Software Quality Journal, 11, 325–338, 2003

Usability Meanings and Interpretations in ISO Standards

ALAIN ABRAN, ADEL KHELIFI and WITOLD SURYN
aabran@ele.etsmtl.ca; adel@khelifi.com; wsuryn@ele.etsmtl.ca
École de Technologie Supérieure, Montréal, Canada

AHMED SEFFAH seffah@cs.concordia.ca
Human-Centered Software Engineering Group, Concordia University, Montreal, Canada

Abstract. The usability of a software product has recently become a key software quality factor. The International Organization for Standardization (ISO) has developed a variety of models to specify and measure software usability but these individual models do not support all usability aspects. Furthermore, they are not yet well integrated into current software engineering practices and lack tool support. The aim of this research is to survey the actual representation (meanings and interpretations) of usability in ISO standards, indicate some of existing limitations and address them by proposing an enhanced, normative model for the evaluation of software usability.

Keywords: software usability, software measures, usability, ISO 9126, ISO 9241, software quality, usability models

1. Introduction

Within the last couple of years, a major shift has occurred in the development, design and deployment of software applications. With the considerable growth of distributed applications, it is to be noted that, especially for the Internet, the developers, technical staff and training instructors no longer have direct access to the end-user of their software systems. Software usability is no longer a luxury, but rather a basic determinant of productivity and of the acceptance of software applications. However, without specific knowledge about the end-user of the software systems, taking into account and implementing the usability and learnability of a software system becomes a major quality challenge in designing these systems.

This paper presents a survey and an evaluation of some of the ISO usability standards and related research in this area. We present an analysis of ISO software usability evaluation models, such as ISO 9241-11 and 9126, and a proposal for their integration into an enhanced model, which can be used as input to the next ISO review of these normative models.

In Section 2 of this paper, we present a survey of usability definitions. In Section 3, usability in ISO/IEC 9126 and ISO 9241 standards is discussed, highlighting both their strengths and their weaknesses. In Section 4, improvements are proposed, and these are incorporated in an enhanced model. Finally, further recommendations for improving the normative models are presented in Section 5.

2. Background

2.1. *Definitions of usability*

The term "usability" refers to a set of multiple concepts, such as execution time, performance, user satisfaction and ease of learning ("learnability"), taken together. But usability has not been defined homogeneously, either by the researchers or by the standardization bodies. Table 1 illustrates how the term has been defined in three distinct standards.

Standards related to usability can be classified in the following categories (Figure 1):

1. Product effect (output, effectiveness and satisfaction at the time of use of the product);
2. Product attributes (interface and interaction);
3. Process used to develop the product;
4. Organization's Capability.

The ISO has developed different standards on usability, and two major categories can be distinguished:

1. Product-oriented standards (ISO 9126, 2001; ISO 14598, 2001)
2. Process-oriented standards (ISO 9241, 1992/2001; ISO 13407, 1999)

Table 1. Usability definitions in Standards

Usability definitions
"The capability of the software product to be understood, learned, used and attractive to the user, when used under specified conditions." (ISO/IEC 9126-1, 2000)
"The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use." (ISO 9241-11, 1998)
"The ease with which a user can learn to operate, prepare inputs for, and interpret outputs of a system or component." (IEEE Std. 610.12-1990)

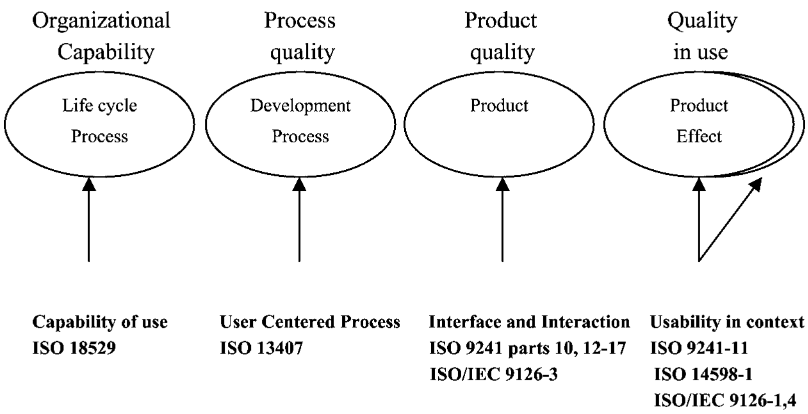

Figure 1. Categories of ISO usability standards.

It should be noted, too, that the properties that software usability takes on might vary, depending on the target audiences of the software system.

A candidate audience for a software system might include end-users, managers and software developers. For each of these audiences, usability is defined from a different viewpoint:

1. For the end-user, software usability is essential because it is a determinant of performance: an application, which features good usability, will allow the user to perform the expected task faster and more efficiently.
2. For managers, usability is a major decision point in selecting a product, as this decision will have a direct influence on the learnability of the chosen system, and hence on the productivity of those who use it.
3. For software developers, usability describes the internal attributes of a system, including issues like design quality, documentation maintainability.

This diversity of viewpoints and their related usability requirements have led to different perspectives on usability in the various ISO models that have been developed over the years by distinct groups of usability experts. Unfortunately, each group of experts built its model without input from the other groups. This led to the use of different terms and labels for the same usability characteristics, or different terms for similar characteristics, without full consistency across these standards; the situation in the literature in general is similar. For example, learnability is defined in ISO 9241-11 as a simple attribute, "time of learning," whereas ISO 9126 defines it as including several attributes such as "comprehensible input and output, instructions readiness, messages readiness . . ."

This illustrates that further work is required to bring convergence and a consensus on a more comprehensive view of usability, which would integrate all viewpoints into a unified model able to withstand the sophistication of the applications built for the user communities.

This paper focuses mainly on an analysis of software usability according to two ISO standards, ISO 9241 and ISO 9126. The strengths and weaknesses identified will then serve as a basis for the design of a consolidated and improved model of usability.

3. Representation of usability in ISO standards

3.1. ISO 9126

The ISO 9126 series of standards (ISO 9126, 2001–2003) addresses in its four parts software quality from the product perspective. Even though it is not exhaustive, this series constitutes the most extensive software quality model developed to date. The approach of its quality model, initially published in 1991 and refined over the next ten years by ISO's group of software engineering experts, is to represent quality as a whole set of characteristics. This international standard divides software quality into six general categories of characteristics: functionalities, reliability, usability, effectiveness, maintainability and portability (Figure 2).

The objective of this series of standards is to provide a framework for the evaluation of software quality. ISO/IEC 9126 does not prescribe specific quality requirements

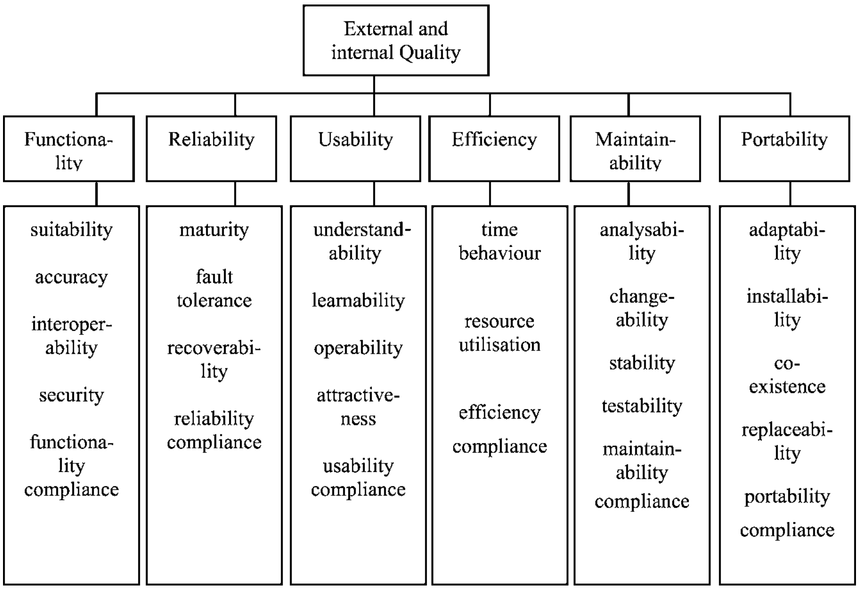

Figure 2. Software quality—ISO 9126.

for software, but rather defines a quality model, which can be applied to every kind of software. This ISO standard includes the user's view and introduces the concept of quality in use.

The 2001–2003 edition of ISO/IEC 9126 is divided into four parts:

1. ISO/IEC 9126-1: Software Engineering—Product quality—Part 1: Quality model.

This standard specifies two distinct structures of models for software quality:

(a) Internal and external quality is modeled with the same set of six characteristics: functionality, reliability, effectiveness, usability, maintainability and portability.
(b) Quality in use characteristics are modeled with four other characteristics: effectiveness, productivity, security and satisfaction.

2. ISO/IEC 9126-2: Software Engineering—Product quality—Part 2: External metrics (the term "metrics" used in ISO/IEC 9126 will be replaced by "measure" in future releases; therefore the term "measure" has been adopted in this paper).

This part describes the measures that can be used to specify or evaluate the behaviour of the software when operated by the user.

3. ISO/IEC 9126-3: Software Engineering—Product quality—Part 3: Internal metrics.

This part describes the measures that can be used to create the requirements that describe the static properties of the interface, which can be evaluated by inspection without operating the software.

4. ISO/IEC 9126-4: Software Engineering—Product quality—Part 4: Quality in use metrics.

This part describes the measures that can be used to specify or evaluate the impact of the use of the software when operated by the user.

3.2. Representaton of usability in ISO 9126

In 1991, ISO 9126 defined usability as "a set of attributes that bear on the effort needed for use and on the individual assessment of such use, by a stated or implied set of users." It then proposed a product-oriented usability approach. Usability was seen as an independent factor of software quality and it focused on software attributes, such as its interface, which make it easy to use (Bevan, 1997).

However, the attributes that a product requires for usability depend on the nature of the user, the task and the environment. In a product-oriented approach, usability is seen as a relatively independent contribution to software quality, as defined now in the 2001 edition of ISO/IEC 9126-1: "The capability of the software product to be understood, learned and liked by the user, when used under specified conditions."

Usable products can be designed by incorporating product characteristics and attributes, which are beneficial to users in particular contexts of use. Users are interpreted directly as interactive system users. They can include operators, as well as direct or indirect users who are influenced by or depend on using the software.

Usability specification and evaluation should address several user environments, which the software can affect, including both use preparation and results evaluation. Usability is defined in this international standard as a specific set of software attributes; it is different and defined from an ergonomic viewpoint with other characteristics such as output and effectiveness as usability components.

To specify software quality, a purchaser needs a model and analytical tools to communicate precisely his requirements concerning the product to be developed. Similarly, a software provider needs to be able to verify with confidence whether or not the product provides the expected level of software quality. This ISO 9126 standard can be used as a reference for contractual agreements between a purchaser and a software producer, and it can be used to eliminate a number of misunderstandings between purchaser and provider.

The principal advantage of a clearly defined and agreed upon model, supported with appropriate measures, is that it clarifies the definition of usability, and proposes measures to provide objective evidence of achievement. It is to be noted that, during the development process, including testing, measurements are usually collected in simulated environments in a laboratory, where industrial context may be lost.

The ISO/IEC 9126 quality model can be used to specify and verify those properties that the software must exhibit before being put into service. Some countries, such as Japan, have adopted it as a national standard.

However, there are still some weaknesses in ISO 9126 which have not yet been fully tackled, such as:

- Unclear architecture at the detail level of the measures.
- Some overlapping of concepts, making the standard challenging for the user community to grasp clearly, such as the usability characteristics of internal and external quality with respect to the quality-in-use set of quality characteristics.

- Lack of a quality requirements standard.
- Lack of guidance in assessing the results of measurement.
- Ambiguous choice of measures.

As well, the current ISO 9126 detailed description of its quality model is static: it does not describe the relationship between project phases and measures at subsequent project milestones. It is important to be able to relate software measures to project tracking and to target values at the time of delivery of the software. Also, it does not give any guidance as to the use of the measures and attributes in the identification and classification of risk (Hyatt and Rosenberg, 1996).

3.3. ISO 9241

ISO 9241 is a suite of international standards on ergonomics requirements for office work carried out using visual display terminals (Figure 3). The definitions of Part 11 of ISO 9241 are put together from a different usability viewpoint. Its key components are: effectiveness, which describes the interaction from a process viewpoint; efficiency, which pays attention to results and the resources involved; and satisfaction, which is a user viewpoint.

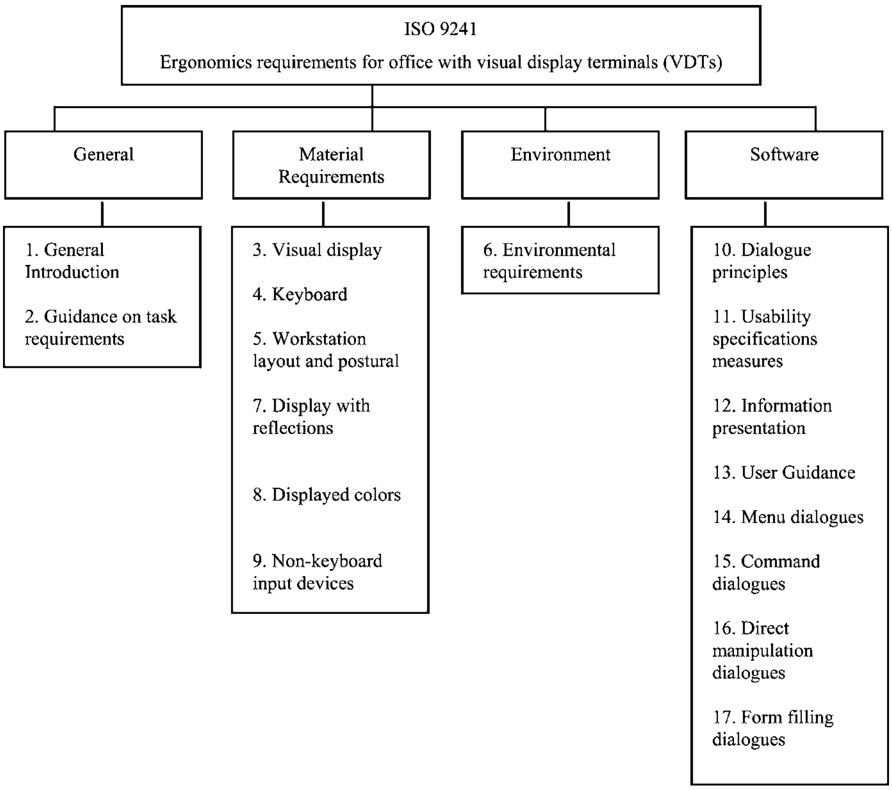

Figure 3. The 17 parts of ISO 9241.

ISO 9241 provides requirements and recommendations concerning hardware, software and environment attributes, which contribute to usability, and concerning subjacent ergonomic principles. Parts 3 to 9 deal with hardware design requirements and guidelines, which can have implications for software. Parts 10 to 17 deal with software attributes.

3.4. Representation of usability in ISO 9241

Part 11 of this standard explains how to identify the information that has to be taken into account when specifying, or evaluating, usability in terms of measures of user performance and satisfaction. Guidance is given on how to describe the context of use of the product and the measures of usability in an explicit way (Bevan and Schoeffel, 2001). In spite of the name, the definitions in part 11 are also known to be applicable to other situations where a user interacts with a product to achieve certain objectives. This extension makes usability a generic usability concept, likely applicable outside its conventional applications in information technology.

Standard ISO 9241 defines usability in the following way: *software is usable when it allows the user to execute his task effectively, efficiently and with satisfaction in the specified context of use*.

According to this standard, measurement of system usability consists of three usability attributes:

1. Effectiveness: How well do the users achieve their goals using the system?
2. Efficiency: What resources are consumed in order to achieve their goals?
3. Satisfaction: How do the users feel about their use of the system? (Wixon and Wilson, 1997)

This standard presents usability guidelines and is used for evaluating usability according to the context of use of the software.

ISO 9241-11 recommends a process-oriented approach for usability, by which the usable interactive system is achieved through a human-centered design process.

Usable products can be designed by incorporating characteristics and attributes in particular contexts of use. This process alone is not sufficient to ensure efficiency, effectiveness and satisfaction when using the product. To verify whether or not the required level of usability is achieved, it is necessary to measure the performance and the satisfaction of users working with the product. The measurement of usability is a complex interaction between users and context of use; this might produce different levels of usability performance for the same product when it is used in different contexts.

Context characteristics (users, tasks and environment) can be sufficient to determine usability as a product characteristic: a change in any relevant aspect of context of use can change product usability. For example, a user interface can be improved by conformance to good practices of dialog design, by ensuring harmony between user and system, through selection and user training or through good task design. A product, which is usable by qualified users, might not be usable by beginners. Work environment aspects, such as noise or office design, can also impact usability.

The approach adopted in Part 11 of ISO 9241 has the following advantages:

1. The ISO 9241-11 model identifies usability aspects and context-of-use components to be taken into consideration during specification, design and usability evaluation.
2. User performance and satisfaction provide direct measurements of usability in a particular context.
3. User performance and satisfaction measurements provide a basis for comparing usability with other design features for the same context.
4. Usability can be defined and verified within quality systems conforming to ISO 9001.

By contrast, this standard also has some weaknesses:

1. It addresses usability strictly from a process perspective, hence tackling only a single viewpoint.
2. ISO 9241-11 does not tackle the learnability characteristic as is recommended by the majority of standards and experts on usability.
3. It does not tackle the security aspects, considered to be very significant by domain experts.

In summary, it is not yet easy to exploit the 2002 normative models for usability evaluation, and users need a very clear demonstration of the application of these models to their software. Similarly, issues of usability at the architecture level are among the most difficult to rectify because changes at this level are inevitably relegated to the dialogue or to the presentation level.

4. An enhanced usability model

4.1. The baseline usability model

In this section, we present our suggestions for improving the current ISO usability models. The analysis of the ISO 9241 definitions indicates that this standard has a broader perspective of usability than ISO 9126: ISO 9241 focuses on tasks and environment questions as organizational factors, and its usability definition focuses on software quality characteristics which are distinct from those of usability in ISO 9126, such as functionality, precision and effectiveness. All these characteristics contribute to software quality.

The two viewpoints on usability are complementary: according to Bevan (Bevan and Schoeffel, 2001), the interactive system does not have intrinsic usability, only an ability to be used in a particular context of use. For example, 9241-11 can help us to understand in which context particular attributes specified in ISO 9126 are required.

We have therefore selected the basic ISO 9241-11 architecture as the baseline for our consolidated model, and our selection criteria for this decision are listed in Table 2.

4.2. Additional usability characteristics

Some researchers have proposed their own usability model, through additional definitions or attributes of this concept, often including the *learnability* characteristic for

Table 2. ISO 9241

The only normative model that specifically addresses usability.
Adopted by experts in HCI (Human Computer Interaction). See, for example: (Jordan et al., 1996).
Explains how to identify information for specifying or evaluating usability. Guidance is provided to describe product context of use (hardware, software or service) and required measurements for usability in an explicit manner.
Considered by experts to be the instrument best suited to interpreting the VDTs directive. Also, accepted by the EU commission as a vehicle to demonstrate compliance with the directive. The standard is harmonized in Europe and published as a German DIN standard [6].
Defines usability on the basis of 3 characteristics: Effectiveness, efficiency and satisfaction.

Table 3. Learnability as a usability characteristic

Nielsen (1993) and Schneiderman (1998) characterized five attributes of usability: *learnability*, effectiveness, tolerance for errors, satisfaction and memorization.
Boehm et al. (1978) defined software usability as the extent to which the product is *convenient and practical to use*.
Jones (1997): Usability is the total effort required to *learn*, operate and use software or hardware.
Gilb (1996) identifies some measurable attributes of usability; one of them being *learnability*.
ISO/IEC 9126-1 defines usability as the capacity of the software product to be included/understood, *learned*, used and attractive to the user, when it is used under specified conditions. IEEE Std. 610.12 defines usability as the ease with which a user can *learn* how to operate, prepare inputs for, and interpret the outputs of, a system or component.
The MUSiC project identifies *learnability* as the attribute of usability, in Bevan et al. (1991).

usability (Table 3). Figures 4 and 5 present the usability models of two groups of authors who used the same set of usability characteristics, but who differ as to the levels of the proposed measures to use for these characteristics: Dix et al. (1993) and Nielsen (1994).

Other researchers, as well as other standards organizations, have identified other viewpoints on usability, and have included another usability characteristic, referred to as the *security* characteristic (Table 4). Table 4 lists some of the standards where security is included within their usability model.

These standards consider that good usability is a significant condition for human security in critical systems, such as medical apparatus or nuclear power stations.

4.3. Enhanced usability model

A more comprehensive model of usability should include both process-related and product-related usability characteristics. Having already selected ISO 9241-11 as our

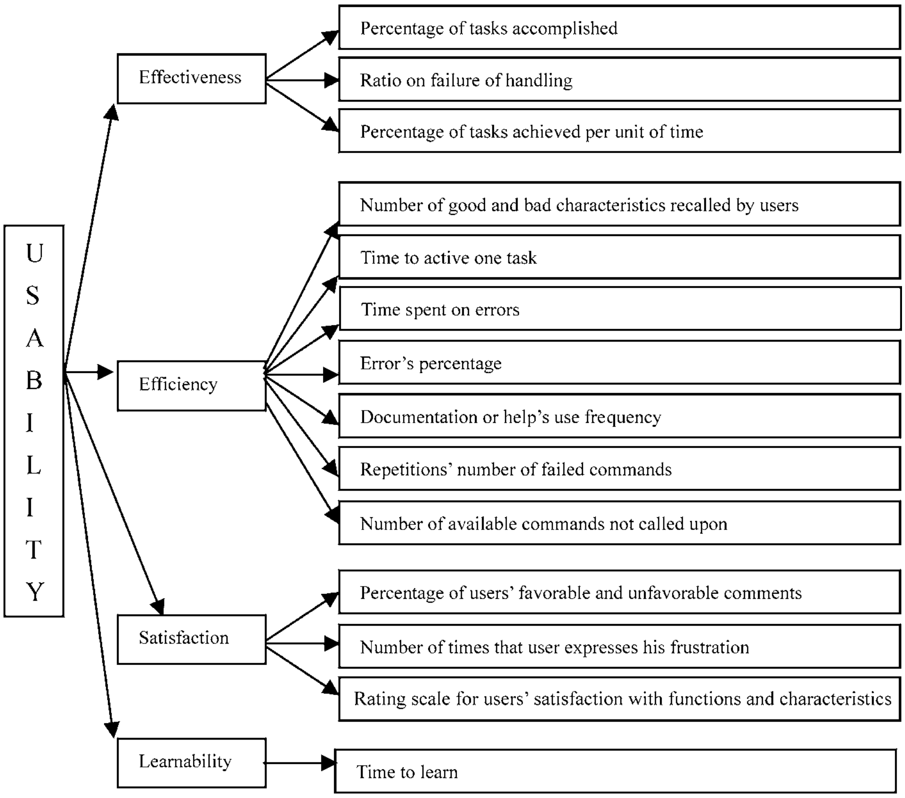

Figure 4. Usability model (Dix et al., 1993).

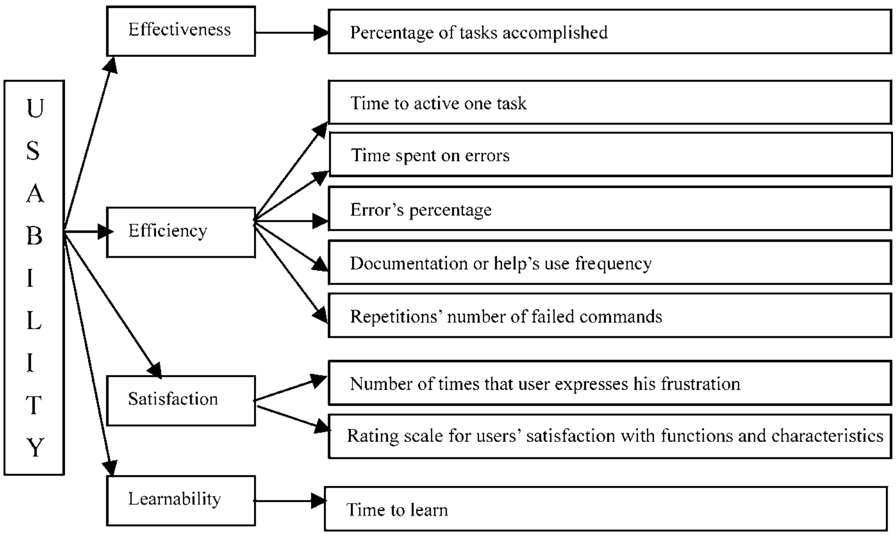

Figure 5. Usability model (Nielsen, 1994).

Table 4. Security as a usability characteristic

ITSEC: Information Technology *Security* Evaluation Criteria.
Standard IEC 300 presents software as *security*-critical.
ISO 13407 (1999) describes human-centered design as a multidisciplinary activity incorporating human factors and ergonomic and technical knowledge with the objective of raising efficiency and effectiveness, improving human working conditions, and opposing possible unfavorable effects of use on human health, *security* and performance.
ISO/IEC 9126 defines *security,* which is a sub-characteristic, as a set of software attributes which relates to its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data.
Security is a characteristic of the CHI, which is particularly important in an industrial context (FAA, 1998).

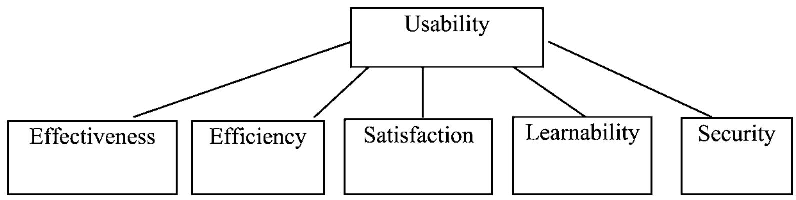

Figure 6. Enhanced Usability Model.

baseline for an enhanced model, we must then integrate into this baseline the other relevant usability characteristics from both 9126 and other sources, such as learnability and security.

This enhanced model, referred to as the Enhanced Usability Model is presented in Figure 6.

The three-layer structure of ISO 9126 (characteristics, sub-characteristics and measures) was then used to complete the Enhanced Model of Usability. In particular, relevant candidate measures proposed by authors have been analyzed, and a recommended set chosen. The resulting usability model, with specific measurement proposals, is presented in Figure 7.

5. Conclusion

In this paper we summarized our investigations on various models for software usability measurement, in particular the ISO 9126 and 9241 standards. The fact that these distinct ISO standards dealing with software usability were not designed from the same perspective was highlighted. Similarly, it was noted that definitions given by experts and researchers have not yet been harmonized. For instance, usability model ISO 9241-11 was developed by a group of experts specializing in Human–Computer Interaction, whereas the ISO 9126 model was developed by another group of experts specializing in software engineering. This illustrates that the models of usability measurement currently being proposed to the industry are not yet mature and that further

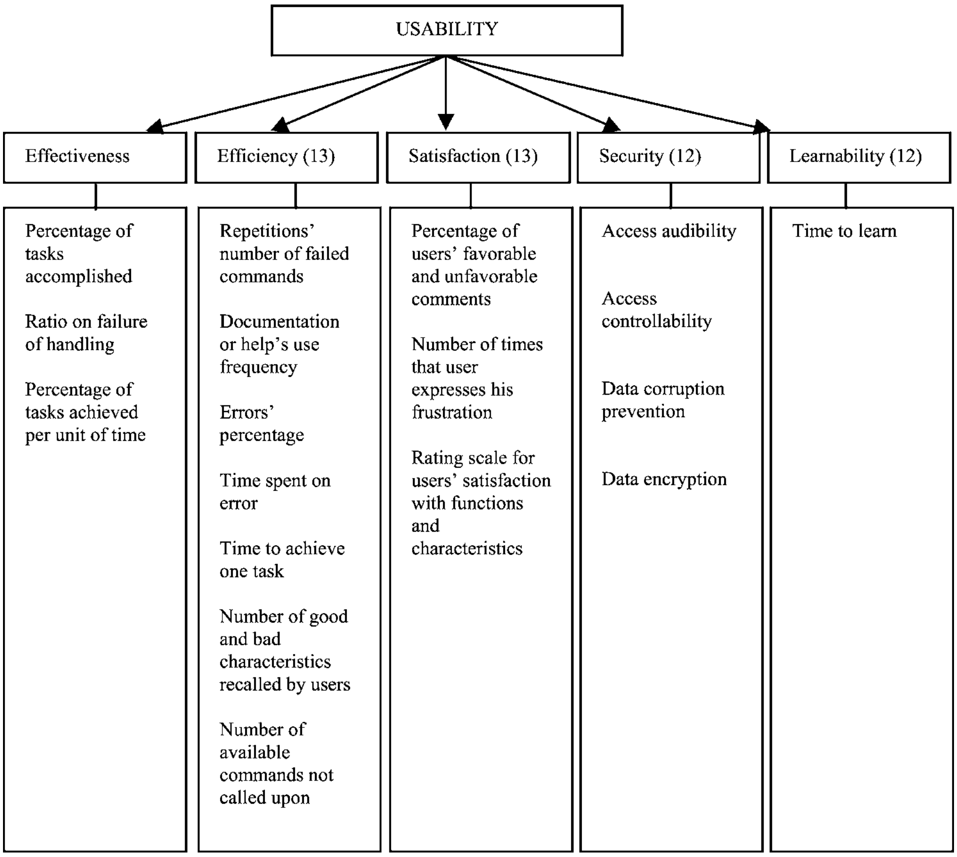

Figure 7. Consolidated and normative usability.

work is required to develop a greater consensus on an integrated and more comprehensive model.

Furthermore, these standards and methods equally depend on individuals involved in research and on technical issues. Software and usability professionals outside the standardization process may have strong and valuable understanding on usability measurement, but fail to use and apply the existing models because the standards are confusing and bureaucratic. It is unclear how all the existing lists, rules and criteria are related (if at all) and whether one list may be more useful than others. To understand the various checklists and relationships among all the factors that affect usability, the concept of usability needs to be broken down in a way that allows comparisons from both theoretical and practical viewpoints. An analysis of existing models and standards shows us that there is a need for a consistent and interpretive repository (or knowledge map) of Usability/Quality in Use measurement, which includes factors, criteria, measures descriptions and interpretations.

On the basis of the analysis of the strengths and weaknesses of both standards, as well as an investigation of other models proposed by various authors, a revised list for a usability measures was proposed, including the addition of two characteristics to the ISO 9241-11 standard which had been selected as our baseline: "learnability" and "security."

References

Bevan, N. 1995. Human–computer interaction standards, In *Proceedings of the 6th International Conference on Human Computer Interaction*, Yokohama, July, eds Anzai and Ogawa, Elsevier.

Bevan, N. 1997. ISO 9126, EAGLES evaluation group workshop, *Evaluation in Natural Language Engineering: Standards and Sharing*, Brussels, November 26th and 27th, http://www.cst.ku.dk/projects/eagles2/workshop/ISOnigel.html.

Bevan, N., Kirakowski, J. and Maissel, J. 1991. What is usability? *Proceedings of the 4th International Conference on HCI*, Stuttgart, September.

Bevan, N. and Schoeffel, R. 2001. A proposed standard for consumer product usability, UAHCI, New Orleans, August.

Boehm, B.W., Brown, J.R., Kaspar, J.R., Lipow, M., MacLeod, G.J. and Merrit, M.J. 1978. *Characteristics of Software Quality*, Amsterdam, North-Holland.

Dix, A., Finlay, J., Abowd, G. and Beale, R. 1993. *Human–Computer Interaction*, Englewood Cliffs, NJ, Prentice-Hall.

FAA. 1998. Standard Terminal Automation Replacement System, Human Factors Team Report of the Computer–Human Interface Re-Evaluation, 23 February.

Gilb, T. 1996. Level 6, Why we can't get there from here, *IEEE Software* 13(1): 97–98.

Gore, A. 1998. A letter from Vice President Al Gore, *Common Ground* 8(3): 1.

Hyatt, L.E. and Rosenberg, L.H. 1996. A software quality model and metrics for identifying project risks and assessing software quality, *Software Product Assurance Workshop*, 19–21 March, ESTEC, Noordwijk, The Netherlands.

ISO 9241. 1992/2001. Ergonomics Requirements for Office with Visual Display Terminals (VDTs), Geneva, International Organization for Standardization.

ISO 13407. 1999. User-Centered Design of Interactive

ISO/IEC 9126. 2001. Software Product Evaluation—Quality Characteristics and Guidelines for the User, Geneva, International Organization for Standardization.

ISO/IEC 14598. 2001. Information Technology—Software Product Evaluation.

ISO TR 18529. 2000. Ergonomics of Human–System Interaction—Human-Centered Lifecycle Process Descriptions.

Jones, C. 1997. *Software Quality Analysis and Guidelines for Success*, International Thomson Computer Press, p. 492.

Jordan, P.W., Thomas, B., Weerdmeester, B.A. and McClellard, I.L. (eds) 1996. *Usability Evaluation in Industry*, London, Taylor & Francis.

KANBRIEF. 1998. Kommission Arbeitsschutz und Normung, March. http://www.kan.de/content/englishcontent/publications-kanbrief/brief983.htm.

Nielsen, J. 1993. *Usability Engineering*, London, Academic Press.

Nielsen, J. 1994. *Usability Engineering*, Boston, Academic Press.

Preece, J. 1994. *Human–Computer Interaction*, Addison-Wesley.

Schneiderman, B. 1998. *Designing the User Interface*, Addison Wesley.

Wixon, D. and Wilson, C. 1997. The usability engineering framework for product design and evaluation, In *Handbook of Human–Computer Interaction*, ed. M. Helander, Amsterdam, p. 665.

Alain Abran is a Professor and the Director of the Software Engineering Research Laboratory at the École de Technologie Supérieure (ETS)-Université du Québec. He is currently Co-Executive Editor of the Guide to the Software Engineering Body of Knowledge project. He is also actively involved in software engineering standards as the international secretary for ISO/IEC JTC1 SC7—Software and System Engineering; he is also Co-Chair of the Common Software Measurement International Consortium (COSMIC). Dr. Abran has more than 20 years of industry experience in information systems development and software engineering. Dr. Abran holds a Ph.D. in electrical and computer engineering (1994) from École Polytechnique de Montréal (Canada) and master degrees in management sciences (1974) and electrical engineering (1975) from University of Ottawa. His research interests include software productivity and estimation models, software engineering foundations, software quality, software functional size measurement, software risk management and software maintenance management.
URL: www.lrgl.uqam.ca

Adel Khelifi is a software engineering Ph.D. student at École de Technologie Supérieure, Montreal, Canada. He holds a master degree in management information system from University of Quebec in Montreal. Mr. Khelifi has over eight years of software experience, including services at Tunisia Ministry of Finances, Quebec Ministry of Immigration and at the United Nations Organization in Montreal. His research interests include functional size measurement, software quality and software usability. Mr. Khelifi is also involved in the on-going work of the ISO/IEC JTC 1/SC7 about software functional size.

Witold Suryn is a Professor at the École de Technologie Supérieure, Montreal, Canada (engineering school of the Université du Québec network of institutions) where he teaches graduate and undergraduate software engineering courses and conducts research in the domain of software quality engineering, software engineering body of knowledge and software engineering fundamental principles. Dr. Witold Suryn is also the Director of the Software Quality Engineering Research Laboratory (*I*QuaL) at the École de Technologie Supérieure and the Convener of ISO SC7 WG6 SG1 working group.

Ahmed Seffah is a Professor of Software and Usability Engineering at Concordia University. He is the Co-Founder of the Concordia Usability/Empirical Studies Laboratory and the Chair of the Human-Centered Software Engineering Group. Professor Seffah's interests are at the intersection of human–computer interaction and software engineering, with an emphasis on usability engineering, user interface design, human-centered software engineering, empirical studies, theoretical models for usability and quality in use measurement as well as patterns as a vehicle for capturing and incorporating human experiences in software engineering.