

СТРУЧНИ ЧЛАНЦИ  
ПРОФЕСИОНАЛНЫЕ СТАТЬИ  
PROFESSIONAL PAPERS

## SOFTWARE QUALITY STANDARDS

*Nebojša D. Đorđević*

Serbian Armed Forces, Army HQ, Niš, Republic of Serbia

e-mail: djnebojsa@mts.rs,

ORCID iD:  <http://orcid.org/0000-0003-3730-0085>

DOI: 10.5937/vojtehg65-10668

FIELD: Computer Sciences, IT  
ARTICLE TYPE: Professional Paper  
ARTICLE LANGUAGE: English

### *Summary:*

*In order to achieve the desired software quality, it is necessary to have information about the existing standards in this field and follow them in all phases of the software life cycle. This study emphasizes the importance of applying the standards in the field of software quality. In the last 20 years, the International Organization for Standards has played an important role in the development of quality standards of a number of software products. At the beginning of this work, the evolution of standards in the field of software quality is summarized and then the current series ISO/IEC standard called SQaRE are described in detail. This area is very important because the knowledge of software quality standards and their application contributes to the development of quality software that will be widely used and durable. The application of the described standards and the evaluation approaches for the quantification of software quality depends on the resources available for carrying out the method (time, money, the number of evaluators and their expertise, the number of users for testing, facilities and test equipment), the required level of objectivity and its application to various stages of the development of applications.*

*Key words: ISO, quality standard, software quality.*

## Introduction

Software is an intellectual product and its acquisition represents a significant investment which must be justified. In order to justify such

investments, it is necessary that software meets the needs of users at the expected level.

The justification of investments is often assessed and there is always a dilemma whether the right decision is made regarding a choice or purchase. A mental process of evaluating the functionality and other measurable characteristics is performed in the context of real needs. For investments in intangible assets, such as software, mental processes are more complex and facts are abstract and unconvincing. Therefore, methods for the formalization of mental processes of decision-making are necessary, based on clear and unambiguous assessment results.

Precise contracts for the purchase of software and respect are the best way to protect both the contracting authority and the supplier. It is preferable that contracts governing this professional matter, in order to avoid misunderstandings regarding the matter and its concepts, rely on adequate standards. In addition to providing legal protection for suppliers (particularly in cases where the use of product due to human error can result in endangering the lives and health of people or greater material damage), the use of standards ensures that the quality of the development process and the end products are at the required and expected level.

With the increasing number of software applications, the importance of software quality also increased and, to manage software quality, it was necessary to establish a methodology for objective quantitative evaluation of software products and the process of their development.

To define software quality, it is necessary to measure and evaluate many features that allow the determination of software quality, where the metrics of software quality plays a significant role. Therefore, it is necessary to define the characteristics that significantly affect software performance and quality but are often not so explicit. In order to understand and measure quality, scientists often built models of quality. There are several models of quality which suggest the ways various quality attributes can be interconnected. The first quality models were developed in critical application areas. A quality model is a set of characteristics and relationships between them, which provides a basis for specifying quality requirements and quality assessment (ISO/IEC 9126, 1991). There are a number of different approaches to define the set of characteristics that need to be measured to determine software quality, or define the model of quality, ranging from the classical Boehm (Boehm, 1978), McCall (McCall, et al, 1978, pp.133-139) to new approaches (ISO/IEC 25010, 2010).

## Software quality and standards

The market was setting such requests and expectations for software that only industrial production was able to meet. In order to organize industrial production, it is essential that there are standards by which they are developed. Since software development is very dynamic, official standards could not accompany it; the first standards to appear on the market were standards that represent a common agreement between groups of producers or standards of some major corporations (eg. Microsoft) that are generally accepted in the market. Such standards are called industry standards and are the basis for initiating and adopting official standards by international and national organizations concerned with standardization (eg. International Standard Organisation - ISO, the American National Standards Institute - ANSI, Federal Bureau for Standardization).

The US Department of Defense (US DoD) and the US military are the biggest users of software and among the first to identify the need and publish a series of standards in the field of security and protection of information and communication technologies. These standards are imposed on other NATO members and later adopted or adapted for civilian use. By publishing MIL-STD-498 standard in 1994, the US DoD attempted to consolidate, in a single document, all the requirements for software development, quality and documentation. In 1995, the International Organization for Standardization (ISO) published the standard (ISO/IEC 12207, 1997) which made all military standards (MIL-STD) nul and void. Although the publication of military standards and their application for civilian purposes started the wheel of history, their application in the industry was not accepted; there was an initiative to apply commercial standards wherever they had already existed.

However, the quality of software developed for military purposes is subject to more stringent requirements and criteria arising from the specific character of military organizations. Today, in the field of software, there are currently over 1,000 official standards, adopted by the international or national standards organizations, out of which more than 350 are in the field of software engineering. However, the situation in Serbia is quite different, and in relation to global trends of standardization in this area, Serbia is in considerable delay. The National Bureau of Standards has accepted the recommendations of international and European organizations for standardization and mainly takes the ISO/IEC standards to which it gives national prefaces and necessary explanations.

It should not be expected that all international standards are perfectly systematised and harmonized regarding the methodology and content, but they are undoubtedly a powerful tool which can help the improvement of design methods and ensure the quality of finished software products. However, in order to review various usability aspects, the adopted standards will be explored as well as the recommendations in the field of usability that establish the methods, principles, procedures for testing and evaluating the characteristics of software quality and provide the basis for the establishment of a methodology of evaluation and certification of software products.

Standardization in information technologies contributes to establishing more efficient information functions, their greater stability and easier transition. The application of international, national and internal standards in the development of software products creates conditions for the development of efficient, cost-effective, secure and reliable software products. By standardizing the software development process, its planning, quantifying and monitoring, documentation and continuous improvement and development, the preconditions are created for the realization of software products of defined quality. A well-documented system in accordance with the standards is easily replaceable, portable from one hardware and software platform to another and it protects investment.

Evaluation and standardization of tools used in the development of information systems create a possibility that the quality of the process and the end product is at the desired and expected level. From the perspective of complex information systems whose development and implementation involve several organizations, the application of standards not only provides the appropriate quality of the final product and process development, but it also creates opportunities for exchanging projects between individual organizations, facilitates user training and creates the conditions for joint work on projects of representatives of different organizations.

The rapid development of software in various fields conditioned the development of standards and tools for software development and management. Diversity has caused problems in software management and has caused a need to define a common framework that would allow everyone involved in software development, design and management to "speak the same language."

The standards are formulated so that they can be adapted to the needs of particular organizations, projects or specific applications. They

can be applied in cases where software is a stand-alone entity or a component of a complex system.

The current situation in software technology does not yet provide enough good and widely accepted scheme for evaluating the quality of software products. Since 1976, a lot has been done by individuals in defining the basis for software quality. Many models of McCall (McCall et al, 1977), (McCall et al, 1978, pp.133-139), Boehm (Boehm et al, 1976, pp.592-605), (Boehm et al, 1978), Eason (Eason, 1984, pp.133-143), Shackel (Shackel, 1991), Nielsen (Nielsen, 1993) and others were adopted and extended.

For a long time, reliability was the only way to measure quality. Over time, various studies submitted other models as well. Although such studies have been helpful, they have also created confusion, because they offered many aspects of quality. So, there is a need for only one quality model.

When purchasing and implementing software, there are a number of common issues that are always relevant, such as:

- Does the software has the required functions?
- How much is the software reliable?
- How effective is the software?
- Is the software easy to use?

These common questions initiated the emergence of ISO/IEC 9126 at the end of 1991. The first considerations originated from 1978, and in 1985 the development of ISO/IEC 9126 began. The adoption of this standard was an important turning point in the standardization of software quality and the method for its measurement.

Today, standards are applied in many fields. As information technology is developing rapidly in every respect and fulfill all spheres of life, standards are becoming an element that brings order and allows the continuity of progress.

The standards issued by the International Standards Organisation (ISO) in the broadest sense can be classified into two categories:

1. Standards-oriented product that determine the characteristics and functional requirements of the product (ISO 9126, 2001), (ISO 14598, 2001).

2. Standards-oriented processes which determine the way in which products need to be developed (ISO 9241, 1998), (ISO 13407, 1999).

However, the most commonly recommended standards for software rely on the concept of general quality standards and are grouped according to application areas:

- Standards of software quality,

- Standards of documentation,
- Standards of life cycles, and
- Standards of particular professions.

The application of these standards facilitates the software evaluation, quality forecasting and objective evaluation in order to make a decision on the election of a new product, a comparison with other products, evaluation of the positive and negative effects of a product or to make a decision on product improvement or replacement.

For projects where usability is extremely important, the standard CIF (Common Industry Format for Usability Test Reports) is recommended, approved by the organization ANSI/NCITS, which comprehensively and accurately elaborates the requirements and usability evaluation methods, the ways of testing software and hardware, and reporting form and content.

General standards for software have been mentioned so far, but, to define the quality of software for specific purposes, it is necessary to set appropriate standards for software in the field of so-called "professional" standards, which would apply when contracting specialized software.

### *ISO 9241-11 standard*

The International Organization for Standardization ISO accepted (1998) this standard in order to facilitate the process of contracting and assessment of fulfillment of defined characteristics. ISO 9241-11 (ISO 9241-11, 1998) standard was used for the identification of important information which must be taken into consideration for assessing the performance of usability and user satisfaction (Folmer & Bosch, 2004).

According to ISO 9241 part 11, usability is defined as "the extent to which a product can be used by specific users to achieve specific goals with efficiency, effectiveness, and satisfaction in a specified context of use" (ISO 9241-11, 1998). This definition combines two different views from the perspective of the user. One is to look at user performance that includes effectiveness and efficiency, and the other is its position that deals with the issue of satisfaction.

The definition of usability consisted of four elements: users, user goals, product and context of use. The context implies the "users, tasks and equipment and the physical and social environment in which the product is used." Usability in (ISO 9241-11, 1998) contained three measurable attributes of usability: effectiveness, efficiency and satisfaction (Laurusdottir, 2009), (Quesenbery, 2003).

Effectiveness: users achieve certain goals with accuracy and completeness. The system must be accurate and complete to contain features so that users do not have difficulty in achieving their objectives.

Efficiency: The resources of the system to be used with the accuracy and completeness through which to accomplish their goals.

Satisfaction: Customers must be satisfied using the system.

ISO 9241-11 explains the benefits of measuring usability in relation to performance and customer satisfaction, and stresses that the usability of the visual display depends on the context of use, i.e. that the level of usability achieved will depend on the particular circumstances in which the product is used.

### *ISO / IEC 9126 standard*

In the early 1990s, in software engineering, there was an attempt to consolidate many aspects of quality into a single model which would act as a global standard for measuring the quality of software. One of the main objectives of international standards is to establish consistency and compatibility in a specific field. This standard, known in literature as ISO 9126 (ISO 9126, 1991) standard, was to help in the understanding of the negotiation between the client software and the manufacturer, and to recommend to what extent and what quality characteristics software must have. As a basis for an international standard of quality software, McCall's model is recommended. ISO 9126 defines the quality of the product as a set of product characteristics (Maryoly, 2003). One of the main differences between the ISO and models made by McCall and Boehm is that the ISO model is a model of strict hierarchies.

The first version of the standard defines the quality model which had six main characteristics that contribute to the quality (functionality, reliability, usability, efficiency, maintainability and portability), with 20 subcharacteristics. Each of the characteristic refers to exactly one attribute. The characteristics are related to the user's view of software, not to an internal view of its designer. The standard recommends the direct measurement of the characteristics, but does not give details of how measurements are carried out.

This was the basic model, whose quality was to be adapted, if necessary, to a specific project. ISO/IEC 9126 did not contain attributes and metrics in it, not even in the form of a proposal, nor it prescribed methods of measurement, ranking and evaluation; it offered general guidelines for a model of a software quality evaluation process.

This model helped to clarify what is valued in the software being made and used, but none of the models included a reason why some of the features were included and not the others and why certain attributes appeared in the hierarchy. In addition, it did not contain guidance on how to compose a lower level features into those of a higher level in order to obtain a total quality assessment. All these problems have made it difficult to determine whether a given model is complete or consistent.

In the period from 2001 to 2004, the International Organization for Standardization revised earlier versions of the standard (ISO/IEC 9126, 1991) and issued its expanded version.

Since model and quality metrics are useful not only for evaluating quality, but also for defining the requirements for quality and for other purposes, it was decided that the process of evaluating software quality should be isolated in a brand new series of standards ISO/IEC 14598 (Azuma, 2001). This standard is now divided into two standards: ISO/IEC 9126 (ISO/IEC 9126, 2001) and ISO/IEC 14598. However, it turned out that this had been a bad decision because the matter got complicated and the popularity of both standards decreased.

ISO/IEC 9126 consisted of four parts: an international standard and three technical reports. The first part of ISO/IEC 9126-1 was related to concepts, ie. the preferred model of software quality. The novelty was that the quality model was twofold: it consisted of the model of internal / external quality and the model of quality in the software use. The quality model divides the quality attributes according to six characteristics: functionality, reliability, usability, efficiency, maintainability and portability. These characteristics are further divided into 27 subcharacteristics that can be measured by internal or external metrics. The models of the internal/external quality had identical characteristics and subcharacteristics. They differed only in metrics by which they were quantified, and which were defined elsewhere in the standard. For each characteristic, and subcharacteristics, the ability of the software is determined by a set of internal/external attributes that can be measured.

In the context of ISO/IEC 9126-1 standard, quality in use is how a complete system on which the software runs is seen by the end user and it is measured by the results of the software use. The attributes of internal and external quality are the cause, and the attributes of quality in use are the effect. According to Bevan (Bevan, 1999): "Quality in use is the goal, and the quality of the software product is the means by which this goal is achieved." Therefore, quality in use is a combined effect of the characteristics of internal and external quality of the end user. It can be measured by the extent to which



specific users can achieve specified tasks with effectiveness, productivity, safety and satisfaction (four features of quality in use) in the specific context of use.

However, although there are three views on quality, we should not forget that these are only different perspectives of the same thing, and that each of them has a relationship with the other two. By measuring and evaluating quality in use, external software quality can be confirmed. Furthermore, measurement and evaluation of external quality can verify internal software quality, and the examination of internal quality can result in the conclusions about making necessary improvements to the software. Similarly, taking into account the attributes of internal quality is a prerequisite for achieving the required external behavior, and considering the attributes of external quality is a prerequisite for achieving quality in use.

The second part of ISO/IEC TR 9126-2 (ISO/IEC 9126-2, 2003) applies to the external characteristics of software quality metrics. The third part of the ISO/IEC 9126-3 (ISO/IEC 9126-3, 2003) provides internal metrics to quantify the characteristics of software quality. Finally, the fourth part of ISO/IEC 9126-4 (ISO/IEC 9126-4, 2004) standard contains a basic set of metrics for each quality characteristic in use, the instructions for their application and examples of how they are used in the life cycle of the software product.

### *ISO / IEC 14598 standard*

The process of evaluating the quality of the software was originally defined by ISO/IEC 14598 (ISO/IEC 14598-1, 1999) to provide methods for measuring, assessing and evaluating the quality of a software product.

ISO/IEC 14598-1 provides the basis for this series by defining the basic concepts and describing where to use the characteristics and quality metrics defined in ISO/IEC 9126 series. ISO/IEC 14598-1 consists of several parts that appear under the general title of Information technology - Software product evaluation. Evaluating Software Quality refers to the model of quality, methods of evaluation, measurement software, and support tools. This standard is intended for designers, those who carry out procurement and independent assessors.

The series of international standards ISO/IEC 14598 provides guidelines and requirements for the evaluation process for:

- Development of new products or improvements of existing ones (ISO/IEC 14598-3, 2000)
- Procurement of products or reusing existing products (ISO/IEC 14598-4, 1999)
- Independent evaluation of the requirements of the designer, supplier, or a third party (ISO/IEC 14598-5, 1998).

Software testing is the most efficient and the best way of achieving and maintaining the quality of software products. It is based on the rules and principles of quality within the standards relating to quality and, in some way, it implements all predefined methods and techniques for creating and maintaining high-quality software.

### *SQuaRE – Series of ISO standards for software quality*

Bearing in mind the shortcomings of ISO/IEC 9126 standard, which are identified in the literature by other authors, eg. (Azuma, 2001), (Al-Qutaish, 2010, pp.205-228), (Suryn and Gil, 2005), (Olsina and Molina, 2008), the International Organization for Standardization decided to issue a new series of standards that will regulate this area, with a better organization and under a single label.

The new standard had a working name SQuaRE (Software Product Quality Requirements and Evaluation). It represents the second generation of standards for software quality and is issued with the intention to eventually replace the ISO/IEC 9126 and ISO/IEC 14598 series, and is made according to the following guidelines:

- Connecting the ISO/IEC 9126 and ISO/IEC 14598 series into one harmonized standard,
- The introduction of a new organization and standard,
- The introduction of a new reference model,
- The introduction of detailed guides,
- Introduction of standards on quality requirements,
- The introduction of a manual for series practical use with examples,
- Coordination and harmonization of the measurement model with ISO/IEC 15939 Software engineering - Software measurement process.

SQuaRE means a series of fourteen ISO/IEC standards and technical reports related to software quality, grouped into five thematic sections or parts:

1) Quality management (ISO/IEC 2500n) provides a general picture of the evaluation process and quality models, which is important for understanding the complete series of standards. It defines common models, terms and definitions to which all other standards of the SQuaRE series refer.

2) Quality Model (ISO/IEC 2501n) introduces a detailed model of quality (including features for internal, external and quality in use, which are further broken down into subcharacteristics) and instructions for practical use of the quality model.

3) Measurement of quality (ISO/IEC 2502n) comprises standards that include the reference model for measuring the quality of software products, metrics for internal, external and quality in use and the instructions for their practical use.

4) Requirements for quality (series ISO/IEC 2503n) help in specifying the quality required and a process of defining requirements is mapped to technical processes defined in ISO/IEC 15288 standard, and

5) Evaluation of quality (series ISO/IEC 2504n) provides requirements, recommendations and guidelines for software product evaluation, performed by either evaluators, customers or those who develop software.

The main differences between the series SQuaRE and ISO/IEC 9126, or ISO/IEC 14598 are:

- the introduction of a general reference model;
- the harmonization and coordination of guidelines for measuring and evaluating the quality of software;
- the introduction of a part on quality requirements;
- the existence of detailed manuals for using each part separately;
- the introduction of guidelines for practical use in a form of an example;
- harmonization of the content of this series with the terminology related to software measurement (used in ISO 15939 standard).

Nigel Bevan, an international expert in software usability, had a strong influence on this series of standards, emphasizing the importance of quality in use (Bevan et al., 1991).

### Model of quality in use in ISO 25010 standard

The ISO has just recently developed a more comprehensive definition of quality in use, which has usability, flexibility and security as subcharacteristics that can be quantified from the perspective of different stakeholders, including users, managers and those who maintain the software. ISO/IEC 25010 maintains the previously adopted three views on quality: internal, external quality and quality in use. However, it extends the concept of product quality from six (ISO/IEC 9126-1) to eight characteristics (Figure 1).

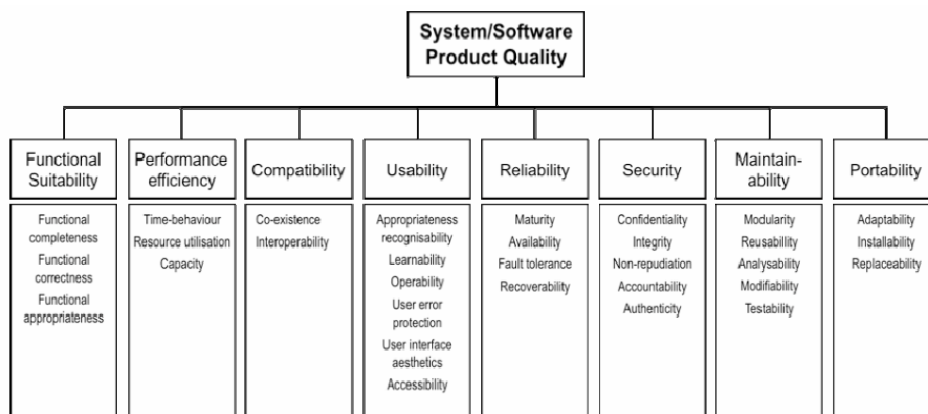


Figure 1 – Product quality model (Figure 4 in ISO/IEC 25010, 2010)  
 Рис. 1 – Модель качества продукции (Рис. 4 в ISO/IEC 25010, 2010)  
 Слика 1 – Модел квалитета производа (Слика 4 у ISO/IEC 25010, 2010)

Compared with ISO/IEC 9126-1, the usability feature was renamed operability and has a broader meaning. Some subcharacteristics, such as suitability for learning, remained while new ones were added (such as technical accessibility, compliance and the ability of applications to be helpful to users). Compatibility was added as a new feature, while Safety was singled out as a special feature instead of being a subcharacteristic of the Functionality characteristic in the previous standard: other names were slightly changed to be more descriptive.

The second model in ISO 25010 standard relates to the quality in use (Figure 2) and includes the earlier characteristics of quality in use from ISO 9126-1 standard, with some new ones added.



Figure 2 – Model of quality in use (Figure 3 in ISO/IEC 25010, 2010)  
 Рис. 2 – Модель качества в потреблении (Рис.3 в ISO/IEC 25010, 2010)  
 Слика 2 – Модел квалитета у употреби (Слика 3 у ISO/IEC 25010, 2010)

The characteristics of effectiveness and satisfaction of ISO/IEC 9126-1 were inserted into the new standard as the subcharacteristics of usability in use, while productivity was renamed efficiency in use. Effectiveness in use is the ability of the software product to enable users to achieve goals with accuracy and completeness in the context of a specific use. Satisfaction in use is the ability of the software product to satisfy users in a specific context of use. Efficiency in use is the ability of the software product to enable users to use the appropriate amount of resources in relation to the effectiveness achieved in a specified context of use. In addition, flexibility in use was added to address different contexts of use. Safety in use is the ability of the software product to achieve acceptable levels of risk of damage to people, business, software, property or the environment in the context of a specific use.

Traditionally, in accordance with the first ISO definition of usability in (ISO/IEC 9126, 1991), usability is seen through the attributes of the user interface that make the product easy to use, as an attribute of software quality made by a set of characteristics that contribute to the ability of the software product to be user-friendly, versatile and attractive for the user, when used under specified conditions. However, this definition of usability of the user interface is inconsistent with the definition of usability defined in (ISO 9241-11, 1998), which looks at usability from an ergonomic point of view, as a degree to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

This broader interpretation of usability is built into the changed ISO 9126-1, 2001, named "quality in use" as a quality from the user's perspective when using the product (Bevan, 1999). When the ISO/IEC 9126-1 quality model was built in series SQuaRE (such as ISO/IEC 25010), some ISO/IEC national bodies commented on the discrepancy between the narrow definition of usability inherited from ISO/IEC 9126 and the general definitions of the CIF. In order to harmonize the definition of usability SQuaRE with the CIF, the characteristic of usability was renamed to operability and it has a broader meaning. This made it possible to define usability as a characteristic of quality in use, with the subcharacteristics of effectiveness, efficiency and satisfaction. The quality in use in ISO/IEC CD 25010.3 has two characteristics: Safety of 9126-1, as well as a new feature: Flexibility.

### The quality of the life cycle

The connection between process quality and product quality is obvious. Process quality affects the quality of the product while the quality of the product affects quality in use, and vice versa. The internal attributes influence the external attributes, and the external attributes influence quality in use. It is obvious that the quality of intermediary products is measured by internal characteristics, and that the quality of the final product reflects in the external attributes and the product usage value. The relationship between process quality, product quality and quality in use in ISO 9126-1/25010 is shown in Figure 3.

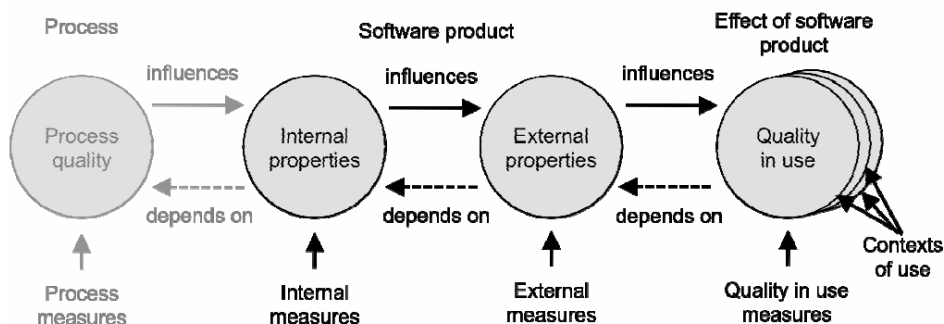


Figure 3 – Quality in the lifecycle (Figure C.2 in ISO/IEC 25010, 2010)  
 Рис. 3 – Качество в жизненном цикле (Рис. 2 в ISO/IEC 25010, 2010)  
 Слика 3 – Квалитет у животном циклусу (Слика С.2 у ISO/IEC 25010, 2010)

Therefore, the requirements relating to the quality of the software product include the assessment criteria for internal quality, external quality and quality in use to meet the needs of designers, those who maintain the product, purchasers and users. (ISO/IEC 14598-1, 1999) The quality in use of the system is characterized by the impact that software has on the interested parties. It is determined by the quality of software, hardware and operational environment and by the characteristics of users, tasks and social environment. All of these factors contribute to the quality in use of the system.

The features of the software product determine the quality of products in specific contexts of use (Table 2 in ISO 25010). The characteristics of product quality such as functional suitability, efficiency of performance, usability, reliability and safety will have a significant impact on the quality in use for basic users. Compatibility, maintainability and portability will have a significant impact on quality in use for people who maintain the system.

The properties of quality are the natural characteristics of software that contribute to quality and they can be classified into one or more characteristics or subcharacteristics. The quality properties can be measured using some measurement methods that represent a logical sequence of operations for quantifying a characteristic on a particular measurement scale.

Software quality is usually assessed by measuring the static parameters of the intermediate product (internal attributes), and by measuring the code behavior while running (external attributes) or by measuring the attributes of quality in use. The basis for measuring the quality is provided by the framework defined back in 1999 in ISO/IEC 14598-1.

## Software Quality Metrics

Although it might seem that the area of quality metrics is defined, and that the application of metrics for monitoring and evaluating the quality and usability is clear enough, there is a lot of uncertainty. The formalization of the evaluation area is missing as well as the formalization of the measurement of various quality indicators.

To measure means to assign, according to certain rules, a certain number, then usually called a value, to a certain object.

Measurement means determining how many times the measuring quantity (usually referred to as a unit of measurement) is contained in the value to be measured. The value measurement result is a number, and it is necessary to make a good distinction between the value and the number obtained by its measurement. This number, together with the appointment of a value unit, is called a number of a unit of measure or a numerical value of a quantity.

Measurement is the process of determining the quantity, size or degree (of something) using a standard set of measures and the measurement procedure defined by metrics, while metrics defines a measurement system or standard, scales and units of measurement for monitoring efficiency indicators (Ted, Angelika, 2006).

Quality in use is the quality of the software product from the user's point of view when used in a specific environment and within the context of a specific use. It measures the degree to which users can achieve their goals in a particular environment, and does not measure the properties of the software itself. To meet the criteria for external quality, it is not enough to meet the criteria for internal quality. Also, meeting the criteria for external quality usually is not enough to meet the criteria for quality in use.

When quality is discussed, it usually considers meeting the demands for quality from the user's point of view. Then the user requirements for quality refer to the quality of the product in use. In order to incorporate meeting the demands for quality into the software development process, it is necessary to evaluate the software product in all phases of its life cycle.

Quality is assessed by measuring the internal (usually the static measurement of intermediate products) and external (measuring the behavior of the code while being executed) attributes or by measuring the attributes of quality in use.

To ensure the required quality, it is necessary to measure and evaluate many features that allow determining software quality, where software quality metrics that will be used to evaluate each characteristic plays an important role. To measure the quality of software, according to the ISO quality model, internal, external and quality in use metrics are used.

Internal quality of the software product is measured and evaluated based on the requirements of internal quality. The quality details can be improved during the implementation of the code and tests, but the basic quality of the software product represented by internal quality remains unchanged, except in the case of re-design.



Internal metrics is a quantitative scale and a method of measurement which can be used to measure attributes or quality characteristics of the software that is a part of the software, directly or indirectly (not the result of the measurement of the system behavior). Internal metrics are useful during design or coding, in the early stages of the software life cycle. Internal metrics can be applied to software products that are not executed (such as specifications or source code) during designing and coding. During the design of software products, intermediate products can be measured using internal metrics that measure internal properties. The main purpose of internal metrics is to ensure achieving the required external quality and quality in use. Internal metrics provide assistance to users, people who perform the evaluation, people who perform the testing and designers, enabling evaluation of the quality of the software product and show the quality elements long before the software product becomes executable. Internal metrics measure the internal attributes or indicate the external attributes by analysing the static properties of intermediate products or the software product being provided. The measurements of internal metrics use the frequency of the composition of software elements which appear, for example, in the source code commands, control charts, data flow, and displays of state changes.

External quality is the quality of execution of software, which is usually measured and evaluated during testing in a simulated environment with simulated data using external metrics. During testing, there is a need to detect and eliminate most errors, but there may be some errors left after tests. Since it is difficult to correct the basic software architecture or the basic settings of software development, the development bases do not change during testing.

External metric is a quantitative scale and a method of measurement, which can be used to measure the attributes or the characteristics of the software that is the result of the behavior of a system whose part the software is. External metrics are applicable to the software that runs during testing or use in later stages of development, or even during use in real working environment. External metrics use software product measurements derived from the measurements of the behavior of the system of which it is a part, by testing, running and monitoring the software being run or the system. Before ordering or using the software product, it should be evaluated by using the metrics based on business objects linked with the use, operation and management of the product in the specified

organizational and technical environment. External metrics provide assistance to users, people who perform the evaluation, people who perform the testing as well as designers and enable them to evaluate the quality of the software product during testing or execution.

Quality in use metrics relate to the extent to which the product meets the needs of users to achieve specified goals with effectiveness, productivity, safety and satisfaction in the context of use. Evaluating quality in use validates software product quality in user - task scenarios.

The link between quality in use and other quality characteristics of software products depends on the type of the user:

- for the end user, quality in use is mainly a result of functionality, reliability, usability and efficiency;
- the persons responsible for the implementation of software, quality in use is mainly a result of maintainability;
- for persons who carry the software, quality in use is mainly a result of portability.

## Conclusion

The concept of software quality has many dimensions and cannot be easily defined. To ensure the required quality, it is necessary to monitor various parameters, make plans and develop and implement standards and quality system documentation, which will be applied to software products.

Software quality has significance both for manufacturers and users of software and varies depending on the viewing angle. To get a software product that has been developed in accordance with the specification that meets customer requirements and does not contain errors, software quality is determined by measuring multiple parameters. In doing so, it is necessary to choose measurement parameters measure and provide appropriate testing methodologies and techniques. A particular problem is the definition of metrics, i.e. the type of measurement that is correlated to the software system.

Therefore, it is important that software development takes place in accordance with prescribed standards in order to avoid subsequent interventions - software modifications in the later phases of the software life cycle.

Of course, there are different aspects of quality and different metrics related to the phases of the product life cycle. A defined quality model is

used for the evaluation of the quality of software products in the phase of defining goals for the quality of products or intermediate products. Evaluation of attributes may be carried out by measuring the consequences or by direct measurement. In order to create the list of parameters related to quality, the process of hierarchical decomposition of the software product is used. Of course, it is not possible to measure all characteristics in all possible cases.

By measuring and evaluating quality in use, software external quality can be confirmed. Furthermore, measurement and evaluation of external quality can verify internal software quality, and the examination of internal quality can lead to the conclusions about the necessary improvements of the software production process. Similarly, taking into account the attributes of internal quality is a prerequisite for achieving the required external behavior, and considering the attributes of external quality is a prerequisite for achieving quality in use. Therefore, quality in use is a combined effect of the characteristics of internal and external quality on the end user. The attributes of internal and external quality are the cause, and the attributes of quality in use are the effect. From the abovementioned, Bevan (Bevan, 1999) draws the conclusion: "Quality in use is the goal, and the quality of the software product is the means by which this goal is achieved."

According to international standards (ISO/IEC 9126-1, ISO/IEC 25010), quality in use is how the end user sees a complete system on which the software runs and it is measured by the results of software usage.

The existing models of quality described in the current standards are intended to describe the quality of traditional software products and are suitable for the evaluation of the quality of traditional software products as well as for the identification of usability problems of traditional graphic user interfaces. However, the characteristics of software quality are not enough to describe the quality of a broader set of specific applications based on Web technology.

Models of quality in these standards provide a range of quality characteristics relevant to a wide range of interested parties, such as developers, system integrators, owners, maintenance teams, contractors, experts on security and quality control and users. However, a complete set of quality characteristics of these models may not be relevant to all types of users. Thus, for each type of users the importance of quality characteristics in each model should be taken into account prior to the finalization of the sets of quality characteristics to be used. The relative importance of quality characteristics will depend on high objectives and

project tasks. Therefore, the model should be tailored (adjusted) before use as a part of the decomposition of requirements for the identification of the most important characteristics and subcharacteristics, in accordance with the objectives and tasks of the parties concerned.

Today, there are a number of methods for evaluating software quality. Choosing the right method is not an easy task, since it depends not only on the software product type, but also on the objectives of the development project and the context of use. In fact, the choice of a method depends on various criteria, the most important ones being resources required to perform the method (time, money, the number of evaluators and their expertise, the number of users for testing, place and test equipment), the required level of objectivity and the possibility of applying it in different stages of the application development.

### *References*

Al-Qutaish, R. 2010. ISO 9126: Analysis of Quality Models and Measures. In: A. Abrain Ed., Software Metrics and Software Metrology. John Wiley & Sons, Inc., pp.205-228. ch. 10.

Azuma, M. 2001. SquaRE: The next generation of the ISO/IEC 9126 and 14598 international standards series on software product quality.

Bevan, N., Kirakowski, J., & Maissel, J. 1991. What is Usability?. In: Proceedings of the 4th International Conference on HCI. Stuttgart.

Bevan, N. 1999. Common Industry format usability tests. Serco usability services 4 sandy lane, Teddington, Middx, TW11 0 DU, UK.

Boehm, B.W., Brown, J.R., & Lipow, M. 1976. Quantitative Evaluation of Software Quality. In: Proceedings of the 2nd International Conference on Software Engineering. , pp.592-605.

Boehm, B.W., & et. all., 1978. Characteristics of Software Quality. UTRW series on Software Technologies. North Holland. Vol. 1.

Eason, 1984. Towards the experimental study of usability. Behaviour and Information Technology, 3(2), pp.133-143.

ISO 9241-11, 1998, Ergonomic requirements for office work with visual display terminals (VDTs). Part 11: Guidance on usability.

ISO 13407, 1999, Human Centred Design Process for Interactive Systems.

ISO/IEC 9126-1, 2001, Software Engineering - Product Quality: Part 1: Quality Model.

ISO/IEC TR 9126-2, 2003, Software Engineering - Product Quality. Part 2: External Metrics.

ISO/IEC TR 9126-3, 2003, Software Engineering - Product Quality. Part 3: Internal Metrics.

ISO/IEC TR 9126-4, 2004, Software Engineering - Product Quality. Part 4: Quality in Use Metrics.

ISO/IEC 14598-1, 1999, Information Technology - Software Product Evaluation. Part 1: General Overview.

ISO/IEC 14598-3, 2000, Information Technology - Software Product Evaluation. Part 3: Process for Developers.

ISO/IEC 14598-4, 1999, Information Technology - Software Product Evaluation. Part 4: Process for Acquirers.

ISO/IEC 14598-5, 1998, Information Technology - Software Product Evaluation. Part 5: Process for Evaluators.

ISO/IEC 25010.3, 2010, Systems and software engineering Software product Quality Requirements and Evaluation (SQuaRE) – Software product quality and system quality in use models.

JUS ISO/IEC 12207, 1997, Informaciona tehnologija - procesi žiivotnog ciklusa softvera.

Laurusdottir, M.K. 2009. Listen to your users, The Effect of Usability Evaluation on Software Development Practice. Uppsala University, Department of information technology.

McCall, J., Richards, P., & Walters, G. 1977. Factors in Software Quality. NTIS AD-A049-014.

McCall, J.A., & Cavano, J.P. 1978. A Framework for the Measurement of Software Quality. In: Proceedings of the ACM Software Quality Assurance Workshop. , pp.133-139.

Olsina, L., & Molina, H. 2008. How To Measure And Evaluate Web Applications In A Consistent Way. U G. Rossi & et al. Ur., Web Engineering: Modelling and Implementing Web Applications. London: Springer. pp.385-420.

Quesenbery, W. 2003. Dimensions of usability. U M. Albers & B. Mazur Ur., Content and complexity: Information design in technical communication. Mahwah, NJ: Lawrence Erlbaum Associates.

Suryan, W., & Gil, B. 2005. ISO/IEC 9126-3 internal quality measures: Are they still useful.

Ted, H., & Angelika, P. 2006. Measuring the effectiveness of your ISMS implementations based on ISO/IEC 2700. BSI.

---

## СТАНДАРТЫ КАЧЕСТВА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Небойша Д. Джорджевич

Вооруженные Силы Республики Сербия, Управление сухопутных войск, г. Ниш, Республика Сербия

ОБЛАСТЬ: вычислительная техника и информатика, информационные технологии

ВИД СТАТЬИ: профессиональная статья

ЯЗЫК СТАТЬИ: английский

### Резюме:

*Для достижения желаемого качества программного обеспечения, в первую очередь необходимо изучить существующие стандарты в данной области, и соблюдать их на всех этапах жизненного цикла программного обеспечения. В данной статье подчеркнута значимость применения стандартов в области качества программного обеспечения. Международная организация по стандартам за последних 20 лет сыграла важную роль в развитии многих стандартов качества программного обеспечения. В первой части статьи представлен краткий обзор эволюции стандартов в области качества программного обеспечения, а затем подробно описана действующая серия, под названием SQuaRE, являющаяся неотъемлемой частью ISO/IEC стандартов. Данная область очень важна, так как знание стандартов качества программного обеспечения и их применение, непосредственно влияют на развитие качества программного обеспечения, тем самым обеспечивая ему широкое применение на протяжении длительного периода времени. Применение описанных стандартов и эволюционный подход к квантификации характеристик качества программного обеспечения зависит от доступности ресурсов для осуществления необходимых методов (время, деньги, необходимое количество специалистов-тестировщиков, необходимое количество пользователей-тестировщиков, место и оборудование для тестирования) и уровня объективности, а также возможности применения на различных этапах разработки приложений.*

Ключевые слова: *ISO, стандарт качества, качество программного обеспечения.*

---

## СТАНДАРДИ КВАЛИТЕТА СОФТВЕРА

Небојша Д. Ђорђевић

Војска Србије, Команда Копнене војске, Ниш, Република Србија

ОБЛАСТ: рачунарство и информатика, информационе технологије

ВРСТА ЧЛАНКА: стручни чланак

ЈЕЗИК ЧЛАНКА: енглески

**Сажетак:**

За достизање жељеног квалитета софтвера најпре је потребно поседовати информације о постојећим стандардима из ове области, а затим их поштовати у свим фазама животног циклуса софтвера. У раду је наглашен значај примене стандарда из области квалитета софтвера. Међународна организација за стандарде је у последњих 20 година одиграла важну улогу у развоју више стандарда квалитета софтверских производа. На почетку рада приказана је еволуција стандарда из области квалитета софтвера, а затим је детаљно описана актуелна серија ISO/IEC стандарда под називом SQuaRE. Ова област је веома значајна, јер познавање стандарда квалитета софтвера и њихова примена доприноси развоју квалитетног софтвера која ће му обезбедити широку употребу и трајање. Примена описаних стандарда и евалуационих приступа за квантификацију особина квалитета софтвера зависи од расположивих ресурса за извођење метода (време, новац, број потребних евалуатора и њихова стручност, број корисника за тестирање, место и опрема за тестирање), потребног нивоа објективности, те могућности примене у разним фазама израде апликације.

Кључне речи: *ISO, стандард квалитета, квалитет софтвера.*

Paper received on / Дата получения работы / Датум пријема чланка: 06. 04. 2016.  
 Manuscript corrections submitted on / Дата получения исправленной версии работы /  
 Датум достављања исправки рукописа: 01. 10. 2016.  
 Paper accepted for publishing on / Дата окончательного согласования работы / Датум  
 коначног прихватања чланка за објављивање: 03. 10. 2016.

© 2017 The Author. Published by Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, втг.мо.упр.срб). This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2017 Автор. Опубликовано в «Военно-технический вестник / Vojnotehnički glasnik / Military Technical Courier» (www.vtg.mod.gov.rs, втг.мо.упр.срб). Данная статья в открытом доступе и распространяется в соответствии с лицензией «Creative Commons» (<http://creativecommons.org/licenses/by/3.0/rs/>).

© 2017 Аутор. Објавио Војнотехнички гласник / Vojnotehnički glasnik / Military Technical Courier (www.vtg.mod.gov.rs, втг.мо.упр.срб). Ово је чланак отвореног приступа и дистрибуира се у складу са Creative Commons licencom (<http://creativecommons.org/licenses/by/3.0/rs/>).

