Dublin Institute of Technology
ARROW@DIT

Conference papers School of Computing

1998-01-01

Usable software and its attributes: a synthesis of software quality: European Community law and human-computer interaction.

Ronan Fitzpatrick
Dublin Institute of Technology, ronan.fitzpatrick@comp.dit.ie

Catherine Higgins
Dublin Institute of Technology

Follow this and additional works at: http://arrow.dit.ie/scschcomcon

Part of the Computer Engineering Commons

Recommended Citation

Fitzpatrick, Ronan and Higgins, Catherine: Usable software and its attributes: a synthesis of software quality: European Community law and human-computer interaction. People and Computers XIII. Proceedings of the HCI'98 Conference published by Springer, London, United Kingdom.

This Article is brought to you for free and open access by the School of Computing at ARROW@DIT. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@DIT. For more information, please contact yvonne.desmond@dit.ie, arrow.admin@dit.ie, brian.widdis@dit.ie.

Usable Software and its Attributes: A synthesis of Software Quality, European Community Law and Human-Computer Interaction

Ronan Fitzpatrick

Department of Mathematics, Statistics and Computer Science,
Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland.
Tel: *+353 (1) 4024610*
Fax: *+353 (1) 4024994*
Email: *rfitzpatrick@maths.kst.dit.ie*

Catherine Higgins

Department of Mathematics, Statistics and Computer Science,
Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland.
Tel: *+353 (1) 4024834*
Fax: *+353 (1) 4024994*
Email: *chiggins@maths.kst.dit.ie*

Strategic managers and IS professionals who are responsible for specifying, acquiring and producing quality software products are not supported by the endless flow of new international standards, legislation and user requirements. In order to clarify the current situation for everybody concerned with software quality, and especially those interested in usability, there is a need for a new review and evaluation of the various strands that contribute to software quality. By way of review this paper recalls the original software quality factors which were defined twenty years ago by McCall *et al.* and presents a methodical analysis and synthesis of three modern strands which influence these factors. The three strands relate to software quality, statutory obligations and human-computer interaction. All three strands rely on well respected sources which include the European Council Directive on minimum safety and health requirements for work with display screen equipment, ISO/DIS 9241-10 (1993) and ISO/FDIS 9000-3 (1997). This synthesis produces a new set of quality factors, and the paper provides a new perspective of software usability by showing that the external quality factors in this new set are the usability attributes of a software product. New attributes like *suitability, adaptability, functionality, installability* and *safety* are identified and other attributes like *usability* and *integrity* are clarified within the three strands.

Keywords: software quality, European law, international standards, HCI, quality factors, usability, usability models and usability attributes.

1. Introduction

Software usability is described in terms of attributes of a software product, the measures that should be measured for those attributes, and metrics (numbers) which are the results of measurement (Holcomb and Tharp, 1991; Preece *et al*. 1994; ISO/IEC 9126, 1991; ISO/DIS 9241-11, 1995). So, in order to measure usability it is necessary to know what attributes must be measured, the type of measures that must be considered and what metrics to expect. The aim of this paper is to identify the attributes which should be measured. The paper begins with a review of formal definitions associated with software usability (McCall *et al*. 1977; Ravden and Johnson, 1989; ISO/DIS 9241-11, 1995). The reason for this review is to show how computer scientists' view of usability has changed over the past twenty years and to show that there are problems associated with these definitions. The paper continues with an examination of three strands which identify the set of attributes which are used to measure software usability. The strands are the software quality strand, the statutory obligations strand and the human-computer interaction strand. Section 2 introduces some facets of usability and explains the motivation for the three strand approach. Section 3 reviews usability definitions and explains problems associated with them. Section 4 examines the three strands in detail and identifies the usability attributes of a software product.

2. Facets of Software Usability

Usability is a key component in the overall quality of a software product (Porteous *et al.* 1993) which is concerned with making the product easy to learn and [easy to] operate (McCall *et al.* 1977). Usability also has a legal dimension. There are legal obligations for employers to protect the health of employees who use software interfaces (Council Directive 90/270/EEC, 1990). Usability is also a key concept of human-computer interaction (HCI), where, in addition to being concerned with making systems easy to learn and easy to use (Preece *et al.* 1994), it is also concerned with supporting users during their interactions with computers (Shneiderman, 1987). So, usability is a desirable feature that threads its way into different facets (quality, legal and HCI) of computer software. Collectively, these three facets are of interest to quality assurance managers, system designers, system developers, end-users and to those with organisational responsibility for selecting and acquiring usable systems (Reiterer and Oppermann, 1993; Robson, 1994). In its simplest form, I would describe usability as the extent to which a computer system interface supports end-users. Because there are many facets to usability and in order to fully understand what usability is (so that it can be specified and measured), it is necessary to first establish a comprehensive set of attributes that make up usability. In this paper, usability is considered to be an all embracing description of software. So, the attributes that make up usability can also be termed as the attributes of a usable software product.

There are many different definitions and models which clarify the meaning of software usability (McCall *et al.* 1977; Ravden and Johnson, 1989; ISO/IEC 9126, 1991; ISO/DIS 9241-11, 1995; Nielsen, 1993; Bevan and Macleod, 1994). Some of these, e.g. (ISO/IEC 9126, 1991; Nielsen, 1993) concentrate on the attributes that constitute usability while other definitions concentrate on how usability should be measured, e.g. (ISO/DIS 9241-11, 1995; Bevan and Macleod, 1994). However, while these definitions support our understanding of software usability, there are problems associated with them. For example, the definitions that focus on attributes are weak in their support for measures and visa versa.

There is a natural relationship between usability and a quality software interface and it follows that an interface that has a high level of quality will have a high level of usability (Ince, 1994). Consequently, the attributes that influence usability can be viewed as being quality factors. This paper establishes the quality factors that influence usability by reviewing three strands, each of which contributes different quality factors. These strands are reviewed using a quality-focused philosophy and are called the software quality strand, the statutory obligations strand and the human-computer interaction strand.

The software quality strand reviews quality models (McCall *et al.* 1977; Boëhm, 1978) and international standards (ISO 9000-3, 1991; ISO/DIS 9000-3, 1996; (ISO/IEC 9126, 1991; IEEE, 1989) which relate to software quality. The statutory obligations strand addresses the legislation enacted throughout the European Community (Council Directive 90/270/EEC, 1990) which requires that software should be easy to use and easy to adapt. This legislation also sets minimum requirements for the equipment that should be available to users and for the environment in which the users must work. The human-computer interaction strand examines current principles and practice in order to establish the usability requirements of end-users (Shneiderman, 1987).

The motivation for this three strand approach is the growing strategic need within business organisations for quality interfaces, which comply with current legislation and which support end-user employees (Reiterer and Oppermann, 1993; Robson, 1994). Only by combining the three strands is it possible to identify a comprehensive set of quality-focused attributes that influence usability. The presence or absence of these attributes is what is measured during usability measurement (Reiterer and Oppermann, 1993).

Before reviewing the three strands, it is first necessary to examine definitions and models of usability that are used in the software industry and in academia.

3. Definitions and Models of Usability

In this section, four definitions of usability are reviewed to show how computer scientists' views of usability have changed with advances in technology. Academic and commercial models are reviewed, and problems associated with these definitions and models are examined.

3.1 Definitions of Usability

Usability as a software quality factor was defined by McCall *et al.* (1977, p. 3-5) as *"the effort required to learn, operate, prepare input and interpret output of a program."*

To gain a proper understanding of McCall's perspective of usability in 1977, it is appropriate to recall the taxonomy of computers in those days. The environment consisted of mainframe and mini computers running major data processing applications. Staff were simply required to learn how to operate the system, input data, receive output and keep the system running. Software was developed for low specification monitors that used simple combinations of green and black text. Usability was perceived to be confined to operators and their learning process in this environment. The era of end-user computing was only beginning. More recently, Ravden and Johnson (1989, p. 9) defined usability as *"the extent to which an end-user is able to carry out required tasks successfully, and without difficulty, using the computer application system."*

From this definition comes some idea of the complexity of usability, especially considering that there are many different

- profiles of end-users
- skills among end-users
- attitudes among end-users
- complexities of tasks
- measures for success
- interpretations of difficulty.

To these can be added the different equipment that users need and the different environments in which users can work (Council Directive 90/270/EEC, 1990). An important advance in Ravden and Johnson's (1989) definition is that they introduced an element of measure by using the expression "the extent" in their definition.

The International Organisation for Standardisation (ISO) also define usability. In their standard (ISO/IEC 9126, 1991), usability is defined as *"a set of attributes of software which bears on the effort needed for use on an individual assessment of such use by a stated or implied set of users".*

This definition adds to our understanding of usability by considering a set of attributes of software. The standard names three attributes (which it calls sub-characteristics). These are learnability, understandability and operability.

The element of measure is also contained in a new International Standard (ISO/DIS 9241, 1995), which is currently under development. The standard is named "Ergonomic requirements for office work with visual display terminals (VDTs)" and consists of 17 parts. Part 11 (eleven) is specifically concerned with usability and defines it as "*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*".

Source	Definition
McCall *et al.*	The effort required to learn, operate, prepare input and interpret output of a program.
Ravden and Johnson	The extent to which an end-user is able to carry out required tasks successfully, and without difficulty using the computer application system.
ISO/IEC 9126:1991	A set of attributes of software which bear on the effort needed for use and on the individual assessment of such use by a stated or implied set of users.
ISO/DIS 9241-11:1995	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.

Figure 1. *Definitions of usability.*

3.2 Usability Models

Nielsen (1993) explains usability as part of the wider aspect of system acceptability and suggests that usability is part of a much broader scene - see figure 2.

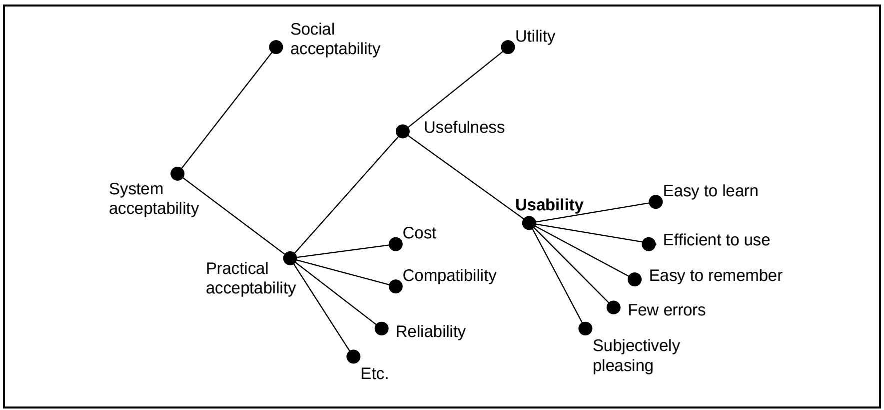

Figure 2. *Nielsen's model of system acceptability.*

Nielsen's approach focuses on social acceptability and practical acceptability. In his text, the concept of social acceptability is not developed to any great extent other than in the context of an example of a possible undesirable system. Systems with cultural influences or subliminal practices would be appropriate for consideration under social acceptability.

The sub-characteristics of practical acceptability of Nielsen's model are not unlike the technical or internal quality characteristics of a software product (Ghezzi *et al.* 1991; (Ince, 1994; ISO/DIS 9000-3, 1996). For example, compatibility is the same as interoperability. Perhaps, to correspond with the Etc. in the model, the author intended that practical acceptability should refer to factors like efficiency, portability, testability, maintainability and reusability as these are the factors that are not mentioned elsewhere in the model. Nielsen's model (1993, p. 25) sub-divides usefulness into utility and usability which are described respectively as: "*the question of whether the functionality of the system in principle can do what is needed [and] the question of how well users can use that functionality*". The usability dimension of the model incorporates "easy to learn", "efficient to use", "easy to remember", "fewer errors" and "subjectively pleasing". All of these are familiar expressions easily associated with the definitions of usability in section 3.1. They are also similar to the external quality characteristics of a software product (Ghezzi *et al.* 1991; Ince, 1994; ISO/DIS 9000-3, 1996). Consequently there are close connections between usability and software quality. "Subjectively pleasing", however, is new and introduces a new view of usability where end-users' subjective evaluations of a system come into play. This approach is also considered by Kirakowski and Corbett (1993) and by Bevan and Macleod (1994).

Dr. Kirakowski of the Human Factors Research Group at University College Cork has conducted extensive research in this area. His work is based on subjective user evaluations and he has developed a method for measuring software usability. This method - Software Usability Measurement Inventory (SUMI) - measures five sub-scales, i.e. efficiency, affect, helpfulness, control and learnability (Kirakowski and Corbett, 1993). The method is an attitude-measuring questionnaire that is completed by end-users.

More recently, Bevan & Macleod (1994, p. 136) have suggested that usability has to be viewed in different ways for different purposes, focusing on one or more of the following complementary views.

> "***a. the product-centred view of usability****: that the usability of a product is the attributes of the product which contribute towards the quality-of-use.*
> ***b. the context-of-use view of usability****: that usability depends on the nature of the user, product, task and environment.*
> ***c. the quality-of-use view of usability****: that usability is the outcome of interaction and can be measured by the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments*".

Another approach to understanding software usability is to consider actual practice as conducted by industrial leaders like IBM, Apple, Hewlett Packard and Microsoft. Kan (1995) reports that *"IBM monitors the CUPRIMDSO satisfaction levels of its software* *products (capability [functionality], usability, performance, reliability, installability, maintainability, documentation, service and overall). Hewlett-Packard focuses on FURPS (functionality, usability, reliability, performance and serviceability)".*

3.3 *Problems with Usability Definitions and Models*

There are three problems associated with the above definitions of software usability. The first problem is that there is no consistent set of attributes of software. While ISO/IEC 9126 (1991) and Nielsen (1993) focus on the attributes of software, their listings of the attributes are different. They are also inconsistent with commercial practice. For example, ISO/IEC 9126 (1991) mentions usability and suggests that its sub-characteristics are learnability, understandability and operability while Nielsen (1993) considers the broader concepts of social acceptability and practical acceptability and lists, easy to learn, efficient to use, easy to remember, few errors and subjectively pleasing. Commercial organisations like IBM and Hewlett-Packard use similar listings (Kan, 1995), but, there is no consistent set. Furthermore, is it not possible to know if a composite list of these four would represents all of the attributes of a usable software product.

The second problem with usability definitions is that, while recent definitions and models concentrate on the need to measure usability and even state what the measures should be, the definitions do not support a universal set of measures. ISO/DIS 9241-11 (1995) and Bevan & Macleod (1994) favour effectiveness, efficiency and satisfaction as usability measures while IBM measure satisfaction only (Kan, 1995). Kirakowski's SUMI product measures efficiency, affect, helpfulness, control and learnability.

The third problem is that it is not clear whether each measure should be applied to all attributes or whether some measures only apply to a selected set.

Therefore, strategic managers who have responsibility for software products are not supported by all this confusion. What end-users want to use, and what strategic managers want to acquire are usable software products which comply with the latest legislation. Therefore, what is now required is a clear listing of the attributes of a usable software product together with the measures that should be applied to these attributes. Then, by applying the measures to the attributes it should be possible to establish a "usability quotient" for any software product. The remainder of this paper focuses on the first of these requirements and identifies these attributes using a three strand approach.

4. Three Strands that Influence Usability

To identify the attributes of a usable software product, three specific strands are examined. The first strand is the software quality strand, which in turn identifies the statutory obligations strand, which in turn identifies the human-computer interaction strand. From these strands, software quality factors are identified and a specific set of these factors (those that directly impact on the end-user) are shown to be the usability attributes of a software product.

4.1 *Software Quality Strand*

The first strand to be examined is concerned with software quality. Studies in this domain began in the 1970s when desirable features for inclusion in software products were quantified by authors like McCall *et al*. (1977) and Boëhm (1978) who both produced quality models. Later, the world-wide success of quality standards like ISO 9000 (1987), resulted in international standards for software quality (ISO 9000-3, 1991; IEEE, 1989).

4.1.1 Quality Models and Quality Factors

Software quality is defined by the Institute of Electrical and Electronics Engineers (IEEE, 1983) as *"the degree to which software possesses a desired combination of attributes"*.

These attributes are typically referred to as quality factors or quality characteristics and models for these were suggested in the late '70s by McCall *et al.* (1977) and Boëhm (1978). Such quality factors include software correctness, reliability, efficiency, integrity, usability, maintainability, testability, flexibility, portability, reusability and interoperability (or interface facility). For a full explanation and recent review of each of McCall's quality factors, the reader is referred to Wallmüller (1994). These factors can be conveniently categorised as external quality factors (mainly relating to HCI issues) and internal quality factors (which relate to the technical excellence of the product.

Ghezzi *et al.* (1991, p. 18) support this view by stating that "*In general, users of the software only care about the external qualities*".

Because external factors affect users, we refer to these as usability factors. So, McCall's quality factors can be sub-divided into external and internal factors. A sub-division of these factors (based on Wallmüller) is set out in figure 3.

Quality Factor	Category
integrity reliability usability correctness efficiency interoperability	External quality factors (i.e. Usability factors)
maintainability testability flexibility reusability portability	Internal quality factors

Figure 3. *Categorised quality factors.*

The reader's attention is drawn to the quality factor called "usability". McCall's definition of this factor is shown in figure 1. The focus of this paper is that the term usability is better used to describe the entire software product and that repeating the term to describe a quality factor is inappropriate. Accordingly, this quality factor will be renamed as learnability and ease-of-use thereby avoiding further confusion. Ghezzi *et al.* (1991) and Daily (1992) suggest that it is important to prioritise these factors. These authors argue that if a software product cannot be installed, then it cannot be launched and therefore cannot be used. As a result, the other quality factors cannot be considered. Daily (1992, p. 19) also addresses this issue and suggests that "*once the software is usable, correct and reliable then efficiency, compatibility [interoperability] and integrity can be considered in more detail*".

The concept of *installability* is new. It is favoured by IBM (Kan, 1995) and supported by Ghezzi *et al.* (1991) and Daily (1992). So, this is the first new quality factor which must be added to McCall's original list.

In the intervening years since 1977/78 (when quality models were first published), there has been enormous technological advances and it is necessary to rethink and revise this area accordingly. Reference to Wallmüller (1994) shows that during this time some of the quality factors have become outdated. The remainder of this paper shows that some quality factors need to be renamed to reflect modern vocabulary and understanding. It also shows that new factors need to be added to McCall's list.

4.1.2 International Quality Standards

The systems professional who has the responsibility for selecting and acquiring quality software might, with good reason, look to the international standard relating to quality as a starting point to provide guidance on the best approach to adopt. In 1991, the International Organisation for Standardisation published "Guidelines for the application of ISO 9001 to the development, supply and maintenance of software" (ISO 9000-3, 1991). It is reasonable to expect that this international standard would address quality factors using the same vocabulary and meaning as used by McCall *et al.* (1977) and by Boëhm (1978). Unfortunately, this was not the case and consequently a new Draft International Standard (ISO/DIS 9000-3), was introduced and approved in June 1996. This became a Final Draft International Standard in 1997. This Final Draft (ISO/FDIS 9000-3) goes a long way towards resolving the deficiencies of its predecessor. For example, section 4.4.4 (p. 11) of the Final Draft uses language which is in keeping with established quality models. It reads *"The requirements may include, but not be limited to the following characteristics: functionality, reliability, usability, efficiency, maintainability and portability (see ISO/IEC 9126). Sub- characteristics, may be specified, for example security. Safety considerations and statutory obligations may also be specified.*

If the software product needs to interface with other software or hardware products, the interfaces between the software product to be developed and other software or hardware products should be specified ...".

So, using familiar terminology, the proposed standard is now recognising six quality factors by name. A seventh factor, *interface facility* (interfaces between software products) is covered by the second paragraph. Elsewhere in the document (clause 4.10) *testability* is mentioned. *Functionality* is a new quality factor, so, it is the second addition to McCall's list. The inclusion of *security, safety* and *statutory obligations* are welcome additions to the draft.

Security in the draft standard appears to relate to integrity as stated by McCall *et al.* (1977) and as stated by Boëhm (1978). McCall *et al.* describe it as being concerned with putting into place controls which guard against programs and data being incorrectly altered either by accident or by design. As an external quality factor, it supports user confidence in the software. To comply with the quality focus of this paper the term *security* (as opposed to integrity) will be used.

There are two aspects to *safety*. First there is the issue of operator safety which is covered by law and will be addressed in section 4.3 under the statutory obligations strand. The second aspect of safety is the safety of the general public. This is a special application of software for which designers and specifiers of safety-critical systems need to specify.

To reflect McCall's vocabulary, the *interface facility* as mentioned in the second paragraph of section 4.4.4 can be renamed *interoperability* (Ince, 1994).

However, the most significant aspect of clause 4.4.4 of the draft international standard is the inclusion of the expression "statutory obligations". This immediately brings into play all statutory regulations relating to health and safety issues including those relating to the minimum safety and health requirements for work with display screen equipment (Council Directive, 1990). This is the justification for the second strand in this three strand approach. This Council Directive is described in section 4.2.

So, from this review of the software quality strand, *installability*, *functionality* and *safety* are new factors to be added to McCall's list. McCall's integrity needs to be renamed as *security* and interface facility (in the standard) needs to be renamed *interoperability*.

That concludes the review of the software quality strand - the first of the three strands being considered in this paper. The second strand - statutory obligations - is examined in detail in the next section to identify additional factors that influence usability.

4.2 Statutory Obligations Strand

The second strand that impacts on usability is legislation and in keeping with clause 4.4.4 of ISO/DIS 9000-3 (1996), this strand is addressed as statutory obligations.

Statutory obligations are concerned with regulations which relate to health and safety issues but particularly those relating to the minimum safety and health requirements for work with display screen equipment. These obligations are outlined in a European Directive (Council Directive, 1990). National Governments also legislate for the safety and health of workers in their own countries. Both the

Directive and general regulations relating to end-users health and safety are now explained.

4.2.1 European Display Screen Directive

On the 29th May 1990, the Council for the European Communities published a directive relating to minimum safety and health requirements for those working with display screens (Council Directive, 1990). This directive became fully effective from 31 December 1996.

The directive sets out the employer's obligations and the employee's entitlements in relation to matters like:

- Analysis of workstations to ensure compliance with the directive.
- Training of employees.
- Employees daily work routine.
- The need for employee consultation and participation.
- Procedures for the protection of worker's eyes and eyesight.

The definition of a workstation (given by the directive) clearly includes software, so employers, as part of their analysis, training and consultation procedures must take cognisance of current best practice in human-computer interaction. This is further stated in the annex of the directive which sets out the minimum requirements under the heading Operator/Computer Interface. The five principles set out in part 3 of the Council Directive (1990) are relevant to usability and external software quality, and are set out in figure 4. Closer examination reveals that they are quality factors. "Suitable for the task" is easily expressed as *suitability*, "easy to use" could be usability (but to avoid adding further confusion it will be referred to here as *ease-of-use*). And the third principle, "adaptable to the operator's level of knowledge" is *adaptability*. Feedback, format & pace and software ergonomics all correspond to the golden rules for dialogue design (Shneiderman, 1987). These rules are an essential component of human-computer interaction and are the justification for the third strand and are explained in section 4.3.

So, from this strand three more quality factors can be added to those identified in the earlier strands. These new factors are *suitability*, *ease-of-use* and *adaptability* and have been derived from European Council Directive (1990). The full list of new software quality factors that have been identified, so far, are *installability, functionality, safety, suitability, ease-of-use* and *adaptability*. Later, these will be combined with McCall's list to create an updated set of quality factors. At that stage, two factors (integrity and interface facility) will be renamed as *security* and *interoperability* respectively.

That concludes the examination of the second strand and in the next section the human-computer interaction strand will be examined for further quality factors.

3. OPERATOR/COMPUTER INTERFACE
In designing, selecting, commissioning and modifying software, and in designing tasks using display screen equipment, the employer shall take into account the following principles:

a. software must be suitable for the task;
b. software must be easy to use and, where appropriate, adaptable to the operator's level of knowledge or experience; no quantitative or qualitative checking facility may be used without the knowledge of the workers;
c. systems must provide feedback to workers on their performance;
d. systems must display information in a format and at a pace which are adapted to operators;
e. the principles of software ergonomics must be applied, in particular to human data processing.

Figure 4. *European Council Directive 90/270/EEC 1990*
Summary of minimum safety and health requirements.

4.3 Human-Computer Interaction Strand

Human-computer interaction is described as "*the study of people, computer technology and the way these influence each other*" (Dix *et al.* 1993). It is the third strand to be examined in this paper. Authors in this domain (Shneiderman, 1987; Dix *et al.* 1993; Preece *et al.* 1994) address these topics in three categories. These categories are human issues, technology issues and interaction issues and they are described in the following sub-sections under the headings Human dimensions in HCI, The computer's capabilities in HCI and Users interacting with systems.

4.3.1 Human Dimensions in HCI

Issues that contribute to effective human usage of computers are well defined as part of the science of human psychology. The issues involved are human behaviour, human memory, ability to learn, human knowledge acquisition, cognitive issues, human perception of the working of the system and how these workings are best conceptualised (Shneiderman, 1987). Other issues that must be considered relate to the profile of the user and include the user's physical abilities and motor skills, previous knowledge or expertise in the domain, general education and training and the overall attitude of the user towards technology (see ISO/DIS 9241-11, 1995, for a full listing of user considerations). Through study and understanding of these issues, HCI professionals can specify and design interfaces that support these human factors. A useful way of illustrating these issues is to review some examples of their practical implementation. For example, the Internet might support users wishing to search for books in a second-hand bookshop. Typical users will want to browse many different departments searching for items that appeal to them. The interface style (metaphor) selected to present such an application on screen might include a series of floor plans with departments like local history, early printed and antiquarian books, maps, historical documents, prints, military, nautical history and similar divisions. Simple pointing and clicking accesses the preferred department for browsing, so there is no relearning expected of the user. All of the departments use names familiar to the user. Furthermore, the software applications developed to support these users are popularly referred to as browsers. So it is easy for the non-technical user to have the impression of browsing through the familiar departments of a bookshop. The system has become transparent to the user. Another example of a practical implementation of a software interface assisting human factors might be where interface designers support coherence by grouping similar tasks together. Similar design strategies can be used to best match other human factors with modes of interaction and other computer capabilities.

4.3.2 The Computer's Capabilities in HCI

The main technological focus of HCI is concerned with devices for human interaction with computers. Generally the devices used reflect the preferred dialogue style. A dialogue style is one of a number of methods by which users interact with the system (Shneiderman, 1987). These methods have evolved from command line solutions in the early days of computing to the hands-free, voice recognition systems which are becoming available. The most common dialogue styles are, command line interaction, batch programs, form filling, menu selection, query language, voice recognition WIMPS (Wigits, Icons, Menus and Pointers) and hyperlinks. Another term, WIRPS (Wireless, Intelligent, Remote, Probes and Sensors), can be used to describe the dialogue style of hostile environments. The evolution of these styles has been driven by a desire to improve the overall usability of the interface. For example, command line interfaces normally use keyboards as the input device while voice communication requires microphones. An excellent review of input and output devices is given by Preece *et al.* (1994).

Achieving the objectives of HCI is enhanced by the proper alignment of the input/output devices, both with the tasks to be completed and with the skills of the users. For example, secretaries with keyboarding skills are obviously more effective using a keyboard for word processing tasks while supermarket checkout operators are obviously more effective using a barcode scanner as their input device. Voice recognition and gesture recognition also enable easier interaction by users with differing skills. Different types of devices are needed for different environments. Office, home and educational environments are generally regarded as safe environments. Workshop floor and engineering plants are described as harsh environments while underwater and radioactive environments are hazardous environments. All three environments have very different device requirements.

4.3.3 Users Interacting with Systems

There are two distinct topics of interest in this area which affect end-users and which will be used to identify further characteristics that impact on usability. These are the established good principles and guidelines for dialogue design and the equipment and environment available to the user.

4.3.4 Principles and Guidelines for Dialogue Design

Principles and guidelines for dialogue design have been suggested for interface evaluation (Shneiderman, 1987; Ravden and Johnson, 1989). Naturally these same principles and guidelines can also be used for specifying the requirements for interface design. Typically, these principles address:

- Consistency of screen presentation
- Visual clarity on screen
- Informative feedback to users
- Compatibility with user conventions and expectations
- Error prevention and correction
- Appropriate functionality
- User control, confidence and satisfaction
- General user support

Dialogue principles are currently being addressed by the proposed international standard for ergonomic requirements for office work with visual display terminals (ISO/DIS 9241-10, 1993). The proposed standard addresses some of the issues covered by the above list together with some familiar quality factors which were identified in earlier strands. The seven dialogue principles in the standard and how they might be expressed as quality factors are:

• Suitability for task	Suitability
• Self descriptiveness	Usability (Ease-of-use)
• Controllability	Usability (Ease-of-use)
• Conformity with user expectations	Usability (Ease-of-use)
• Error tolerance	Security
• Suitability for individualisation	Adaptability
• Suitability for learning	Learnability

Self descriptiveness is the standard's terminology for informative user feedback, controllability relates to user control/user pacing of the use of the product and conformity with user expectation addresses compatibility with user conventions. User feedback, user control/pacing and compatibility with user conventions are all part of Shneiderman's golden rules which were the issues in section 4.2. that justified the human-computer interaction strand.

4.3.5 Equipment and Environment

User productivity, confidence and satisfaction are all supported by the proper equipment to perform the tasks and by a proper environment in which to work (Preece, 1994). So, HCI specialists are particularly interested in ensuring that these two issues are also addressed. The Council Directive (1990) has focused on this aspect and has set out a full schedule of minimum requirements. See figure 4.

Associated with the equipment and the environment is the health and safety of users. The research literature in the field of ergonomics shows considerable concern for a vast array of human disorders and explains how to design interaction in order to best prevent them. These include musculoskeletal disorders like Repetitive Strain Injury (RSI), Work Related Upper Limb Disorders (WRULDs), radiation emissions and facial rash (Euro Review, 1994; DST, 1997).

Combining all of the above topics, it is easy to see how human-computer interaction is concerned with the broad range of issues which contribute to the development of usable systems interfaces. The proper combination of all the topics, (i.e. human issues, technology issues and interaction issues), make the computer operator's role easier to perform, less prone to error, less anxious, builds confidence and many other psychological considerations that impact on computer users (Shneiderman, 1987; Reiterer and Oppermann, 1993).

Central to these topics have been the disciplines of psychology and ergonomics, both of which have contributed to defining best practice to support those who interact with computers. The overall aim of HCI should be to devise usable interfaces that employ the most suitable metaphor and then layout the screen so that human memory, coherence, cognition, perception, learning and previous knowledge are all supported to maximum effect. Interfaces should be designed to be as adaptable as possible in order to better support all end-user skills. Finally, the environment should be made as safe and comfortable as possible using selected devices which best suit the tasks to be performed. The ultimate objective is to create interfaces that are totally transparent to the users.

Like the quality strand and the statutory obligations strand, the human-computer interaction strand also identifies quality factors. These factors include *suitability*, *usability (ease-of-use), security*, *adaptability* and *learnability* (see - Principles and guidelines for dialogue design). *Learnability* is a new factor and must be added to McCall's list. This strand also identifies the needs of different users, particularly their needs in different environments using equipment appropriate to that environment. This in turn has given rise to the study of the context of use. Furthermore, the human-computer interaction strand provides a series of checklists and guidelines which combine current best practice for interface development.

This concludes the review of the three strands that identify quality factors. Currently, as three separate strands, their scope is very broad with considerable duplication. For the benefit of systems professionals, one composite table that combines the different strands is needed. Such a table should reflect the changing significance of the original quality factors suggested by McCall, the guidelines offered by ISO/DIS 9000-3 (1996), the statutory obligations resulting from Council Directive (1990) and HCI developments (ISO/DIS 9241-10, 1993). Such a table is shown in figure 5. To prepare this table, McCall's model is used as a foundation and it is sub-divided to show external and internal quality factors. Simple priority is also incorporated in the external factors. From figure 3, *reliability, correctness* and *efficiency* are all included together with the internal quality factors. Note that usability is not included at this stage and is replaced by *ease-of-use*. From the software quality strand, *installability, functionality* and *safety* are included. Integrity is renamed as *security* in order to better reflect the wording of ISO/DIS 9000-3 (1996) and interface facility (in ISO/DIS 9000-3, 1996) is renamed as *interoperability* to reflect McCall's vocabulary.

To fulfil the ISO requirement that statutory obligations must be complied with, the items set out in sub-division Operator/Computer Interface of the Council Directive (1990) are also included in the table. These items are *suitability*, *ease-of-use* and *adaptability*. From the human-computer interaction strand, *learnability* is added.

It is now necessary to return to McCall's original definition of usability, i.e. easy to learn and operate. Both of these issues are now catered for as quality factors in their own right, i.e. *learnability* and *ease-of-use*, and, as both are included in the new list, usability from McCall's original list is obsolete and is omitted.

Quality Factor	Category	Strand
• suitability • installability • functionality • adaptability • ease-of-use • learnability • interoperability • reliability • safety • security • correctness • efficiency	External quality factors (i.e. Usability factors)	Statutory obligations Quality Quality Statutory obligations Statutory obligations HCI Original quality factor Original quality factor Quality Renamed quality factor Original quality factor Original quality factor
• maintainability • testability • flexibility • reusability • portability	Internal quality factors	All original quality factors as proposed by McCall *et al*.

Figure 5. *Software quality factors table.*

4.4 The Usability Attributes of a Software Product

In section 2 usability is described as "the extent to which a computer system interface supports end-users" and in section 4.1, it is explained that it is preferable to describe external quality factors as usability factors. To confirm that this preference is valid, it

Attribute	McCall *et al.*	Comments/Source
• suitability		To comply with EU law - Council Directive (1990)
• installability		To reflect commercial practice To comply with ISO/FDIS 9000-3 (1997)
• functionality		To comply with ISO/FDIS 9000-3 (1997)
• adaptability		To comply with EU law - Council Directive (1990)
• ease-of-use	usability	To comply with EU law - Council Directive (1990)
• learnability		To comply with ISO/DIS 9241-10 (1993)
• interoperability	interoperability	Original quality factor
• reliability	reliability	Original quality factor
• safety		To comply with ISO/FDIS 9000-3 (1997)
• security	integrity	To reflect the wording of ISO/FDIS 9000-3 (1997)
• correctness	correctness	Original quality factor
• efficiency	efficiency	Original quality factor

Figure 6. *Usability attributes of a software product.*

is only necessary to apply the following simple query to each quality factor. Does the individual quality factor support the end-user? If it does, then it is a usability attribute. Applying this technique, the software quality factors in figure 5 can be transposed to a list of usability attributes as set out in figure 6 and called the usability attributes of a software product (or the attributes of a usable software product).

The attributes set out in figure 6 are those that impact the end-user. They are external quality factors and include attributes which must be considered during software usability measurement and evaluation in order to comply with current ISO standards and European Community law.

5. Conclusion

The quality of user interfaces is a central part of software development, not least because of European Community Law. This paper explained how the study of software usability has advanced over the past twenty years by reviewing four formal usability definitions. This review showed that some of the definitions focus on software attributes while other definitions focus on usability measures. The paper showed that in order for management to assess usability there is a need for a consistent set of usability quality attributes.

The approach used to identify this set of attributes involves a methodical analysis of well regarded sources in order to establish academic thinking and commercial practice. The paper uses a quality-focused self-justifying synthesis of three strands and identifies a new critical set of quality factors. It is then shown that the external quality factors in this set are the "usability attributes of a software product".

Which of the three strands is the most important is an issue that might arise for strategic managers. Both quality and HCI issues are matters of organisational policy, which may be decided by management. But statutory obligations are part of European Community law and must be complied with.

Acknowledgements

Thanks are due to Dr. Jakob Nielsen (permission to reproduce his model), to Mr. Brian Abbot of the National Standards Authority of Ireland (making ISO Standards available) and to Professor Alan Dix of Staffordshire University for his support.

References

N Bevan AND M Macleod (1994) "Usability measurement in context", *Behaviour and Information Technology*, Taylor & Francis Ltd, Basingstoke, UK, Vol(1 & 2)

B Boëhm (1978) *Characteristics of software quality*, Vol 1 of TRW series on software technology, North-Holland, Amsterdam, Netherlands

Council Directive (90/270/EEC) "1990 Minimum safety and health requirements for work with display screen equipment", *Official journal of the European Communities*, pp. L 156/14-18

K Daily (1992) *Quality management for software*, NCC Blackwell Ltd., Oxford, UK

A Dix, J Finlay, G Abowd and R Beale (1993) *Human-computer interaction*, Prentice Hall, Hemel Hempstead, UK

Euro Review (1994) *Euro review on research in health and safety at work*, European Foundation for the improvement of living and working conditions, Shankill, Co. Dublin, Ireland

C Ghezzi, M Jazayeri and D Mandrioli (1991) *Fundamentals of software engineering*, Prentice Hall, New Jersey, USA

R Holcomb and A Tharp (1991) "Users, a software usability model and product evaluation", *Interacting with Computers*, Butterworth-Heinemann, Oxford, UK, Vol 3(2) pp. 155-166

IEEE (1983) *IEEE Standard glossary of software engineering terminology*, Institute of Electrical and Electronics Engineers, IEEE Std 729-1983

IEEE (1989) *IEEE Standard for software quality assurance plans*, Institute of Electrical and Electronics Engineers, IEEE Std 730-1989

D Ince (1994) *ISO 9001 and software quality assurance*, McGraw-Hill, UK

ISO 9000 (1987) *International Standard. Quality management and quality assurance standards*, International Organisation for Standardisation, Genève, Switzerland

ISO 9000-3 (1991) *International Standard. Quality management and quality assurance standards - Part 3:Guidelines for the application of ISO 9001 to the development, supply and maintenance of software*, International Organisation for Standardisation, Genève, Switzerland

ISO/DIS 9000-3 (1996) *Committee Draft International Standard. Quality management and quality assurance standards - part 3:Guidelines for the application of ISO 9001 to the design, development, supply, installation and maintenance of computer software*, International Organisation for Standardisation, Switzerland

ISO/FDIS 9000-3 (1997) *Final Draft International Standard. Quality management and quality assurance standards - part 3:Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software*, International Organisation for Standardisation, Genève, Switzerland

ISO/IEC 9126 (1991) *International Standard. Information technology - Software product evaluation - quality characteristics and guidelines for their use*, International Organisation for Standardisation, Genève, Switzerland

ISO/DIS 9241-11 (1995) *Draft International Standard. Ergonomic requirements for office work with visual display terminals (VDTs). Part 11:Guidance on usability*, International Organisation for standardisation, Genève, Switzerland

S Kan (1995) *Metrics and models in software quality engineering*, Addison-Wesley.

J Kirakowski and M Corbett (1993) "SUMI:the software usability measurement inventory", *British Journal of Educational Technology*, Vol 24(3), pp. 210-212

J A McCall, P K Richards and G F Walters (1977) *Factors in software quality*, Vols I-III, Rome Aid Defence Centre, Italy

J Nielsen (1993) *Usability engineering*, Academic Press Limited, London, UK

M Porteous, J Kirakowski AND M Corbett (1993) *SUMI users handbook*, Human Factors Research Group, University College, Cork, Ireland

J Preece, Y Rogers, H Sharpe, D Benyon, S Holland and T Carey (1994) *Human-computer interaction*, Addison-Wesley, Wokingham, UK

S Ravden and G Johnson (1989) *Evaluating usability of human computer interfaces:a practical method*, Ellis Horwood Ltd., Chichester, UK

H Reiterer and R Oppermann (1993) "Evaluation of user interfaces:EVADIS II - a comprehensive evaluation approach", *Behaviour and information technology*, Taylor and Francis, Basingstoke, UK, Vol 12(3), pp. 137-148

W Robson (1994) *Strategic management and information systems:an integrated approach*, Pitman Publishing, London, UK

B Shneiderman (1987) *Designing the user interface:strategies for effective human-computer interaction*, Addison-Wesley, USA

E Wallmüller (1994) *Software quality assurance:A quality approach*, Prentice-Hall International, Hertfordshire, UK