SquaRE The next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality

Motoei AZUMA

Abstract
In this presentation, overview of current 9126 and 14598 series are introduced followed by some problems to be solved. Then the concept and architecture of SQuaRE are discussed. Finally, new parts, i.e., Quality in Use, Quality Requirements, and Elementally Metrics are explained.

1. Introduction
Software product quality has been increasingly important in the computerized society. Quite a few efforts have been made and contributed for improving the quality. A software product, which simply does not contain any defect and therefore is reliable, satisfies only the necessary condition. It can claim to be sufficient quality only when it satisfies requirements for other quality characteristics, such as functionality and usability.

ISO/IEC 9126-1991 [1] defined six quality characteristics. Since then, ISO/IEC JTC1/SC7/WG6 has contributed to develop two series of international standards on software product quality, i.e., ISO/IEC 9126 and ISO/IEC 14598 series. Five parts of ISO/IEC 14598 (Part 1 through Part 5) have been already published [2], [3], [4], [5]. ISO/IEC 9126-1 and ISO/IEC 14598-6 will be published soon. Remaining * Parts, i.e. ISO/IEC 9126-2, -3 and – 4, will be finished soon as they are at the final letter ballot stage.

While working on the two series of standards, SC7/WG6 identified additional problems and issues in the series for the further improvement. However, in order to avoid unnecessary delay in the project, WG6 decided to solve these problems and issues in the next generation of the series. It means that WG6 finishes the existing task first and then start editing new documents. The plan for the next generation of 9126 and 14598 series has been discussed in parallel with the work on the current series.

Initial idea of the SQuaRE as well as the name was introduced in the WG6 Kanazawa meeting in November 1999 and refined in the Madrid meeting in May 2000. The plan was introduced at the SC7 Madrid plenary meeting and approved by the SC7. This paper introduces the concept and outline of the SQuaRE. The SQuaRE plan will be realized after the formal procedure of both JTC1 and JTC1/SC7. Therefore, the plan, contents and document numbers may be modified by discussion in SC7 and SC7/WG6.

2. History and achievements of ISO/IEC 9126 and 14598 series
The project 7.13, which was originally responsible for “Software product quality evaluation”, was started in 1985, by nominating a list of quality characteristics and their definitions. The WG members initially referred to Boehm model [7] and McCall model [8]. It also decided that quality evaluation process be included in the standard.

However, membership in the WG was not stable, which made the work not so productive. The 1st-DP (Draft Proposal) was submitted in 1986. The international standard was published in 1991 as “ISO/IEC 9126: Information technology-Software product evaluation-Quality characteristics and the guidelines for their use”.

Based on the needs for more information on the software product quality, the project 07.13, which was started as a single standard project, was divided into three sub-projects; 07.13.01 Quality Characteristics, 07.13.02 Quality subcharacteristics, and 07.13.03 Measurement and Rating, at the SC7 Budapest meeting in November 1989.
Based on the growth in technologies and advance in the work, the project 07.13 was re-arranged as five sub-projects in June 1992, and as eight sub-projects at the SC7 Tokyo meeting in May 1993. Then the project 07.13 was divided into 07.13.01 and 07.13.02, which are responsible for ISO/IEC 9126 and series respectively, at the SC7 Ottawa meeting in May 1994. Next May in 1995, the titles of the ISO/IEC 14598 series were modified. WG6 further decided to modify the name of ISO/IEC 9126 series and 9126-1, and proposed to add new part 9126-4 Quality in use metrics [9]. As the result, current ISO/IEC 9126 and 14598 series are as follows [10].


3. Problems of ISO/IEC 9126 and 14598 series

ISO/IEC 9126-1991 has been and widely referred and used and is considered as a successful standard. Some countries including Japan national body (JISC) have adopted it as a national standard. However, though the last part (ISO/IEC 9126-1: Part 4 Quality in use metrics) was added fairly recently in 1998, while standardization work was under way, WG6 identified new problems and issues for further improvement. They are caused mainly by changes in environment and advances in the information technologies.

Needs for unique new architecture and umbrella guide

There are two international standard series, i.e. ISO/IEC 9126 and 14598, which are closely related to each other. It is difficult to remember those meaningless numbers. WG6 considered that these facts might be a factor that makes difficult to be more popular.

Good architecture is always important not only for a software system but also for a series of standards, because a series of standard is also a kind of software system. Needs for a good guide is identified. It is also important that the series of standard has the identification name of the series, which is easy to identify and remember.

![Figure 1: Current architecture of ISO/IEC 9126 and 14598 series](image-url)
New issues on metrics

While discussing metrics standards, some new issues were raised.

There are those metrics that are commonly used for composing different kinds of metrics, such as internal metrics and external metrics.

Some metrics, which is necessary for evaluating the product quality, are used for measuring not an attribute of the software product. Examples are compilation times, and total man-hour for reviewing. Such metrics are mainly used for normalization, for example, the number of fault divided by testing man-hour. Therefore, this kind of metrics should be listed in the series.

There are such metrics that have no verified direct co-relation with quality characteristics, but generally recognized as related with product quality. Examples are Cyclomatic Complexity, Cohesion (of a module), and fan-in. WG6 tentatively calls this class of metrics as Pure Internal Metrics.

Some important information on metrics is distributed in several parts of the series, making difficult to find and understand all related necessary information for an in-depth understanding. For example, general requirements for metrics are listed in ISO/IEC 9126-1 as normative reference, while the common section that explains metrics in general are in each of 9126 Part 2, Part 3, and Part 4.

(Note: Two kinds of the term “requirement” are used in this paper. Italic “requirements” means the “requirement” by the standard, which the user of the standard must conform. Normal “requirements” means as that is used commonly in the software engineering.)

Lack of quality requirement standard

As stated before, the original ISO/IEC 9126 series was separated into 9126 and 14598 series, because the quality model and metrics are useful not only for product evaluation but also for other purpose including quality requirements specification. Quality evaluation is possible and meaningful only when quality requirements are clearly specified. However, if the quality requirement standard be proposed not as a part on the series but as an independent standard, then it will cause another confusion to the users. Therefore WG6 decided that the quality requirement be proposed as a part of the next major update of the standard series.

4. SQuaRE overview
4.1. Concept

Taking the above issues and other technical issues for improving the series, SC7/WG6 decided to redesign the architecture of the series and named it SQuaRE (Software Quality Requirements and Evaluation).

There are two important aspects, product aspect and process aspect in the field of software quality assurance. SQuaRE focuses on the product side. SQuaRE recommends the use of a quality model, which refines the required quality into characteristics and subcharacteristics and clarifies the relationship among them.

A software product, which only contains few defects or faults and therefore behaves correctly, cannot be regarded as a high quality product. It is very important that the product meets the users needs. In other words, the product must fit the intended users’ characteristics, such as experience of keyboard operation and knowledge of the graphical user interface, and purpose of use as well as the target task.

The basic concept is that in order to achieve product quality, quality requirements be defined for all the quality characteristics based on the quality model and associated metrics. During development, test and use/operation process, measurement and evaluation are carried by using quality model and associated metrics again. Some important concepts of SQuaRE are as follows.
**Lifecycle View**

In order to develop, acquire or use a software product of satisfactory quality, it is necessary that the quality be identified, planned, measured and evaluated through its lifecycle using quality metrics based on the quality model.

At first, needs should be identified from the users’ view, by identifying the purpose of use, target task, users’ profile or characteristics, and effects of use as well as the expected results. These identified needs are transformed into requirement specifications as quantitatively as possible using external metrics. The specified requirements are useful as the evaluation criteria. Then the specified requirements are transformed into the design. Design strategies and goals are to be clarified at the beginning of the design stage. SQuaRE and lifecycle are explained more in detail in clause 6.

**Quality model**

In order to acquire the suitable quality product, various quality characteristics and subcharacteristics, such as usability and efficiency, should be. Quality model is the set of characteristics and sub-characteristics and the relationships among them. In general, a quality model depends on the category of target product. For example, a car and a television need different quality model. However, if a new quality model is developed for every time when a product in a same category is developed or acquired, it is waste of human resource. Therefore, standard quality model is useful.

ISO/IEC 9126 was developed for this reason. It recommends it be used as a default model, and make change on it when it is necessary. ISO/IEC 9126-1 improved the original model by defining subcharacteristics, separating internal and external characteristics and adding quality in use characteristics. SQuaRE will inherit the model. However, the issue of clarifying the relation between the model and pure internal attributes remains unsolved.

**Metrics and measurement**

A software product is an entity that has various properties, which makes it difficult to measure the entity itself directly. That is why quality characteristics are used. In order to define quality requirements and to evaluate the product quality, each quality characteristic and subcharacteristic should be measurable.

However, quality characteristics and subcharacteristics are not measurable directly by their nature. For example, many attributes, such as an icon, a pull down menu, and an error message contribute usability. Therefore, it is suggested that those attributes be measured using base metrics. Then the measured values are transformed into the higher level of value that indicate a characteristic or subcharacteristic, using a formula, which compose a quality metric.

Therefore, most of the metrics provides a formula. A variable that compose the formula is not always obtained by measuring an attribute of the target entity. For example, a measure that indicates security of a software product (target) is obtained by measuring the number of the failed attempts (users’ behaviour) and the number of tried attempts to unauthorized access. Depending on the stage of the lifecycle, different attributes should be measured, because different materials, such as specification, source code, executable codes are available for measurement.

In the case of software measurement, it is widely known that the value for the same measure, for example LOC (Lines Of Code), may be quite different, if measurement is carried by using different measurement method (or counting rule). Therefore, the “Metric” is defined as “the defined measurement method and measurement scale”

**Management**

SQuaRE encourages the users to establish supporting function, which may be independent organization, department or a person in charge of it. The requirements for
the supporting function and guides are in the “Part 11: Planning and management”. Their role is to acquire the new standards, technique and tools, as well as to carry out technology assessment, technology transfer and technology management.

Because information technologies are changing and evolving very rapidly, it is very important for the organization that develop and/or acquire a software product to watch and trace the advance and evolution in the related technology, assess them and transfer them to the relevant projects.

**Provision of html documents**

Different from, for example, hard disc specification standards, SQuaRE is aiming at a large number of and wide variety of audiences. Therefore, it is necessary that sufficient information be provided before they purchase the standards.

SQuaRE is the Multi-parts standard, which consists of total 14 Parts. Most of them are closely related to each other. Therefore, SC7/WG6 is investigating the possibility of publishing the SQuaRE as a set of hyper-media documents. It is also planning to provide headline information of each part by WWW (World Wide Web).

### 4.2. Architecture

Based on the above concept, the SQuaRE architecture was defined.

SQuaRE consists of five divisions, such as software quality general division, quality model division, quality metrics division, quality requirement division, and quality evaluation division, which are shown in figure 2.

![Figure 2: Architecture of the SQuaRE](image)

5. **Outline of each division and part**

The first division of SQuaRE is located in the middle of the “square”, which the remaining four divisions compose. This first division states the general requirements for a software product quality, and explains the overview of the SQuaRE and how to manage
technologies necessary for improving software product quality by using the SQuaRE.

5.1. Software product quality general division
This division states the general requirements, overview of the SQuaRE and how to manage the technologies necessary for the use of SQuaRE. This division consists of two parts, Part 10: General overview and guide, and Part 11: Planning and management.

Part 10: General overview and guide to the SQuaRE (new)
The purpose of the Part 10 is to states general requirements for the software product quality and to provide guidance for the use of the series of documents that compound the SQuaRE model. It introduces the SQuaRE model, explaining its architecture, the planned evolution, the other parts and possible usage. It also provides a general overview of the evaluation process, as well as of the quality model, becoming an important document for the comprehension of the complete series.

Part 11: Planning and management (revision of 14598-2)
The purpose of the Part 11 is to provide explanation about the planning and management for use of quality model and metrics for quality requirements specification and evaluation. It includes requirements, recommendations and guidelines for a supporting function, which is responsible for acquiring, standardizing transferring and managing technologies in the area of SQuaRE. It also gives guideline for assessment of the organization for supporting function and product evaluation.

5.2. Quality model * division
“Quality model” is the set of characteristics and the relationships between them, which provide the basis for specifying quality requirements and evaluating quality. This division states the general requirements for a quality model, recommended model, and guides to customize and use the model. It consists of the following single part.

Part 20: Quality model and guide (revision of 9126-1)
The purpose of the part includes describing a model for software product quality, which consists of two parts, i.e., internal and external quality model part and quality in use model part, and guiding users to apply the model to a specific product. It includes characteristics and sub-characteristics for internal and external quality, characteristics for quality in use, and guide to use the model.

5.3. Quality metrics division
“Metric” is the defined measurement method and the measurement scale. This division consists of general requirements for quality metrics, the lists of recommended metrics, and guide for use the metrics. It consists of the following six parts.

Part 30: Metrics reference model and guide (new)
This part gives general information about the quality metrics as an umbrella document of the metrics division. It provides introductory information, the reference model and the definitions that are common to a set of base metrics, internal metrics, external metrics, and quality in use metrics. It includes the general requirements (shall) for use of metrics, which is currently stated in the 9126-1 normative annex. It also includes the guidance to users for selecting or developing, and applying metrics from the Part 31, Part 32, Part 33 and Part 34.

Part 31: Base metrics (new)
This new part will define and specify a set of recommended base metrics, which can be used during the whole software development lifecycle. Base metrics are specific by their nature and form the common denominator for several software quality characteristics or sub-characteristics within the SQuaRE. This set of metrics can be used not for the quality
evaluation only, but also for the prediction of other software engineering characteristics (e.g. development effort, minimal coding and testing time).

This part will contain: a set of base metrics, guides to use base metrics, methods of the base metrics data gathering, a set of data (measured values) recommended to be recorded during the lifecycle, and an example of use of base metrics.

**Part 32: Internal metrics (revision of 9126-3)**

This Part defines internal metrics for quantitatively measuring internal software quality in terms of characteristics and sub-characteristics defined in ISO/IEC 9126-20. It includes a basic set of internal metrics for each sub-characteristic, and guides and examples of how to use internal metrics during the software product life cycle.

**Part 33: External metrics (revision of 9126-2)**

This Part defines external metrics for quantitatively measuring external software quality in terms of characteristics and sub-characteristics defined in ISO/IEC 9126-20. It includes a basic set of external metrics for each sub-characteristic, and guides and examples of how to use metrics during the software product life cycle.

**Part 34: Quality in use metrics (revision of 9126-4)**

This Part defines quality in use metrics for the quality in use characteristics defined in ISO/IEC 9126-20. It includes a basic set of quality in use metrics for each quality in use characteristic, and guides and examples of how to use metrics during the software product life cycle.

**9126-35: Documentation of evaluation module (revision of 14598-6)**

An evaluation module is a package of evaluation technology for measuring software quality characteristics, sub-characteristics or attributes. Currently this part is a part of 14598 product evaluation series. However, as it defines how to specify a metric and how to use it, it is more reasonable that it be a sub-part of new metrics part. This part defines the structure and content of the documentation to be used to describe an Evaluation Module.

### 5.4. Quality requirement division

This division consists of single part that includes quality requirements and guide for use of it.

**Part 40: Quality requirements (new)**

This Part enables software product quality to be specified as quality requirement. The requirement to be tracked, validated and managed with evaluation from different perspectives by those associated with acquisition, requirements analysis, development, use, evaluation, support, maintenance, quality assurance and audit of software. It also includes guide to use the model and metrics for requirement definition.

### 5.5. Quality evaluation division

This division consists of the following four parts.

**Part 50: Evaluation process overview (revision of 14598-1)**

This part succeeds 14598-1, but inherits only general part for product evaluation. It introduces the other parts of this division. It contains general requirements for specification and evaluation of software quality and clarifies the general concepts. This part provides a framework for evaluating the quality of all types of software product and states the requirements for methods of software product measurement and evaluation.

**Part 51: Developers process (revision of 14598-3)**

This part provides requirements and recommendations for the practical implementation of software product evaluation when the evaluation is conducted in parallel with the development and carried out by the developer.
Part 52: Acquirers process (revision of 14598-4)

This part contains requirements, recommendations and guidelines for the systematic measurement, assessment and evaluation of software product quality during acquisition of “off-the-shelf” software products, custom software products, or modifications to existing software products.

Part 53: Evaluators process (revision of 14598-5)

This part provides requirements and recommendations for the practical implementation of software product evaluation, when several parties need to understand, accept and trust evaluation results. In particular, it may be used to apply the concepts described in ISO/IEC 9126-1n.

6. Guides to use the SQuaRE

SQuaRE can be used by various users, for various purposes and, to various products. Examples of SQuaRE are software product developers, users and acquirers, as well as quality assurance staff and managers. Guides for using 9126 and 14598 series are scattered in the current version, which make the use of the series difficult. Therefore, all the necessary information to use SQuaRE at the beginning is contained in the ISO/IEC 9126-10. ISO/IEC 9126-10 explains the relation among QIU (Quality In Use) characteristics, external characteristics, internal characteristics, and software lifecycle (Figure 3).

At the beginning of software product development, it is necessary that needs be clearly identified. The needs can be specified using QIU metrics. Then requirements are specified as the external specification in order to satisfy the needs. The requirements and their criteria can be specified using external metrics. The requirements are converted into the internal design specifications. Internal metrics can be used for specifying the criteria of the design. Typical use of SQuaRE is as follows.

![Figure 3: ISO/IEC 9126-1 U-Model](image-url)
6.1. Needs analysis stage

Needs is the expectation for the effects of the product when it is used. As requirements do not always reflect real users’ needs, needs analysis is the most important process of the software lifecycle. By the definition of QIU, it varies depending on the context of use.

Quality in use scenario or use case scenario may be useful for identifying the context of use. Then the listed scenarios must be classified, selected and transformed into the context of use. The next step is to categorize each selected context of use, using such criteria as target task, method of usage, environment, and frequency of use. Finally, identified needs can be specified for each QIU characteristics. The level of satisfaction should be defined for each context of use, using the indicator. In order to define level of satisfaction, at least one indicator should be defined for each QIU characteristic.

6.2. Requirement analysis stage

The next activity is to specify external quality requirements based on the defined user quality needs. There is no automated procedure for the conversion from identified needs to the requirement specification. Each context of use and associated level of satisfaction for each QIU characteristic must be examined when the conversion is done. Quality requirements shall be defined for every external quality characteristics, possibly using external quality metrics that is defined in part 33.

6.3. Design and implementation stage

In order to design and develop the software product that satisfies requirements, the designer should develop a design strategy, which includes selecting appropriate design method and tools as well as defining architecture, at the beginning of the stage. For example, good and solid architecture can improve maintainability, and object oriented technique can improve usability. For the purpose, 9126-10, -11, -20, and –30, -31, 32 may be helpful. However, as SQuaRE focuses on product side, it may not help the work directly, while designing and implementing the product.

6.4. Formal review stage

Though this stage may vary depending on the lifecycle model and process design, the software product evaluation should be done iteratively depending on the previous design and implementation stage. The review can be categorized into design review and code review. The former uses the design documents as the target, and the latter uses source code for review. None of them is usually executable. Therefore, internal metrics in 9126-32 are useful in this stage.

However, when such lifecycle model as prototyping or incremental development is applied, external metrics and, some times, quality in use metrics are also useful.

6.5. System test stage

The target software is executable, in this stage. The target attributes of measurement are behaviours of the system. In most of the case, external metrics are used. The measured values are compared with those criteria that are defined in the requirement specifications. As developers, acquirers, and third party evaluators may be involved in the evaluation process, they can be possible users of SQuaRE. Therefore, 9126-33 as well as 9126-51, 52, and –53 are main parts of SQuaRE in this stage.

6.6. Use and maintenance stage

After the users begin to use the software product, it is important that real contexts of use
be collected with measured values, that data be analysed and to use the information for the next version. For this reason, QIU characteristics are measured at the users’ site. Measured values are transformed into the higher level of measures or indicators, and the results are to be compared with the evaluation criteria. Finally, the results are to be interpreted and reported.

7. Conclusion
The work for developing SQuaRE has been recently just initiated. The plan of SQuaRE, as summarized in this paper, is the initial input to the project. SC7/WG6 invites more contributions to improve SQuaRE. It may be modified and should be continuously improved because, both technologies for the SQuaRE and the target software are changing very rapidly, and usually, the work for developing standard is time consuming because of the necessary formal balloting procedure.

8. references

9. Acknowledgement
The author expresses his sincere appreciation to WG6 members, especially editors, co-editors and WG6 secretariat for their continuous efforts and contributions. They are Nigel Bevan, Jorgen Boegh, Gilbert Le Gall, Michael Gayle, Vipula Godamunne, John Harautz, Tom Ipoly, Antonia Jeanrenaud, Ota Novotony, Danilo Scalet, Renate Sitte, Witold Suryn, Yukio Tanitsu, Jiri Vanicek, Atsushi Yamada, and Toshihiro Komiyama as the secretariat.