Journal of System and Software, 1999 (in press)

# Quality in Use:
# Meeting User Needs for Quality

*Nigel Bevan*
Serco Usability Services

## Abstract

There is an increasing demand for software that matches real user needs in a working environment. The paper describes the new framework for software product quality developed for ISO/IEC 9126-1: internal quality (static properties of the code), external quality (behaviour of the software when it is executed) and quality in use (the extent to which the software meets the needs of the user). Quality in use is a broader view of the ergonomic concept of usability in ISO 9241-11. Achieving quality in use requires a user-centred design process which has cultural, strategic and technical implications.

## 1. Introduction

There are increasing expectations for quality, both in the consumer and professional markets. It is no longer sufficient to just deliver products which have technical excellence - products also need to be easy to use and to fit in with the work practices and activities of the consumer and professional user. How is this quality objective to be achieved? Traditional approaches to quality put emphasis on meeting the specified requirements which are primarily functional. Attempts have been made to broaden the perception of quality, for example in ISO/IEC 9126 which categorises quality from a user perspective as functionality, reliability, usability, efficiency, maintainability and portability (Figure 1).

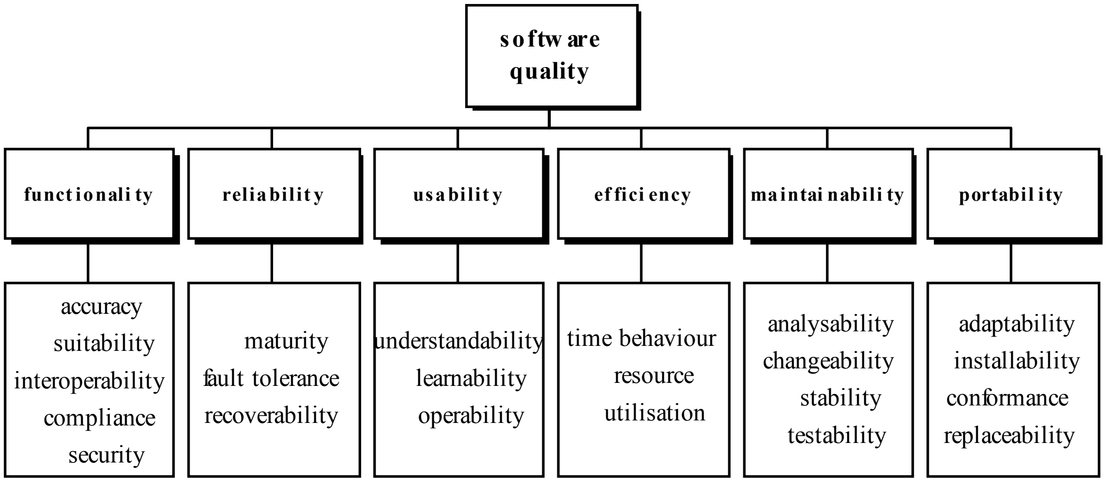

Fig. 1 Software quality characteristics

The ISO/IEC 9126 view was derived from the ISO 8402 (Quality vocabulary) definition of quality:

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

The ISO/IEC 9126 definitions acknowledge that the objective is to meet user needs. But ISO 8402 makes it clear that quality is determined by the presence or absence of the attributes, with the implication that these are specific attributes which can be designed into the product. For software, they would thus be attributes of the source code. When combined with an ISO 9001 compliant quality process, the most natural interpretation is that quality should be specified and evaluated at the level of source code attributes.

ISO 8402 distinguishes this view of quality from measures of the "degree of excellence" resulting from the presence or absence of required attributes. Yet the objective of quality from the user's perspective is for the software to exhibit excellence in the actual conditions of use. To what extent can source code attributes provide confidence that the necessary degree of excellence will be achieved?

## 2. Definitions of usability

This is a particular problem for usability, which is defined in ISO 9126 as:

> **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.

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 there is no reliable way to predict the behaviour of the users of the final product.

To solve this problem ISO/IEC 9126 has been revised to include a new quality model which distinguishes between three different approaches to product quality (Figure 2):

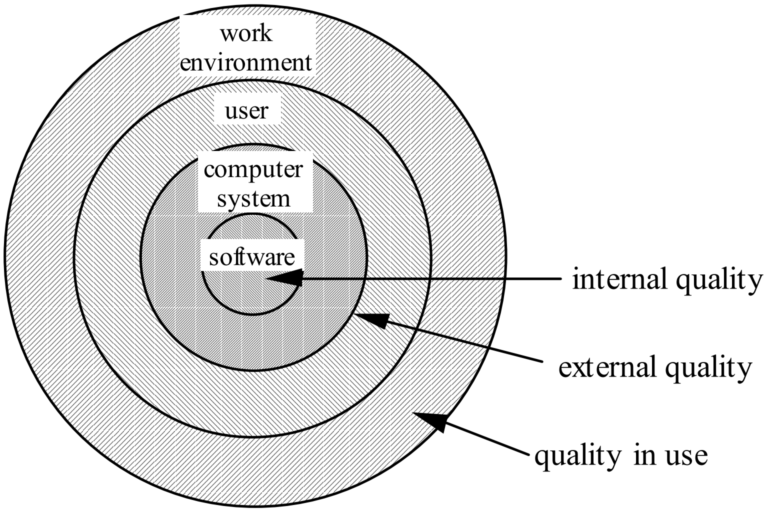

Fig 2 Relationship between internal quality, external quality and quality in use

- internal quality, which is measured by the static properties of the code, typically by inspection (such as path length)
- external quality, which is measured by the dynamic properties of the code when executed (such as response time)
- quality in use, which is measured by the extent to which the software meets the needs of the user in the working environment (such as productivity).

External quality is a result of the combined behaviour of the software and the computer system, while quality in use is the effectiveness, productivity and satisfaction of the user when carrying out representative tasks in a realistic working environment (Bevan, 1995a; 1995b). External measures can be used to validate the internal quality of the software. Quality in use measures the degree of excellence, and can be used to validate the extent to which the software meets user needs. Appropriate internal attributes of the software are a pre-requisite for achieving the required external behaviour, and appropriate external behaviour is a pre-requisite for achieving quality in use (Figure 3).

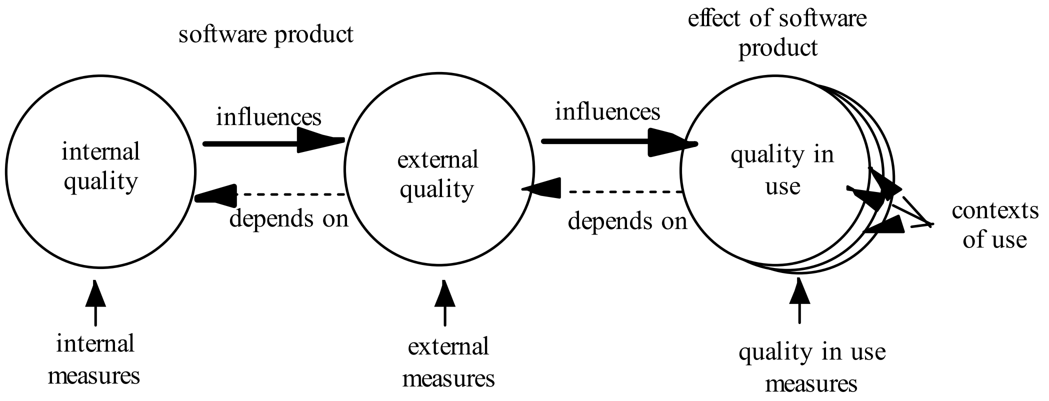

Fig. 3. Approaches to software product quality

## 3. Measuring software product quality

Software quality can be measured internally (by static measures of the code), or externally (by measuring the behaviour of the code when executed). For example, reliability can 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.

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. External properties (such as suitability, accuracy, fault tolerance or time behaviour) will influence the observed quality in use. 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.

Achieving quality in use is thus dependent on meeting criteria for external measures of the behaviour of the software, which in turn is dependent on achieving related criteria for the associated internal measures (Figure 3).

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 is not usually sufficient to ensure achieving criteria for quality in use.

### *3.1 Software metrics*

When developing a software product, the intermediate products should be evaluated using internal metrics which measure intrinsic properties which can be derived from inspecting the code. 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.

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 4). 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.

Fig. 4. ISO/IEC FCD 9126-1 definitions

### *3.2 Quality in use metrics*

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.

This broad view of quality has traditionally been the province of ergonomics, which is concerned with the factors in the physical and social environment which influence the extent to which people can achieve their goals. All these elements of the "work system" determine how people behave and whether they are successful in their tasks. The output of the work system can be measured as effectiveness, productivity, and satisfaction of the users (Figure 5).

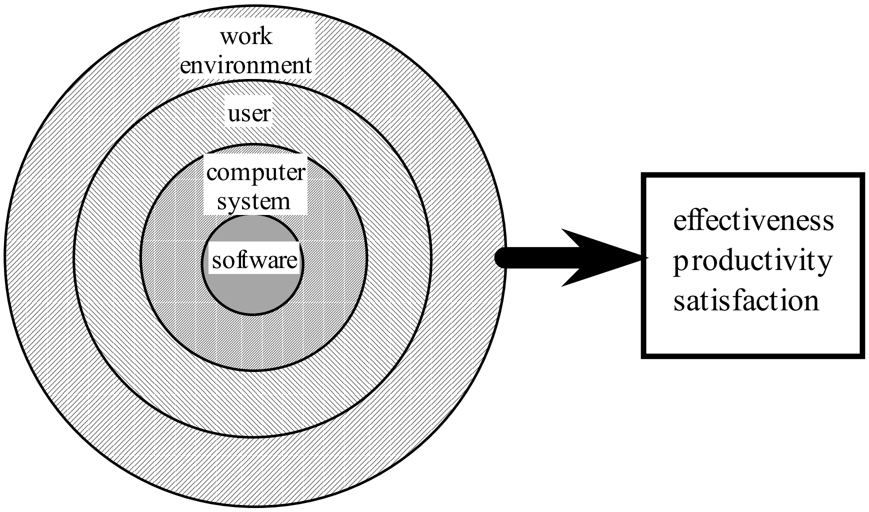

Fig. 5 Work system output

ISO 9241-11 takes this broad approach to measurement of usability. Effectiveness, productivity and satisfaction are influenced not only by the ease of use of the software, but also by the by the functionality, reliability and efficiency, as well as the suitability of the hardware, user and task. Measures of quality in use can thus be interpreted in two ways:

- if all other factors in the context of use are kept constant, the measures can be used to compare different software products or versions of software
- the values are of immediate relevance in a business context, and even if the software cannot be changed, it may be possible to improve quality in use by changes to the hardware, the tasks, or by training the user.

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 productivity relate the level of effectiveness achieved to the expenditure of resources. Relevant resources may include mental or physical effort, time, materials or financial cost. As time is usually the most important resource, productivity is most often measured as effectiveness divided by time.

Measures of satisfaction assess 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 with different types of users, tasks or environments. To specify or measure quality in use requires not only measures of effectiveness, productivity and satisfaction, but also details of the characteristics of the users, their goals and the relevant context of use.

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

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

## 4. Potential benefits of quality in use

Most existing development processes focus primarily on adherence to technical and process specifications. The objective of quality in use is that real products can be used by real people to achieve their tasks in the real world. This requires not only easy-to-use interfaces, but also the appropriate functionality and support for real business activities and work flows.

Quality in use should be the major design objective for an interactive product: that the product can be used for its intended purpose. Increased quality in use brings significant benefits which have been widely documented (e.g. Bias and Mayhew, 1994; Karat, 1992), and include:

- Increased efficiency. A system incorporating good ergonomic design and tailored to the preferred way of working, will allow the user to operate effectively and efficiently rather than lose vital time struggling with a poorly designed user interface and badly thought-out functionality.
- Improved productivity. A good interface to a well designed product will allow the user to concentrate on the task rather than the tool which, if designed inappropriately, can extend rather than reduce the time to do a task, as well as directly affecting other aspects of performance or quality.
- Reduced errors. A significant proportion of so-called “human error” can be attributed to a product with a poorly designed interface to functionality which is not closely matched to the user’s task needs. Avoiding inconsistencies, ambiguities or other interface design faults reduces user error.
- Reduced training. A poorly designed user interface and dialogue can prove a barrier to an otherwise technically sound system. A well designed system designed with a focus on the end-user can reinforce learning, thus reducing training time and effort.
- Improved acceptance. This is particularly important where usage is discretionary. Users would rather use and would be more likely to trust a well designed system which gives access to functionalities which make information easy to find and provides the information in a format which is easy to assimilate and use.

## 5. Barriers to quality in use

With all these potential benefits, why are many systems still not designed for greater quality in use? Achieving quality in use requires a user-centred approach to design, and introducing successful user-centred design into a large organisation requires cultural and technical change as well as strategic commitment:

- **cultural**: all those involved in the development of a system or product must be aware of the issues and activities involved in user-centred design to effect the best design decisions at a micro- and macro- level. Others affected by the system or product, for example end-users, who previously have not been involved in its development, need to be made aware of their new role in design. They must also put into place mechanisms for achieving the technical and cultural change outlined below.
- **technical**: development processes and procedures must include the user centred methods and activities which are appropriate for the organisation and whose purpose and benefit can be clearly demonstrated to the developers. Techniques are required for the selection of appropriate methods for each project, and for the dissemination of lessons learned to other projects in the organisation.
- **strategic**: the organisation and its management must set quality in use as a principal objective for systems development, but until recently there has been no way to accurately specify quality in use requirements prior to design. There is often a contractual requirement to deliver a product which matches the specification, but if the specification is not precise about quality in use, there is little incentive to make this a priority in design. Product developers regard it as inevitable that different users will have different perceptions of the quality of the product. As user perceived quality is regarded as a subjective judgement outside the control of the developer, meeting the technical specification becomes the sole objective of design.

## 6. Achieving quality in use

A series of European-funded projects (MUSiC: Bevan and Macleod, 1994; MAPI: Kirakowski and Bevan, 1997; INUSE, 1998; and RESPECT, 1998) have developed a co-ordinated approach to user centred design, which provide potential solutions to the cultural, technical and strategic barriers.

### *6.1 Cultural: Assessing quality in use maturity*

The INUSE project has developed methods (Earthy, 1998a, 1998b) for assessing an organisation's position on a quality in use maturity scale (which is also being proposed as an extension to SPICE: ISO/IEC TR 15504). This provides an indication of the management commitment to implement user-centred design:

**X: Ignorance** – *"We don't have problems with usability"*
Usability is not discussed as an issue.

**A: Uncertainty** – *"We don't know why we have problems with usability"*
User-Centred processes are not implemented, or fail to achieve their purpose

**B: Awakening** – *"Is it absolutely necessary to always have problems with usability?"*
User-Centred processes are implemented but are performed by inappropriate staff using sub-optimal methods

**C: Enlightenment** – *"Through management commitment and improvement of human-centred processes we are identifying and resolving our problems"*
User-Centred processes are implemented and produce results, but these results do not always give the expected benefits to the software development process

**D: Wisdom** – *"Usability defect prevention is a routine part of our operation"*
User-Centred processes are integrated into the software lifecycle and used to improve all work products

**E: Certainty** – *"We know why we do not have problems with usability"*
The culture of the organisation is user-centred

The experience of the INUSE and RESPECT projects is that much of European industry is at level 1, 2 or sometimes 3 on this scale.

### *6.2 Technical: Supporting user centred design*

ISO 13407 explains how to achieve quality in use by incorporating user centred design activities throughout the life cycle of interactive computer-based systems. User centred design is a multi-disciplinary activity, which incorporates human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and productivity, improving human working conditions, and counteracting the possible adverse effects of use on human health, safety and performance.

There are four user centred design activities that need to take place at all stages during a project. These are to:

- understand and specify the context of use
- specify the user and organisational requirements
- produce design solutions
- evaluate designs against requirements.

The iterative nature of these activities is illustrated in Figure 6. The process involves iterating until the objectives are satisfied. The sequence in which these are performed and the level of effort and detail that is appropriate depends on the design environment and the stage of the design process. INUSE has produced a Handbook of User Centred Design (Daly-Jones et al, 1997) which recommends appropriate methods for implementing user centred design in different environments. These include methods for gaining an in-depth understanding of user needs, documenting the intended context of use, setting usability goals, assessing design solutions from a user perspective prior to implementation, evaluating early prototypes with users, and then measuring whether usability goals have been achieved.

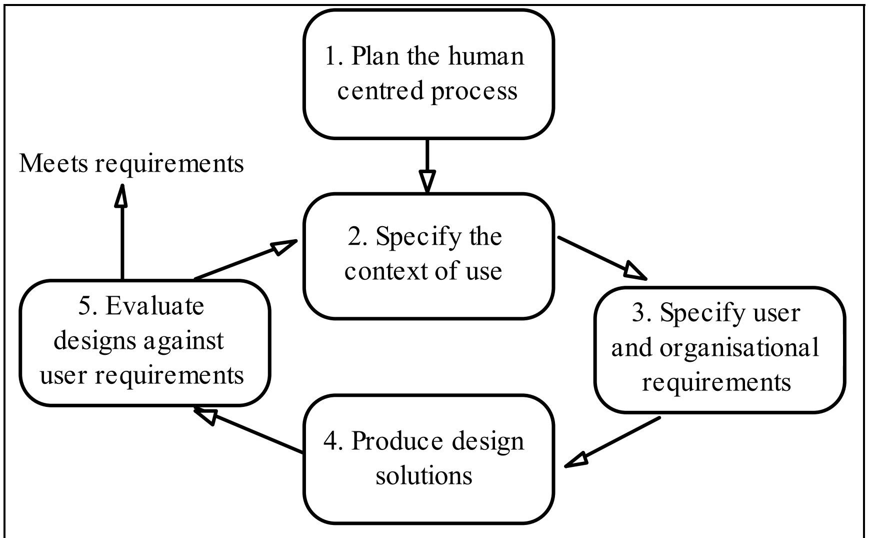
1. Plan the human centred process
Meets requirements
2. Specify the context of use
3. Specify user and organisational requirements
4. Produce design solutions
5. Evaluate designs against user requirements

Fig. 6. User centred design activities

### *6.3 Strategic: Setting quality in use objectives*

The MUSiC project demonstrated that it is possible to develop methods for specifying and testing quality in use based on the principles of the ISO 9241-11, which defines the usability of a product as the “effectiveness, efficiency and satisfaction” with which a well specified group of users can carry out a known set of tasks in a given context of use.

The MUSiC methods have been widely applied in industry, and include: the Usability Context Analysis Guide (Thomas and Bevan, 1995) which provides a procedure for documenting the context of use and context of evaluation; the Performance Measurement Method (Macleod et al, 1997) which provides a reliable and repeatable method for measuring effectiveness and productivity and diagnosing quality in use problems, and SUMI (Kirakowski, 1996) and MUMMS (Kirakowski, 1997) which enable different aspects of user satisfaction to be measured and areas of difficulty to be pin-pointed.

ISO 9001 specifies what is required for a quality system: a documented set of procedures intended to ensure that a product will meet initially stated requirements. A quality system is a desirable (though not sufficient) condition for achieving quality of the end product. Dealing with quality in use as part of a quality system for design and development of products involves the systematic identification of requirements for quality in use. These provide design targets which can be the basis for verification and validation of the resulting design.

ISO 9241-11 describes how the quality in use of a product can be defined, documented and verified as part of a quality system which conforms to ISO 9001 (Figure 7). The overall context of use needs to be identified, requirements for quality in use specified, usability issues monitored during development, and the achieved quality in use evaluated.

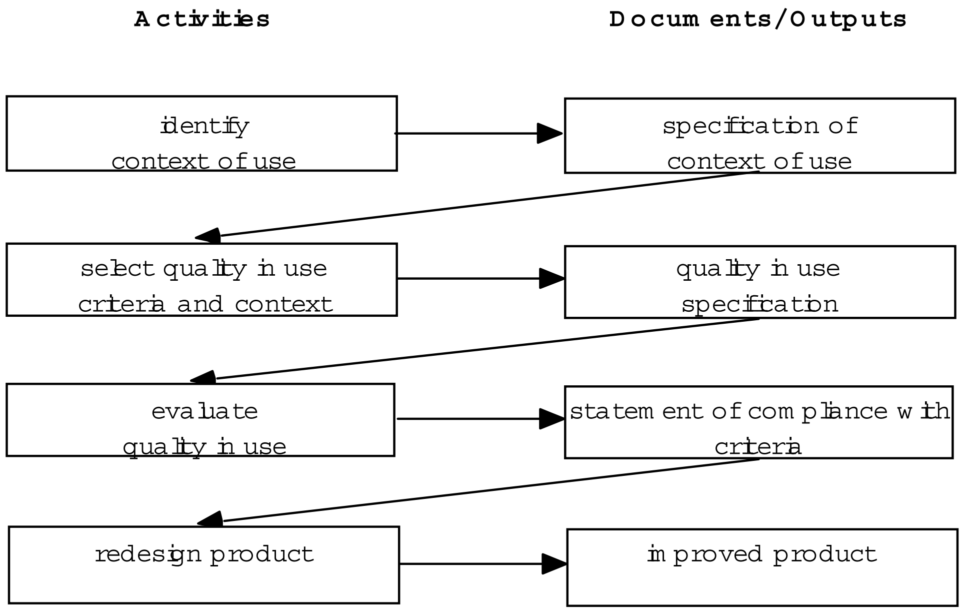
Activities
Documents/Outputs
identify context of use
specification of context of use
select quality in use criteria and context
quality in use specification
evaluate quality in use
statement of compliance with criteria
redesign product
improved product

Fig. 7. Quality Plan

*Identify context of use:* Information about the characteristics of users, their goals and tasks and the environments in which the tasks are carried out provides important information for use in the specification of overall product requirements, prior to development of specific quality in use requirements.

*Specify quality in use requirements:* Prior to development or acquisition, the purchasing organisation should specify the quality in use requirements which the system must meet and against which acceptance testing will be carried out. Specific contexts in which quality in use is to be measured should be identified, measures of effectiveness, productivity and satisfaction selected, and acceptance criteria based on these measures established.

*Monitor quality in use:* At various stages during the development process the developer should evaluate software quality. This information will enable objective decisions to be taken about the need for design changes to enhance quality in use, and about trade-offs which may be appropriate between different quality requirements.

*Evaluate quality in use:* The characteristics of the context in which a product is likely to be used need to be identified. To ensure the validity of test results the users, tasks and environments used for the evaluation should match the real context of use as closely as possible.

## 7. The value of user centred design

A recent study by the ESSI PET project (Vinter et al, 1996) found that 60% of software defects arise from usability errors, while only 15% of are related to functionality. They also found that the most significant defects in terms of cost of repair are those related to incorrect definition of requirements. The value of user centred methods in reducing these problems is confirmed by a survey by Keil and Carmel (1995) which showed that the probability of project success is improved by user-centred design. Yet user centred methods are still not widely used. There is an urgent need to incorporate more information about usability, user centred design and quality in use into education, training and methods for systems development. The INUSE and RESPECT projects have set up a network of Usability Support Centres to provide support for organisations which lack experience in this area.

## 8. Conclusions

The traditional view of product quality in ISO/IEC 9126 (1991) has been extended to include the needs of the user (ISO/IEC FCD 9126-1). Quality in use provides a link between the human factors approach to usability and user centred design. Although usability has traditionally been associated with improving the user interface, the broader goal of quality in use is closely related to business requirements. It provides a potential means for procurers to specify systems in a way which can give them greater confidence that the delivered system will meet their needs.

Achieving quality in use requires a user-centred design process which has cultural, strategic and technical implications. The skills and expertise required to apply methods for quality in use currently reside mainly in the human factors community, and much remains to be done to integrate these activities into existing main stream design practice and into other related standards (Bevan and Azuma, 1997).

## 9. Acknowledgements

I would like to thank the other participants in ISO/IEC JTC1/SC7/WG6 who contributed to the drafts of the standards described in this paper. The work was partly funded by the European Commission RESPECT project.

## References

Bevan N, Measuring usability as quality of use. *Journal of Software Quality*, 4, 115-130 (1995a).

Bevan N, Usability is quality of use, in Anzai & Ogawa (eds) *Proc. 6th International Conference on Human Computer Interaction*, July 1995. Elsevier, 1995b.

Bevan N and Azuma M, Quality in use: Incorporating human factors into the software engineering lifecycle, in *Proceedings of the Third IEEE International Software Engineering Standards Symposium and Forum (ISESS'97)*, p169-179, 1997.

Bevan N and Macleod M, Usability measurement in context. *Behaviour and Information Technology*, 13, 132-145, 1994.

Bias, G. and Mayhew, D. *(Eds)*, *Cost-Justifying Usability*. Academic Press, 1994.

Daly-Jones, O, Thomas, C, Bevan, N., *Handbook of user centred design*. National Physical Laboratory, Teddington, Middx, UK, 1997.

Earthy, J (1998a) Usability Maturity Model: Attitude Scale. INUSE deliverable D5.1.4s, http://www.npl.co.uk/inuse

Earthy, J (1998b) Usability Maturity Model: Processes. INUSE deliverable D5.1.4p, http://www.npl.co.uk/inuse

INUSE, see http://www.npl.co.uk/inuse, 1998.

ISO 8402, *Quality management and quality assurance -- Vocabulary,* 1994.

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

ISO/IEC 9126, *Software product evaluation - Quality characteristics and guidelines for their use*, 1991.

ISO/IEC FCD 9126-1, *Software product quality - Part 1: Quality model,* 1998.

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

ISO FDIS 13407, *User centred design process for interactive systems,* 1998 .

ISO/IEC 14598-1, *Information Technology - Evaluation of Software Products - Part 1 General guide,* 1998.

ISO/IEC TR 15504, *Software process assessment,* 1998.

Karat, C.M., Cost-justifying human factors support on development projects. *Human Factors Society Bulletin,* 35 (11), 1-8, 1992.

Keil, M, & Carmel, E., Customer developer links in software-development. *Communications of the ACM*. 38(5), pp33 – 44, 1995.

Kirakowski J, *The Use of Questionnaire Methods for Usability Assessment*. http://www.ucc.ie/hfrg/questionnaires/sumi/sumipapp.htm, 1996.

Kirakowski J, *Measuring the Usability of Multi-Media Systems.* http://www.ucc.ie/hfrg/questionnaires/mumms/info.html, 1997.

Kirakowski J and Bevan N, MAPI: MUSiC assisted process improvement. In: *Human comfort and security of information systems*, Varghese K and Pfleger S (eds). Springer-Verlag, Berlin, 1997.

Macleod M, Bowden R, Bevan N and Curson I., *The MUSiC Performance Measurement Method.* Behaviour and Information Technology, 16, 1997.

RESPECT, see http://www.npl.co.uk/respect, 1998.

Thomas C and Bevan N (eds), Usability Context Analysis: A practical guide, Version 4. National Physical Laboratory, Teddington, UK, 1995.

Vinter, O., Poulsen, P.M., Lauesen, S, Experience driven software process improvement, *Software Process Improvement '96*, 3rd-5th December, 1996, Brighton, 1996.

**Nigel Bevan**
Serco Usability Services

Alderney House, 4 Sandy Lane
Teddington, Middx, TW11 0DU, UK
tel: +44 181 614 3811
fax: +44 181 614 3765
nbevan@usability.serco.com