Quality and usability: A new framework

Nigel Bevan
Usability Services
National Physical Laboratory
Teddington, Middlesex, TW11 0LW, UK
Nigel.Bevan@npl.co.uk

Abstract

ISO/IEC 9126 (1991) established a practical way of decomposing software quality into a set of characteristics and subcharacteristics. Reconciling this approach to quality with a new standard for usability (ISO 9241-11) has led to a comprehensive framework for software product quality which is being incorporated in a revision to ISO/IEC 9126. The new framework defines three perspectives: internal quality (static properties of the code), external quality (behaviour of the software when it is executed) and quality in use (whether the software meets the needs of the user when it is in use). Quality in use is a broader view of the concept of usability defined in ISO 9241-11. ISO/IEC 14598 describes a process for evaluating software product quality which is consistent with this model.

Software quality characteristics: ISO/IEC 9126

In order to evaluate software it is necessary to select relevant quality characteristics. This can be done using a quality model which breaks software quality down into different characteristics. ISO/IEC 9126 (1991) provides a general-purpose model which defines six broad categories of software quality: functionality, reliability, usability, efficiency, maintainability and portability. These are further broken down into subcharacteristics which have measurable attributes (figure 1).

The ISO/IEC 9126 characteristics and subcharacteristics provide a useful checklist of issues related to quality. The actual characteristics and subcharacteristics which are relevant in any particular situation will depend on the purpose of the evaluation, and should be identified by a quality requirements study.
The ISO/IEC 9126 (1991) definitions of the software quality characteristics are shown in figure 2.

**functionality**: a set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.

**reliability**: a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.

**usability**: a set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users.

**efficiency**: a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.

**maintainability**: set of attributes that bear on the effort needed to make specified modifications.

**portability**: a set of attributes that bear on the ability of software to be transferred from one environment to another.
These definitions follow the style of the definition of quality in ISO 8402 (Quality vocabulary):

**quality**: the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs.

This is a “product” oriented view of quality (Garvin, 1984): “an inherent characteristic of the product determined by the presence or absence of measurable product attributes”. In this view, the quality of a software product can be specified and built in as specific attributes of the code. The ISO/IEC 9126 definitions acknowledge that the objective of these attributes is to meet user needs in the form of functionality, reliability, usability, efficiency, maintainability and portability. But ISO 8402 makes it clear that a product-oriented view of quality should not be confused with measures of the “degree of excellence” resulting from the presence of absence of required attributes. Yet the objective of quality from the user’s perspective is to achieve a degree of excellence in a particular context of use. Despite the apparent user orientation of ISO/IEC 9126, the definitions in terms of attributes imply that software quality should be specified and measured on the basis of attributes of the source code.

**Definitions of usability**

This approach to quality is particularly inappropriate for specifying and evaluating usability (as “a set of attributes that bear on the effort needed for use ...”). Although developers would like to know what attributes to incorporate in the code to reduce the effort required for use, presence or absence of predefined attributes cannot assure usability, as it is usually impossible to know how users will respond until they have actually experienced use of a prototype system.

This is why when defining usability in ISO 9241-11, the ISO software ergonomics committee took an approach to usability based on the degree of excellence of a product:

**usability**: the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

ISO 9241-11 explains how usability can be measured in terms of the degree of excellence in use: effectiveness (the extent to which the intended goals of use are achieved), efficiency (the resources that have to be expended to achieve the intended goals), and satisfaction (the extent to which the user finds the use of the product acceptable). ISO 9241-11 also emphasises that usability is dependent on the context of use and that the level of usability achieved will depend on the specific circumstances in which a product is used. The context of use consists of the users, tasks, equipment (hardware, software and materials), and the physical and social environments which may influence the usability of a product in a work system. Measures of user performance and satisfaction thus assess the overall work system, and, when a product is the focus of concern, these measures provide information about the usability of that product in the particular context of use provided by the rest of the work system.

It is important to note that while this definition provides a practical way to measure usability, it is also measuring the consequences of other software quality characteristics such as the functionality, reliability and the efficiency of the computer system. Changes in these characteristics, or other components of the work system, such as the amount of user training, or improvement of the lighting, can also also have an impact on user performance and satisfaction. For this reason, early drafts of ISO 9241-11 also defined a broader concept:

**quality of use**: the extent to which specified goals can be achieved with effectiveness, efficiency and satisfaction in a specified work system.
However, this was removed from later drafts, as an unnecessary complication. The concept of the *quality of a product in use* (Bevan 1995a, 1995b) did, however, provide the link between the ISO 9241-11 and ISO/IEC 9126 views of usability, and “quality in use” was incorporated as a high level quality objective into the revision to ISO/IEC 9126-1, and the related ISO/IEC 14598-1 standard (Software product evaluation - General guide):

**quality in use**: the extent to which a product used by specified users meets their needs to achieve specified goals with effectiveness, productivity and satisfaction in a specified context of use

The revised ISO/IEC CD 9126-1 now distinguishes three broad approaches to improving the quality of a product (figure 3):

- Set criteria for process quality: attributes of the software development processes, e.g. by application of ISO 9001, or ISO 15504 (SPICE).
- Set criteria for product quality: attributes of the software (internal measures) or the behaviour of the software when tested (external quality).
- Set criteria for quality in use: the extent to which the code meets user needs for effectiveness, productivity and satisfaction in use.

Software product quality can be measured internally (typically by static measures of the code), or externally (typically by measuring the behaviour of the code when executed). The objective is for the product to have the required effect in a particular context of use. Quality in use is the user’s view of quality. Achieving quality in use is dependent on meeting criteria for external measures of the relevant quality sub-characteristics, which in turn is dependent on achieving related criteria for the associated internal measures (Figure 4).
Measures are normally required at all three levels, as meeting criteria for internal measures is not usually sufficient to ensure achievement of criteria for external measures, and meeting criteria for external measures of sub-characteristics is not usually sufficient to ensure achieving criteria for quality in use.

The software quality characteristics in the revision of ISO/IEC 9126 have been redefined in terms of “the capability of the software”, to enable them to be interpreted as either an internal or an external perspective (figure 5). The definitions also refer to “use under specified conditions” to make it clear that quality is not an absolute property, but depends on the context of use.

- **Functionality:** the capability of the software to provide functions which meet stated and implied needs when the software is used under specified conditions.
- **Reliability:** the capability of the software to maintain its level of performance when used under specified conditions.
- **Usability:** the capability of the software to be understood, learned, used and liked by the user, when used under specified conditions.
- **Efficiency:** the capability of the software to provide the required performance, relative to the amount of resources used, under stated conditions.
- **Maintainability:** the capability of the software to be modified. Modifications may include corrections, improvements or adaptation of the software to changes in environment, and in requirements and functional specifications.
- **Portability:** the capability of software to be transferred from one environment to another.

Figure 5. ISO/IEC CD 9126-1 definitions

**Measuring software product quality**

The levels of certain internal attributes have been found to influence the levels of some external measures, so that there is both an external aspect and an internal aspect to most characteristics. For example, reliability may be measured externally by observing the number of failures in a given period of execution time during a trial of the software, and internally by inspecting the detailed specifications and source code to assess the level of fault tolerance.

In the same way, external properties (such as suitability, accuracy, fault tolerance or time behaviour) will influence the observed quality. A failure in quality in use (e.g. the user cannot complete the task) could be traced to external quality (e.g. suitability or operability) and the associated internal attributes which have to be changed.

Internal metrics can be applied to a non executable software product (such as a specification or source code) during designing and coding. When developing a software product the intermediate products should be evaluated using internal metrics which measure intrinsic properties which can be derived from simulated behaviour. The primary purpose of these internal metrics is to ensure that the required external quality is achieved. Internal metrics provide users, evaluators, testers, and developers with the benefit that they are able to evaluate software product quality and address quality issues early before the software product becomes executable.

External metrics use measures of a software product derived from measures of the behaviour of the system of which it is a part, by testing, operating and observing the
executable software or system. Before acquiring or using a software product it should be evaluated using metrics based on business objectives related to the use, exploitation and management of the product in a specified organisational and technical environment. External metrics provide users, evaluators, testers, and developers with the benefit that they are able to evaluate software product quality during testing or operation.

**Quality in use metrics**

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

Quality in use is the user’s view of the quality of a system containing software, and is measured in terms of the result of using the software, rather than properties of the software itself. Quality in use is the combined effect of the software quality characteristics for the user.

The relationship of quality in use to the other software quality characteristics depends on the type of user:

- the end user for whom quality in use is mainly a result of functionality, reliability, usability and efficiency;
- the person maintaining the software for whom quality in use is a result of maintainability;
- the person porting the software for whom quality in use is a result of portability.

ISO 9241-11 explains how quality in use can be measured in terms of user performance and satisfaction: by the extent to which the intended goals of use are achieved, the resources that have to be expended to achieve the intended goals, and the extent to which the user finds the use of the product acceptable. Measures of user performance and satisfaction assess the quality in use of a product in the particular context of use provided by the rest of the working environment.

In order to specify or measure quality in use it is necessary to decompose effectiveness, efficiency and satisfaction and the components of the context of use into sub-components with measurable and verifiable attributes.

Measures of effectiveness relate the goals or sub-goals of the user to the accuracy and completeness with which these goals are achieved.

Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources. Relevant resources may include mental or physical effort, time, materials or financial cost. For example, human efficiency can be measured as effectiveness divided by human effort, temporal efficiency as effectiveness divided by time, and economic efficiency as effectiveness divided by cost.

Measures of satisfaction describe the comfort and acceptability of the use.

Care should be taken in generalising the results of any measurement of quality in use to another context which may have significantly different types of users, tasks or environments. The specification or measurement of the quality in use of a particular product should identify the characteristics of the users, the users’ goals, the relevant context of use, including the tasks and resources involved, and the measures of effectiveness, efficiency and satisfaction which are chosen as being relevant to the goals which have been identified.
Further information

The ISO standards described in this paper can be used to provide a broad and coherent approach to specifying and measuring software product quality: The relevant standards (including those under development) are listed in figure 6. Further information on these standards can be found in Bevan and Azuma, 1997.

ISO/IEC 9126 Software quality characteristics
- ISO/IEC CD 9126-1 Quality characteristics and sub-characteristics
- ISO/IEC PDTR 9126-2: External metrics
- ISO/IEC PDTR 9126-3: Internal metrics

ISO/IEC 14598 Evaluation of Software Products
- ISO/IEC FCD 14598-2: Planning and management
- ISO/IEC FCD 14598-3: Process for developers
- ISO/IEC FCD 14598-4: Process for acquirers
- ISO/IEC CD 14598-6: Documentation of evaluation modules

ISO 9241 Ergonomic requirements for office work with visual display terminals


key: DIS: Draft International Standard, FCD: Final CD, CD: Committee Draft
PDTR: Proposed Draft Technical Report

Figure 6. Relevant standards

Methods for measuring quality in use based on ISO 9241-11 have been developed by the European ESPRIT MUSiC project (Bevan and Macleod, 1994, Macleod et al, 1997). ISO DIS 13407 provides guidance on achieving quality in use by incorporating user centred design activities throughout the life cycle of interactive computer-based systems. The European INUSE consortium (INUSE, 1997) has documented user centred design methods which can be used to achieve quality in use (Daly-Jones et al 1997).

Acknowledgements

This paper includes text and diagrams adapted from ISO/IEC CD 9126-1. I would particularly like to acknowledge the contribution of Professor Azuma, general editor of ISO/IEC 9126-1, and all the other participants in ISO/IEC JTC1/SC7/WG6 and ISO TC159/SC4/WG5/SG2 who contributed to the drafts of the standards mentioned in this paper.

References


7. INUSE (1997) see http://www.npl.co.uk/inuse


9. ISO 9001 (1994) Quality systems - Model for quality assurance in design, development, production, installation and servicing


15. ISO/IEC PDTR 15504 (1997) Software process assessment