Quality in use: incorporating human factors into the software engineering lifecycle

Nigel Bevan

National Physical Laboratory
Division of Information Technology and Computing
Teddington, Middlesex, TW11 0LW, UK
Nigel@hci.npl.co.uk
tel: +44 181 943 6993

ABSTRACT

The relationship between the different approaches to quality in ISO standards is reviewed, contrasting the manufacturing approach to quality in ISO 9000 (quality is conformance to requirements) with the product orientation of ISO 8402 (quality is the presence of specified features) and the goal orientation of quality in use in ISO 14598-1 (quality is meeting user needs). It is shown how ISO 9241-11 enables quality in use to be measured, and ISO 13407 defines the activities necessary in the development lifecycle for achieving quality in use.

APPROACHES TO QUALITY

Although the term quality seems self-explanatory in everyday usage, in practice there are many different views of what it means and how it should be achieved as part of a software production process.

ISO DEFINITIONS OF QUALITY

ISO 9000 is concerned with quality assurance to provide confidence that a product will satisfy given requirements. Interpreted literally, this puts quality in the hands of the person producing the requirements specification - a product may be deemed to have quality even if the requirements specification is inappropriate.

This is one of the interpretations of quality reviewed by Garvin (1984). He describes it as *Manufacturing quality:* a product which conforms to specified requirements.

A different emphasis is given in ISO 8402 which defines quality as the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs. This is an example of what Garvin calls *Product quality:* an inherent characteristic of the product determined by the presence or absence of measurable product attributes.

Many organisations would like to be able to identify those attributes which can be designed into a product or evaluated to ensure quality. ISO 9126 (1992) takes this approach, and categorises the attributes of software quality as: functionality, efficiency, usability, reliability, maintainability and portability.

To the extent that user needs are well-defined and common to the intended users this implies that quality is an inherent attribute of the product. However, if different groups of users have different needs, then they may require different characteristics for a product to have quality for their purposes. Assessment of quality thus becomes dependent on the perception of the user.

USER PERCEIVED QUALITY AND QUALITY IN USE

Garvin defines *User perceived quality* as the combination of product attributes which provide the greatest satisfaction to a specified user.

Most approaches to quality do not deal explicitly with user-perceived quality. User-perceived quality is regarded as an intrinsically inaccurate judgement of product quality. For instance Garvin, 1984, observes that "Perceptions of quality can be as subjective as assessments of aesthetics".

However, there is a more fundamental reason for being concerned with user-perceived quality. Products can only have quality in relation to their intended purpose. For instance, the quality attributes required of an office carpet may be very different from those required of a bedroom carpet. For conventional products this is assumed to be self-evident. For general-purpose products it creates a problem. A text editor could be used by programmers for producing code, or by secretaries for producing letters. Some of the quality attributes required will be the same, but others will be different. Even for a word processor, the functionality, usability and efficiency attributes required by a trained user

may be very different from those required by an occasional user.

Reconciling work on usability with traditional approaches to software quality has led to another broader and potentially important view of quality which has been outside the scope of most existing quality systems. This embraces user-perceived quality by relating quality to the needs of the user of an interactive product.

ISO 14598-1 defines *External quality* as the extent to which a product satisfies stated and implied needs when used under specified conditions.

This moves the focus of quality from the product in isolation to the satisfaction of the needs of particular users in particular situations. The purpose of a product is to help users achieve particular goals, which leads to the definition of *Quality in use* in ISO DIS 14598-1 as the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in specified environments.

A product meets the requirements of the user if it is effective (accurate and complete), efficient in use of time and resources, and satisfying, regardless of the specific attributes it possesses.

Specifying requirements in terms of performance has many benefits. This is recognised in the rules for drafting ISO standards (ISO, 1992) which suggest that to provide design flexibility, standards should specify the performance required of a product rather than the technical attributes needed to achieve the performance.

Quality in use is a means of applying this principle to the performance which a product enables a human to achieve. An example is the ISO standard for VDT display screens (ISO 9241-3). The purpose of the standard is to ensure that the screen has the technical attributes required to achieve quality in use. The current version of the standard is specified in terms of the technical attributes of a traditional CRT. It is intended to extend the standard to permit alternative new technology screens to conform if it can be demonstrated that users are as effective, efficient and satisfied with the new screen as with an existing screen which meets the technical specifications.

SOFTWARE QUALITY IN USE: ISO 14598-1

The purpose of designing an interactive system is to meet the needs of users: to provide quality in use (see Figure 1, from ISO/IEC 14598-1). The internal software attributes will determine the quality of a software product in use in a particular context. Software quality attributes are the cause, quality in use the effect. Quality in use is (or at least should be) the objective, software product quality is the means of achieving it.

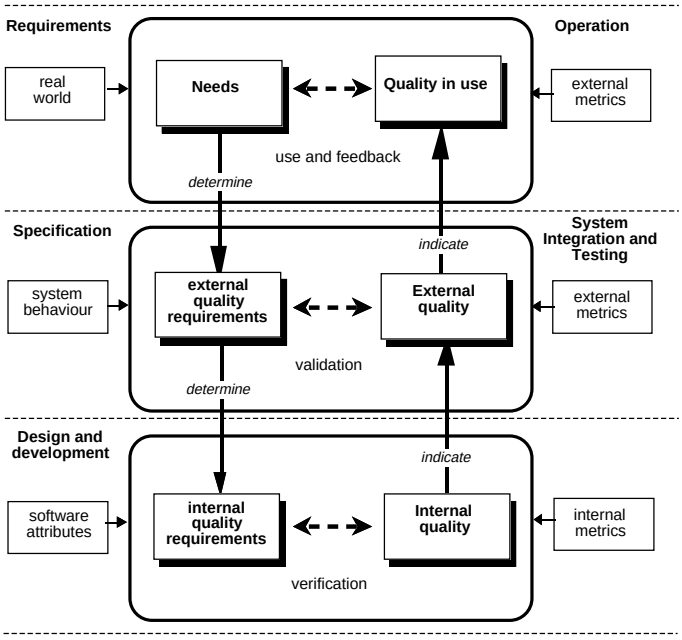

Figure 1. Quality in the software lifecycle

The users' needs can be expressed as a set of requirements for the behaviour of the product in use (for a software product, the behaviour of the software when it is executed). These requirements will depend on the characteristics of each part of the overall system including hardware, software and users.

The requirements should be expressed as metrics which can be measured when the system is used in its intended context, for instance by measures of effectiveness, efficiency and satisfaction. At this level, the required system characteristics could be minimum values for the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in specified environments.

External quality can only be assessed for a complete hardware/software system of which the software product is a part. External metrics are applied when executing the software. The values of external measures necessarily depend on more than the software, so the software has to be evaluated as part of a working system. Software which performs satisfactorily in one environment may show quality defects in another environment. External evaluation of quality characteristics should therefore take place under conditions which emulate as closely as possible the expected conditions of use. External measurements of characteristics are made when the code is complete, though as it may not be possible to emulate the exact conditions of use (e.g. network environment and user characteristics), external measures are often only indicators of the actual quality in use.

The required values of these external metrics provide goals for design. To achieve these goals the internal attributes of the system can be specified as internal requirements. These attributes of the software can be evaluated to produce internal metrics verifying how closely the internal

requirements have been met. Although these attributes contribute to achieving quality in use, users and tasks vary so much that conformance to requirements for internal metrics is rarely sufficient to ensure quality in use.

If the external quality requirements are not achieved the results of the evaluation can be used as feedback to modify the internal software attributes in order to improve the external quality, thus supporting a continual improvement process.

For the purposes of development, internal quality requirements are defined which enable the quality of intermediate products to be verified. The internal properties (e.g. the specification or source code) of the software can be measured by internal metrics. Internal metrics are of most interest during the development process. They can be measured in their own right as essential pre-requisites for external quality. They can also be used as indicators of external attributes. Modularity and traceability are examples of internal attributes which can be measured. Achievement of the required internal quality will contribute to meeting the external requirements of the software in use. Internal software quality metrics can thus be used as indicators to estimate final software quality (see Figure 2).

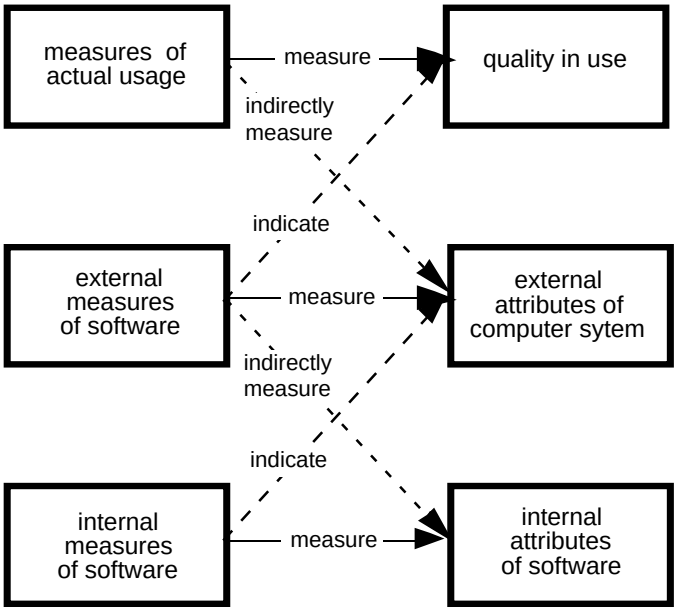

Figure 2 Relationships between measures and attributes

For example, response time is an important measure required to evaluate the usability and the efficiency of the software, but response time cannot be measured during development. In order to evaluate the efficiency of the product during development, path length could be measured based on the intermediate product or specifications. These could be used as indicators which provided rough estimates of response time under certain conditions.

It is very important that internal software quality attributes are directly related to external quality requirements, so that the quality characteristics of software products under development (both intermediate and end item software products) can be assessed with respect to final system in-use quality needs. Internal metrics are of little value unless there is evidence that they are related to external quality.

QUALITY MODEL: ISO 9126

In order to evaluate software it is necessary to select relevant quality characteristics. This should be done using a quality model which breaks software quality down into different characteristics. ISO/IEC 9126 provides a general-purpose model which defines six broad categories of characteristics of the software: functionality, reliability, usability, efficiency, maintainability and portability. These can be further broken down into subcharacteristics which have measurable attributes (figure 3).

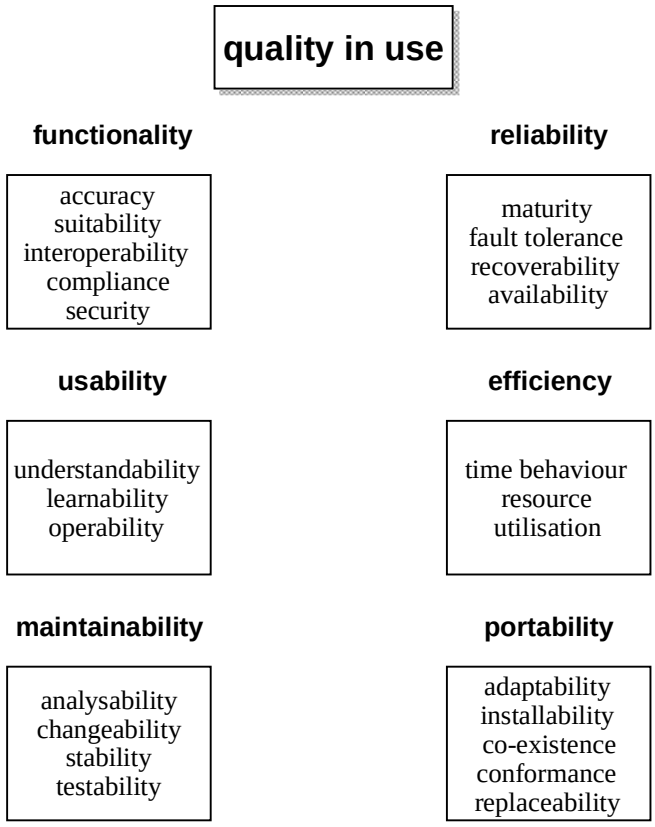

Figure 3. ISO 9126 quality model

Internal software product quality attributes are the measurable properties of a software product that influence its ability to satisfy stated and implied needs. One or more attributes can be used to assess a particular software quality characteristic or subcharacteristic (Figure 4).

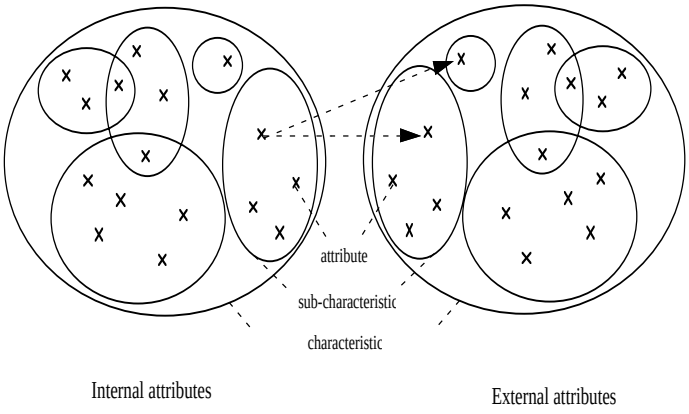

Figure 4: Quality characteristics, subcharacteristics and attributes

Sufficient internal and external attributes need to be identified for each required subcharacteristic.

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 characteristics and subcharacteristics provide a useful checklist of issues related to quality, but other ways of categorising quality may be more appropriate in particular circumstances.

CHOICE OF METRICS

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. The internal attributes are said to be indicators of the external characteristics. Note that the correlation between internal attributes and external measures is never perfect however, and the effect that a given internal attribute has upon an associated external measure will be determined by experience, and will depend on the particular context in which the software is used.

In the same way, external subcharacteristics (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) can be traced to external sub-characteristics (e.g. suitability or operability) and the associated internal attributes which have to be changed.

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. These are primarily external metrics (examples are given in ISO/IEC 9126-2 currently under development).

When developing a software product the intermediate products should be evaluated using metrics which measure intrinsic properties derived from simulated behaviour. The primary purpose of these internal metrics is to ensure that the required external quality is achieved (examples are given in ISO/IEC 9126-3 currently under development).

The particular metrics used will depend on the business priorities for the product and the needs of the evaluator. The ISO/IEC 9126 model supports a variety of evaluation requirements, for example:

- a user or a user's business unit could evaluate the suitability of a software product using metrics for quality in use;

- an acquirer could evaluate a software product against criterion values of external measures of functionality, reliability, usability and efficiency;

- a maintainer could evaluate a software product using metrics for maintainability;

- a person responsible for implementing the software in different environments could evaluate a software product using metrics for portability;

- a developer could evaluate a software product against criterion values using internal measures of any of the quality characteristics.

MEASURING QUALITY IN USE

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 end user. It can be measured by the extent to which the specified users can achieve their goals with effectiveness, task efficiency and satisfaction**.** Effectiveness can be measured by the accuracy and completeness with which users achieve specified goals, task efficiency by the resources expended in relation to task effectiveness, and satisfaction by attitudes to the use of the product. Examples of metrics for quality in use are given in ISO/IEC 9126-2.

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 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

QUALITY IN USE AND USABILITY: ISO 9241-11

Quality in use may be influenced by any of the quality characteristics, and is thus broader than usability, which in this paper is used with the meaning given in ISO 9126-1: understandability, learnability and operability. (Note that the term usability is used in a broader sense in ISO 9241-11 where it has a similar meaning to the term quality in use in ISO 14598-1 and ISO 9126-1.)

The objective of designing and evaluating for quality in use is to enable users to achieve goals and meet needs in a particular context of use. 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.

ISO 9241-11 emphasises that quality in use is dependent on the context of use and that the level of quality in use 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 all influence the quality in use of a product in a working environment.

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.

Specifying and measuring the quality in use of products

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. The components and the relationships between them are illustrated in figure 5.

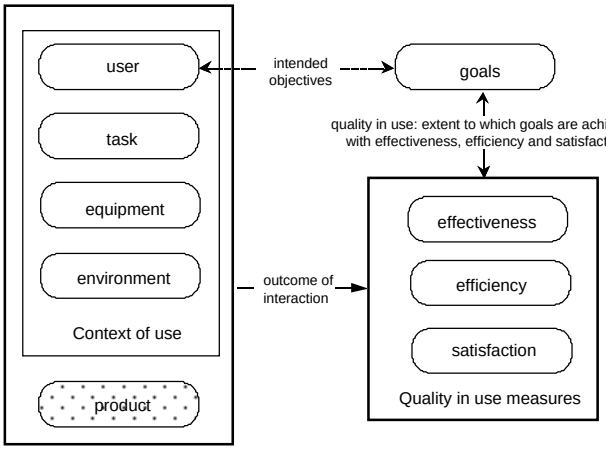

Figure 5: Quality in use framework

Information needed

When specifying or measuring quality in use, the following information is needed:

- a description of the components of the context of use including users, equipment, environments, and tasks. This may be a description of an existing context, or a specification of intended contexts. The relevant aspects of the context and the level of detail required will depend on the scope of the issues being addressed. The description of the context needs to be sufficiently detailed so that those aspects of the context which may have a significant influence on quality in use could be reproduced;
- quality in use measures consisting of target or actual values of effectiveness, efficiency, and satisfaction for the intended contexts.

The specification or measurement of the quality in use of a particular product should identify the overall goal of the user, specific sub-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.

Tasks are the activities undertaken to achieve a goal. Characteristics of tasks which may influence quality in use should be described as part of the context of use, e.g. the frequency and the duration of performance.

CHOICE OF MEASURES

A specification of quality in use consists of target or actual values of effectiveness, efficiency, and satisfaction for the required contexts. It is normally necessary to provide at least one measure for each of effectiveness, efficiency and satisfaction.

Measures of effectiveness relate the goals or sub-goals of the user to the accuracy and completeness with which these goals can be 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 could 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. If measures of quality in use are obtained over short periods of time the values may not take account

of infrequent events which could have a significant impact on quality in use, for example intermittent system errors.

SPECIFICATION AND EVALUATION OF QUALITY IN USE DURING DESIGN

Prior to development of a custom system, a purchasing organisation can use the information in ISO 9241-11 as a framework for specifying the quality in use requirements which the system must meet and against which acceptance testing may be carried out. Specific contexts in which quality in use is to be measured should be identified, measures of effectiveness, efficiency and satisfaction selected, and acceptance criteria based on these measures established.

At various stages during the development process the developer can measure the quality in use achieved against these targets. This information enables 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 quality in use and other requirements.

These activities provide a basis for defining, documenting and verifying quality in use as a part of a quality plan. Figure 6 illustrates how quality in use can be incorporated into a quality plan (e.g. such as described in ISO 9000-3).

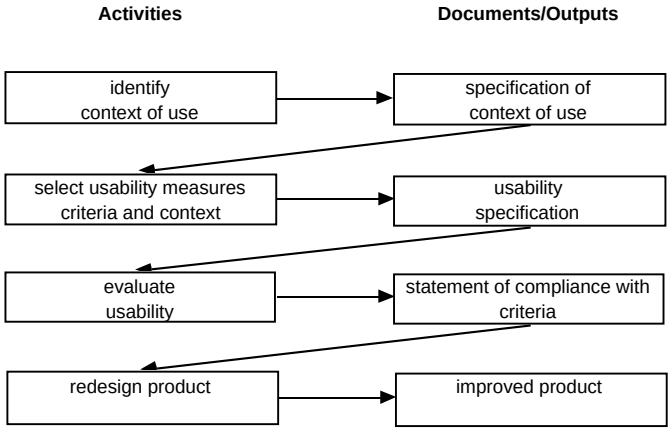

Figure 6: Quality Plan

Other applications of quality in use measures

Measures of quality in use can be used to evaluate how any component of a working environment affects the quality of the product in use. For example, it may be appropriate to consider the amount of user training to be provided, changes in lighting, or re-organisation of the task. In this case the element which is the object of design or evaluation is considered to be subject to potential variation, while the other elements of the working environment are treated as fixed. Measures of effectiveness, efficiency and satisfaction can be used to specify or evaluate the effect on quality in use of different operating systems, different types of lighting, or different amounts of user training respectively.

USER CENTRED DESIGN: ISO 13407

ISO CD 13407 provides guidance on achieving quality in use by incorporating user centred design activities throughout the life cycle of interactive computer-based systems. It describes user centred design as a multi-disciplinary activity, which incorporates human factors and ergonomics knowledge and techniques with the objective of enhancing effectiveness and efficiency, 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 7. The process involves iterating until the objectives are satisfied.

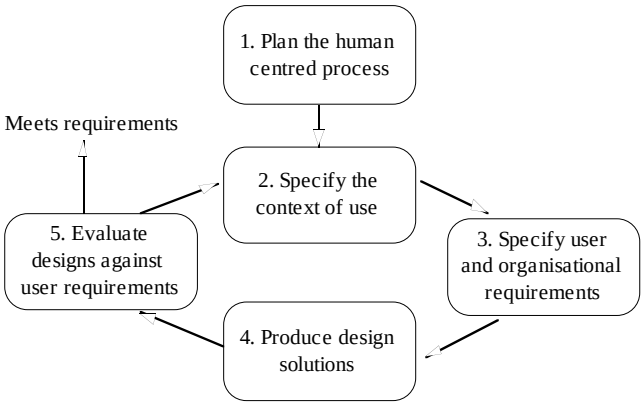

Figure 7 - The interdependence of user centred design activities

The sequence in which these are performed and the level of effort and detail that is appropriate varies depending on the design environment and the stage of the design process.

Understand and specify the context of use

The characteristics of the users, tasks and the organisational and physical environment define the context in which the product is used. It is important to understand and identify the details of this context in order to guide early design decisions, and to provide a basis for evaluation. If an existing product is to be upgraded or enhanced, the context will already be well understood. If there are extensive results from user feedback, help desk reports and other data, these provide a basis for prioritising user requirements for system modifications and changes. For new products or systems, it is necessary to gather information about its context of use.The context in which the product is to be used should be identified in terms of: 1. **the characteristics of the intended users**. Relevant

characteristics of the users can include knowledge, skill, experience, education, training, physical attributes, habits and capabilities. If necessary, define the characteristics of different types of user, for example with different levels of experience or performing different roles (maintainers, installers etc.).2. **the tasks the users are to perform**. The description should include the overall goals of use of the system. The characteristics of tasks that can influence quality in use in typical scenarios should be described, e.g. the frequency and the duration of performance. The description should include the allocation of activities and operational steps between the human and technological resources. Tasks should not be described solely in terms of the functions or features provided by a product or system.

3. **the environment in which the users are to use the product**. The description of the hardware, software and materials can be in terms of a set of products, one or more of which can be the focus of human-centred specification or evaluation, or it can be in terms of a set of attributes or performance characteristics of the hardware, software and other materials. Relevant characteristics of the physical and social environment also need to be described. Aspects which may need to be described include attributes of the wider technical environment (e.g. a local area network), the physical environment (e.g. workplace, furniture), the ambient environment (e.g. temperature, humidity), the legislative environment (e.g. laws, ordinances, directives and standards) and the social and cultural environment (e.g. work practices, organisational structure and attitudes).The output from this activity should be a description of the relevant characteristics of the users, tasks and environment which identifies what aspects have an important impact on the system design.

This description is unlikely to be a single output that is issued once. It is more often a "working document" that is first produced in outline terms and is then reviewed, maintained, extended and updated during the design and development process.

Specify the user and organisational requirements

In most design processes, there is a major activity specifying the functional and other requirements for the product or system. For user centred design, it is essential to extend this activity to include an explicit statement of user and organisational requirements, in relation to the context of use description, in terms of:

- usability of the human-computer interface;
- quality in use: effectiveness and efficiency of task performance and user satisfaction;

Objectives can be set with appropriate trade-offs identified between the different requirements. The requirements should be stated in terms that permit subsequent testing.

Produce design solutions

The next stage is to create potential design solutions by drawing on the established state of the art and the experience and knowledge of the participants. The process therefore involves:

- using existing knowledge to develop proposed multi-disciplinary design solutions;
- making those design solutions more concrete (using simulations, models, mock-ups etc.);
- showing the solutions to users and allowing them to perform tasks (or simulated tasks);
- using this feedback to improve the designs;
- iterating this process until the user centred design goals are met.

Evaluate designs against requirements

Evaluation is an essential step in user centred design and should take place at all stages in the system life cycle. Evaluation can be used to:

- provide feedback which can be used to improve design
- assess whether user and organisational objectives have been achieved
- monitor long term use of the product or system

Early in design the emphasis is on obtaining feedback that can be used to guide design, while later when a realistic prototype is available it is possible to measure whether user and organisational objectives have been achieved.

In the early stages of the development and design process, changes are relatively inexpensive. The longer the process has progressed and the more fully the system is defined, the more expensive the introduction of changes is. It is therefore important to start evaluation as early as possible.

APPLYING THE STANDARDS: MUSIC METHODS

The MUSiC methods were developed by the European MUSiC (Measuring the Usability of Systems in Context) project to provide valid and reliable means for specifying and measuring quality in use, while also giving diagnostic feedback which enables the design to be modified to improve quality in use. MUSiC includes tools and techniques which implement the principles of ISO 9241-11 for specifying the context of use and measuring user performance and satisfaction.

Measuring User Performance

The MUSiC Performance Measurement Method gives reliable measures of the effectiveness and efficiency of system use, by evaluating the extent to which specific task goals are achieved, and the times taken to achieve task goals. It can also give measures of time spent unproductively (for example, overcoming problems and seeking help), plus diagnostic data about the location of such difficulties. The diagnostic information helps identify where specific problems are encountered and where improvements need to be made.

The people observed and the tasks they perform are selected as a result of a context of use study based on the MUSiC

Usability Context Analysis Handbook (Thomas and Bevan, 1995). The method is fully documented in the MUSiC Performance Measurement Handbook (Rengger et al., 1993), and is supported by a software tool, DRUM (Macleod and Rengger, 1993), which greatly speeds up analysis of the video, and helps manage the evaluation.

Effectiveness

Measures of effectiveness relate the goals or sub-goals of using the system to the accuracy and completeness with which these goals can be achieved.

For example if the desired goal is to transcribe a 2-page document into a specified format, then accuracy could be specified or measured by the number of spelling mistakes and the number of deviations from the specified format, and completeness by the number of words of the document transcribed divided by the number of words in the source document.

In the MUSiC Performance Measurement Method the effectiveness with which a user uses a product to carry out a task is comprised of two components: the quantity of the task the user completes, and the quality of the goals the user achieves (Rengger et al 1993). Quantity is a measure of the amount of a task completed by a user. It is defined as the proportion of the task goals represented in the output of the task. Quality is a measure of the degree to which the output achieves the task goals.

As Quantity and Quality are both measured as percentages, Task Effectiveness can be calculated as a percentage value:

$$\text{Task Effectiveness} = 1/100\ (\text{Quantity} \times \text{Quality})\ \%$$

It is sometimes necessary to calculate effectiveness for a number of sub-tasks, for instance this might be the individual elements in a drawing task. The average effectiveness across sub-tasks is a useful measure of the product's capabilities, but may not reflect the effectiveness of the final task output. For instance, if the user was unable to save the final drawing, the overall effectiveness would be zero. Similarly, the effectiveness may be reduced if the final drawing contains additional unwanted elements.

Efficiency

Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources. The resources may be mental or physical effort, which can be used to give measures of human efficiency, or time, which can be used to give a measure of temporal efficiency, or financial cost, which can be used to give a measure of economic efficiency. For example:

$$\text{Temporal Efficiency} = \frac{\text{Effectiveness}}{\text{Task Time}}$$

Efficiency measures can be used to compare the efficiency of:

- Two or more similar products, or versions of a product, when used by the same user groups for the same tasks in the same environments
- Two or more types of users when using the same product for the same tasks in the same environment
- Two or more tasks when carried out by the same users on the same product in the same environment.

Task time itself is sometimes used as a quality in use measure. This is appropriate if all users complete the task satisfactorily (ie with 100% effectiveness). The advantage of the efficiency measure is that it provides a more general measure of work rate by trading off the quantity and quality of output against time.

From the point of view of the organisation employing the user, the resource consumed is the cost to the organisation of the user carrying out the task, for instance:

- The labour costs of the user's time
- The cost of the resources and the equipment used
- The cost of any training required by the user

This provides a means of measuring another approach to quality identified by Garvin (1994) *Economic quality*: a product which provides performance at an acceptable price. For example if the desired goal is to print copies of a report, then efficiency could be specified or measured by the number of usable copies of the report printed, divided by the resources spent on task such as labour hours, process expense and materials consumed.

MEASURING SATISFACTION

Satisfaction is composed of comfort and acceptability of use. Comfort refers to overall physiological or emotional responses to use of the system (whether the user feels good, warm, and pleased, or tense and uncomfortable). Acceptability of use may measure overall attitude towards the system, or the user's perception of specific aspects such as whether the user feels that the system supports the way they carry out their tasks, do they feel in command of the system, is the system helpful and easy to learn. If satisfaction is low when efficiency is high, it is likely that the user's goals do not match the goals selected for measurement of efficiency.

Satisfaction can be specified and measured by attitude rating scales such as SUMI (see below), but for existing systems attitude can also be assessed indirectly, for instance by measures such as the ratio of positive to negative comments during use, rate of absenteeism, or health problem reports. Measures of satisfaction can provide a useful indication of the user's perception of usability, even if it is not possible to obtain measures of effectiveness and efficiency.

Cognitive workload is closely related to comfort: even if a system is apparently acceptable for use, it may be low in comfort if it demands too little or too much mental effort. A

task demanding too little mental effort may result in a lowered efficiency because it leads to boredom and lack of vigilance, which directly lowers effectiveness. Excessive cognitive workload may also result in lowered effectiveness, if it causes information to be missed and results in errors. This is a particularly important issue in situations where safety is critical, e.g. air traffic control and process control. Measures of cognitive workload can be used to predict these types of problems. MUSiC uses the SMEQ (Zijlstra, 1993) and TLX (NASA, 1986) questionnaires.

Subjective usability - SUMI

To measure user satisfaction, and hence assess user perceived software quality, University College Cork has developed the Software Usability Measurement Inventory (SUMI) as part of the MUSiC project (Kirakowski, Porteous and Corbett, 1992). SUMI is an internationally standardised 50-item questionnaire, available in seven languages. It takes approximately 10 minutes to complete, and contains questions such as:

- Using this software is frustrating
- Learning how to use new functions is difficult

The results which SUMI provide are based on an extensive standardisation database built from data on a full range of software products such as word processors, spreadsheets, CAD packages, communications programs etc. SUMI results have been shown to be reliable, and to discriminate between different kinds of software products in a valid manner.

SUMI provides an Overall Assessment and a Usability Profile which breaks the Overall Assessment down into 5 sub-scales:

Affect, Efficiency, Helpfulness, Control, and Learnability.

Item Consensual Analysis can be used to list those questionnaire items on which the software being rated was significantly better or worse than the standard of comparison. This provides valuable evidence of specific short-comings of the software.

CONTEXT USED FOR EVALUATION

Unless the evaluation can take place in conditions of actual use, it will be necessary to decide which attributes of the actual or intended context of use are to be represented in the context used for evaluation. When specifying or evaluating usability it is therefore important that the context selected is representative of the important aspects of the actual or intended context of use. Particular attention should be given to those attributes which are judged to have a significant impact on the quality in use of the overall system.

A systematic method for describing the context of use and specifying the context used for evaluation has been developed by the MUSiC project. In cases where it is not feasible to match all of the components of the context of evaluation to the context of use, particular care must be taken not to over-generalise from the results of the study. A description of the context of evaluation is an essential part of the report of any evaluation.

The MUSiC Usability Context Analysis Handbook incorporates a Context of Use Questionnaire and a Context Report Table, with practical instructions on how to use them to describe a product's context of use, and to specify an appropriate context for evaluation.

A detailed description of the intended context of use should ideally form part of the requirements specification for the product. As this is rarely the case, it is generally necessary, prior to any evaluation, to hold a context meeting. At the meeting, the relevant stakeholders including the product manager, developers, trainers, usability specialists and user representatives work together to agree on how the product is expected to be used and which user types and tasks should be the focus of the evaluation (Macleod, 1994).

EXAMPLE

As an example, the efficiency results obtained for the tasks of depositing and withdrawing cash and cheques in a bank before and after introduction of a new system are shown in Figure 8.

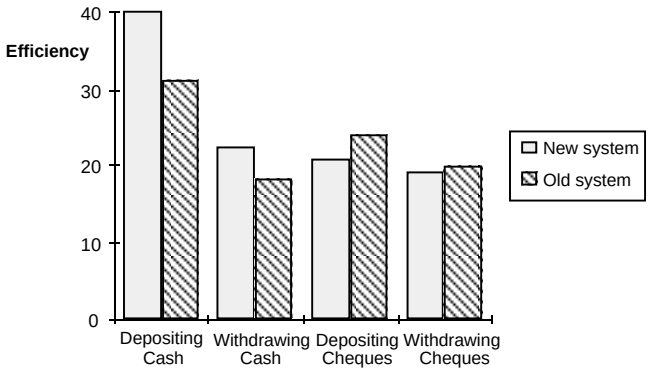

Figure 8. Efficiency (percentage effectiveness per minute)

The new system shows improvements for cash but not cheque transactions. Videotapes of the interaction were inspected to identify the causes of poorer efficiency for cheque transactions, so that this could be improved in the next prototype in order to meet the initial goals.

SUMI was administered after the users had completed the transaction tasks with the old and new system. The results for the old system shown in Figure 9 reflect its antiquated nature. Although it was quite efficient, it was not well-liked. The results for the new system show that the goal of perceived ease of use has been met. The high rating for learnability is also very encouraging, particularly for a prototype.

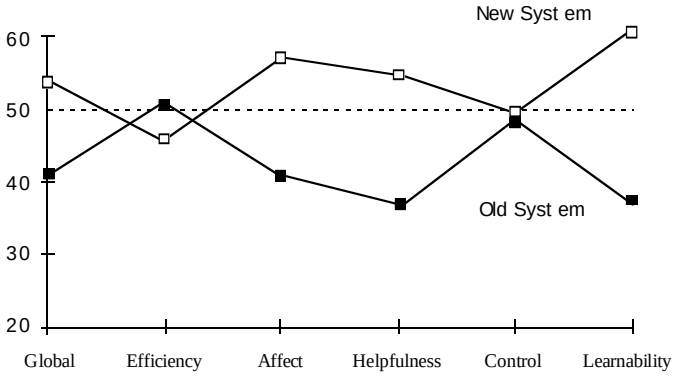

Figure 9. SUMI scores

INUSE METHODS

NPL is now leading a consortium of European organisations developing a broad range of methods (including those described above) to provide a comprehensive approach to achieving quality in use through user centred design (INUSE, 1996). Three broad categories of method shown in Figure 10 have been defined to support the user centred design process in Figure 7.

Planning

- ISO 13407 as a structure
- Usability Context Analysis
- Usability Capability Maturity

Early development

- Guides and standards
- Early prototyping & evaluation methods
- Usability inspection methods

Late development

- Usability testing (performance measurement)
- Subjective assessment questionnaire (SUMI)
- Conformance to guides and standards

Figure 9. Core INUSE Methods

Further work remains to be done to harmonise these activities with the other activities required for software product evaluation described in the relevant parts of ISO 14598 which are currently are under development (figure 10).

ISO/IEC 14598 Information Technology - Software product evaluation

Part 1: General overview
Part 2: Planning and management
Part 3: Process for Developers
Part 4: Process for Acquirers
Part 5: Process for Evaluators
Part 6: Documentation of evaluation modules

Figure 10. ISO/IEC 14598

ACKNOWLEDGEMENTS

This paper includes text and diagrams adapted from drafts of standards I have contributed to: ISO/IEC 14598-1, ISO/IEC 9126-1, ISO 9241-11 and ISO 13407. I would particularly like to acknowledge the contributions of Prof Azuma and the other co-editor of ISO/IEC 14598-1: Andrew Chruscicki, the editor of ISO 13407: Tom Stewart, and all the other participants in ISO/IEC JTC1/SC7/WG6, ISO TC159/SC4/WG5/SG2 and ISO TC159/SC4/WG6 who contributed to the drafts of these standards.

REFERENCES

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

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

Bevan N (1995) Human-Computer Interaction standards. In: Anzai & Ogawa (eds) Proc. 6th International Conference on Human Computer Interaction, July 1995. Elsevier.

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

Garvin (1984) What does "product quality" really mean? Sloane Management Review, Fall 1984.

INUSE (1996) see http://www.npl.co.uk/inuse

ISO (1987) ISO 9001: Quality systems - Model for quality assurance in design/development, production, installation and servicing.

ISO (1991) ISO 9126: Software product evaluation - Quality characteristics and guidelines for their use.

ISO (1992) Directives Part 2 - Methodology for the development of international standards. ISO/IEC, Switzerland.

ISO (1993) ISO 9241-3 Visual display requirements.

ISO (1994) ISO DIS 8402: Quality Vocabulary.

ISO (1996) ISO DIS 9241-11 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability.

ISO (1996) ISO/IEC DIS 14598-1 Information Technology - Evaluation of Software Products - Part 1 General guide.

ISO (1996b) ISO CD 13407 User centred design process for interactive systems.

Kirakowski J (1996) The software usability measurement inventory: background and usage. In: P Jordan, B Thomas, & B Weerdmeester, Usability Evaluation in Industry. Taylor & Frances, UK.

Macleod M, Bowden R and Bevan N. (in press) The MUSiC Performance Measurement Method. submitted for publication.

Macleod M. and Rengger R. (1993), “The Development of DRUM: A Software Tool for Video-assisted Usability Evaluation”. In: People and Computers VIII, Proceedings of the HCI’93 Conference (Loughborough, UK, Sept 1993), CUP.

Nasa-Ames Research Center, Human Performance Group (1986) Collecting NASA Workload Ratings: A Paper-

and-Pencil Package. Moffet Field, CA: NASA-Ames Research Center

Thomas C and Bevan N, eds (1995) "Usability Context Analysis Guide V 4". National Physical Laboratory, Teddington, UK.

Zijlstra, F.R.H. (1993) Efficiency in Work Behaviour: a Design Approach for Modern Tools. Delft: Delft University Press. Work Behaviour: a Design Approach for Modern Tools. Delft: Delft University Press.