software quality and testing.

ISO/IEC 25000 gives general requirements for system and software quality, and provides guidance for the use of ISO/IEC 25000 series of standards. Measurement reference model and guide mainly include quality measurement framework and method. System and software measurement results can be obtained by applying this module, which also provide information for quality evaluation, and as the input of the evaluation process for comparison between systems, software products and data, respectively. In addition, you can also compare the measurement results over time and analyze the trends to obtain the changes in software quality. For example, software quality assurance personnel can use quality measure "estimated latent defect density" to evaluate the estimated latent defect density during the qualification testing. The trend of this measure can be used to assess the status of defect removal and improve software reliability, and as a part of the quality assurance process.

Example 1[6]:

Estimated latent defect density=(C1-C2)/S

—C1: Total number of predicted latent faults in a system and software product.

—C2: Cumulative number of unique faults detected.

—S: Product size.

Quality Measure Element (QME) constructs quality measures of system and software product quality, quality in use, and data quality by applying measurement methods to specific attributes; and records measures that obtained by the combination of measurement functions when necessary. These measures can be used for quality requirements definition, product evaluation and quality assessment. QME can be obtained by software testing, and the testing provides quality source for related activities. For example, operation time and the number of system/software failures in Example 2 can be acquired by testing and counting the number of system/software product failures during the execution of corresponding task within a certain period.

Example 2:

Mean time between failure (MTBF)=A/B

—A: Operation time.

—B: Number of system/software failures occurred.

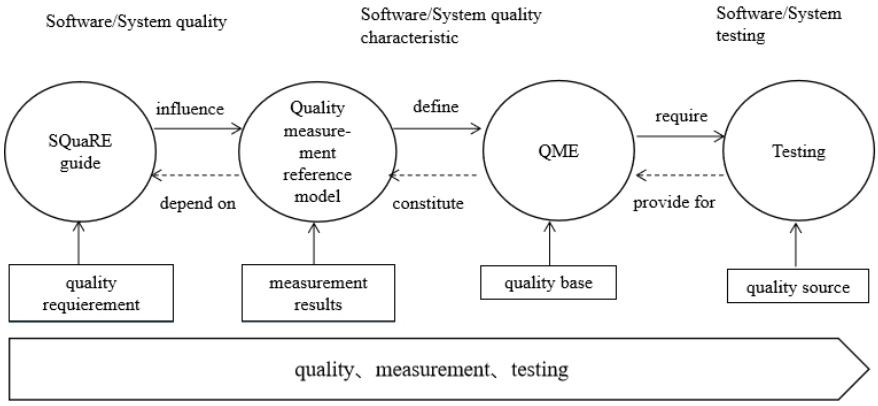

Fig. 5. The relationship between software quality and testing

When obtaining the number of system/software failures occurred by testing, ISO/IEC/IEEE 29119 can be used to design and implement testing, and the QME result could finally get by testing documentations such as test execution log, incident report. Then combine the QME result through measurement function to get QM results. Figure 6 briefly describes this process.

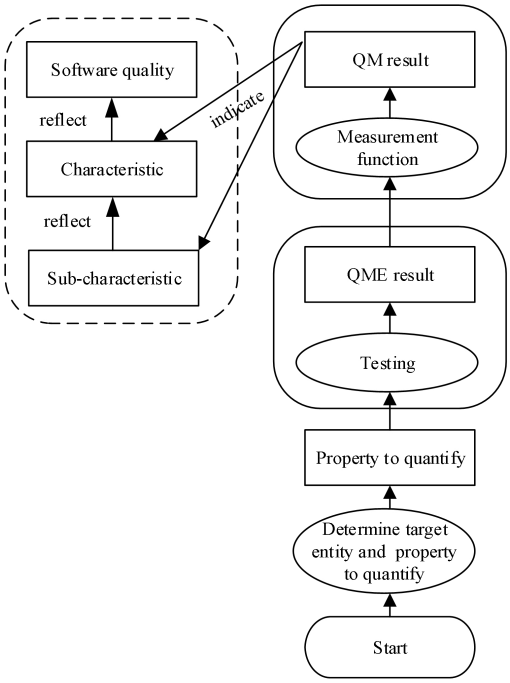

Fig. 6. The process of quality measurement by testing

Besides, ISO/IEC/IEEE 29119-4 Annex A provides the examples of how the characteristics defined in ISO/IEC25010:2011 can be mapped to the testing types and test design techniques [5]. It covers 16 types of quality-related testing, such as compatibility testing, backup/recovery testing, disaster recovery testing, performance-related testing, etc. Test design techniques defined in ISO/IEC/IEEE 29119-4 can be applied to the quality-related testing. A test design technique may be mapped to several characteristics and sub-characteristics, for example, boundary value analysis technique can be used to test the characteristics of functional suitability, performance efficiency, usability, reliability, and security. Likewise, testing cases of a characteristic can be designed by several testing design techniques, for example reliability characteristic maps to the techniques of boundary value analysis, equivalence partitioning, random testing, state transition testing, error guessing.

V. PROBLEMS AND SUGGESTIONS

Recent years, relevant international standardization organizations have actively planned and developed software quality and software testing standards. These standards have been also examined as a tool for performing various analyses and evaluations on systems and software quality. However, there are still some problems. Some problems and suggestions are given below.

1) There is an urgent need in the field of emerging technologies.

[Problem]

From Chapter II and Chapter III, we can see that the existing software quality and software testing standards are mainly oriented to traditional information technology, and they provide general quality models, quality measurement, quality evaluation, and testing process methods, which cannot be directly applied to emerging technology, such as big data, blockchain, etc.

[Suggestion]

It is suggested that standards in emerging technology fields be developed based on the emerging industries needs and in combination with existing software quality and testing standards. And ISO/IEC JTC1/SC7 WG6 has launched a study group on the future direction of the SQuaRE series to catch up the advent of new technologies, for example, enhancing the quality models, and reporting in TRs and to provide guidance to apply the models to the new technologies [7].

2) The relationship between software quality and software testing standards is relatively independent.

[Problem]

Among the existing standards, the correlation between software quality and software testing standards is relatively low. Most of the standards related to quality measurement provide some quality measures, without specific guidance on how to obtain the corresponding quality measure element by relevant testing.

For example, quality measure "mean response time" in ISO/IEC 25023 (PTb-1-G), the standard only provides the description and measurement function of this measure (see Table I [8]). No guidance is given for the acquisition of A_i in the measurement function.

TABLE I. MEAN RESPONSE TIME MEASURE

ID	Name	Description	Measurement function
PTb-1-G	Mean response time	How long is the mean time taken by the system to response to a user task or system task?	$X = \sum_{i=1}^{n} (A_i)/n$ A_i=Time taken by the system to respond to a specific user task or system task at i-th measurement n=Number of responses measured

Besides, at present there are few standards on testing methods, and quality measurement standards are not combined with specific testing methods.

[Suggestion]

Therefore, it is suggested that software quality measurement standards should be combined with software testing standards in the way of implementation guidelines or TRs, to enhance the operability of quality measurement elements and to achieve coordination between standards.

When implementing the quality measurement standards, it can be combined with enterprise internal standards and operation instructions to obtain quality measurement element values.

3) Enhance the operability of ISO/IEC 25000 series of standard

[Problem]

Some standards are a little abstract, and it is difficult for the users to make full use of them in practical applications, for example ISO/IEC 25030 and ISO/IEC 25040. It is necessary to provide guidance on how to use these standards.

[Suggestion]

The guidance could be in TRs, providing guidelines and examples in different systems or usage scenarios based on the high demand area, to promote the operability of ISO/IEC 25000 series of standard.

4) Strengthen the application of ISO/IEC 25000 series if standard in software development process.

[Problem]

In practical application, ISO/IEC 25000 series of standard is more used to evaluate the quality of the developed system and software, while ISO 9000 and CMMI are mostly used for quality management in the development process. ISO/IEC 25000 series of standard were not adopted in the whole process, resulting in the disconnection between quality of the development process and the quality of the final system and software product. There are problems of the qualitied quality of development process and the unqualified quality of the final product.

[Suggestion]

It is recommended to introduce ISO/IEC 25000 series of standards at all stages of the development process, to make up for the lack of quality measurement index requirements of ISO 9000 and CMMI. It can make full use of the advantages of the two types of quality standards to ensure that the final system and software meet the quality requirements.

All in all, in the process of standard development, software quality should be taken as the core, software testing techniques and methods as the support; and the standards development of software quality and testing should be strengthened in combination with the emerging technical fields, and further enrich standard system. So that software quality and software testing standards are interrelated and interactive.

VI. CONCLUSION

This paper discusses international standardization of software quality and software testing. Taking ISO/IEC 25000 series of standards and ISO/IEC/IEEE 29119 as examples, the paper analyzes the relationship between software quality and software testing, and finally summarizes the problems existing in current software quality and software testing standards and gives some suggestions. Properly applying system/software quality and testing standards would contributes to specify system/software quality requirements, evaluate software quality, improve the ability to manage and execute software testing, thereby improving the quality of system and software.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their suggestions. The work in this paper is supported by China's National Key R&D Program (No.2018YFB1403404).

REFERENCES

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

[2] ISO/IEC/IEEE. Software and systems engineering—Software testing—Part 1: Concepts and definitions: ISO/IEC/IEEE 29119-1:2013[S]. Switzerland:ISO,2013.

[3] ISO/IEC/IEEE. Software and systems engineering—Software testing—Part 2: Test processes: ISO/IEC/IEEE 29119-2:2013[S]. Switzerland:ISO,2013.

[4] ISO/IEC/IEEE. Software and systems engineering—Software testing—Part 3: Test documentation: ISO/IEC/IEEE 29119-3:2013[S]. Switzerland:ISO,2013.

[5] ISO/IEC/IEEE. Software and systems engineering—Software testing—Part 4: Test techniques: ISO/IEC/IEEE 29119-4:2015[S]. Switzerland:ISO,2015.

[6] ISO/IEC. Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Quality measurement framework: ISO/IEC 25020:2019 [S]. Switzerland:ISO,2019.

[7] Nakajima T. Study group report on square future direction. CEUR Workshop Proceedings. 2020; 2545:1-5.

[8] ISO/IEC. Systems and software engineering—Systems and software Quality Requirements and Evaluation (SQuaRE)—Measurement of system and software product quality: ISO/IEC 25023:2016[S]. Switzerland:ISO,2016.