Specifying usability requirements and test criteria for interactive systems

Thomas Geis, Dr. Wolfgang Dzida, Wolfgang Redtenbacher

Research Report
Fb 1010

baua:
Bundesanstalt für Arbeitsschutz
und Arbeitsmedizin

**Series of publications of the
German Federal Institute for
Occupational Safety and Health (FIOSH)**

Research

German Federal Institute for Occupational Safety and Health (FIOSH)
[Bundesanstalt für Arbeitsschutz und Arbeitsmedizin (BAuA)]

Research report Fb 1010

Interactive systems as part of computerized work

Specifying usability requirements and test criteria for interactive systems

Consequences for new releases
of software-related standards within the ISO 9241 series

Dortmund, May 2003

Thomas Geis
Dr. Wolfgang Dzida
Wolfgang Redtenbacher

Specifying usability requirements and test criteria for interactive systems

Abstract

On the occasion of reviewing the usability standard ISO 9241-10, the experiences of applying the software parts of ISO 9241 (parts 10 to 17) in analysis, design and evaluation of interactive systems are reflected and summarised in this research report. Also, suggestions for improving the software parts of ISO 9241 are put forward for the international standards committee and for practitioners who want to apply the standard series in usability design and quality assurance more properly. Terms and definitions such as design principle, requirement, and criterion are refined to specify the underlying concepts more distinctly, especially with regard to their application in requirements analysis and conformity testing. The content of the adapted definitions has been harmonised with established terms in requirements engineering and quality assurance. An application framework is introduced to help requirements engineers and usabilty designers to apply the dialogue design principles of ISO 9241-10 for verifying the quality level of context-related usability requirements. Potential misunderstandings about the specification of usability requirements in terms of user performance are clarified. Guidance is given to manufacturers of interactive systems on how to specify usability requirements with regard to the context of use (ISO 9241-11). The application framework also aims at assisting usability assessors in specifying usability test criteria. Looking ahead to the review of further parts of ISO 9241, a lot of suggestions are provided for being considered in the standardisation process regarding the structure and content of the multipart standard ISO 9241.

Key words: usability, usability standards, ISO 9241-10, requirements specification, conformity with standards, quality test, computerised work place

Spezifikation von Anforderungen und Testkriterien der Gebrauchstauglichkeit (Usability) interaktiver Produkte

Kurzreferat

Anlässlich der Überarbeitung von DIN EN ISO 9241-10 sind die Erfahrungen mit der Anwendung der Softwareteile der DIN EN ISO 9241 innerhalb von Analyse-, Design- und Prüfprojekten aufgearbeitet worden und werden in diesem Forschungsbericht zusammengefasst. Darüber hinaus werden Vorschläge für die Verbesserung der Softwareteile innerhalb der Normenreihe DIN EN ISO 9241 gemacht, die von deutscher Seite in das internationale Normungsprojekt eingebracht werden. Von den Verbesserungen sollen auch Praktiker profitieren, wenn sie die Norm beim Produktentwurf und bei der Sicherung der Gebrauchstauglichkeit anwenden wollen. Begriffe und Definitionen, z.B. Gestaltungsprinzip, Nutzungsanforderung oder Testkriterium wurden präziser formuliert, um die zugrunde liegenden Konzepte besser zu unterscheiden, insbesondere mit Blick auf ihre Anwendung bei der Anforderungsanalyse und beim Konformitätstesten. Der Inhalt der angepassten Begriffe wurde mit denen harmonisiert, die in Anforderungsanalyse und Qualitätssicherung eingeführt sind. Ein Anwendungsrahmen wird eingeführt, um Analytiker und Softwaredesigner bei der Anwendung der Gestaltungsgrundsätze des Dialogs besser anzuleiten, insbesondere bei der Verifikation des Qualitätsniveaus von kontext-bezogenen Nutzungsanforderungen mit Rücksicht auf DIN EN ISO 9241-10. Mögliche Missverständnisse über die Spezifikation von Nutzungsanforderungen in Form von Tätigkeitsanforderungen wurden geklärt. Den Herstellern interaktiver Systeme wird eine praktische Anleitung gegeben, wie Nutzungsanforderungen spezifiziert werden können, insbesondere mit Blick auf den Nutzungskontext (DIN EN ISO 9241-11). Der Anwendungsrahmen soll auch den Fachkräften für das Testen von Gebrauchstauglichkeit dienen, die notwendigen Test-Kriterien zu bestimmen. Vorausschauend wird auf das künftige Review weiterer Normen der Reihe DIN EN ISO 9241 aufmerksam gemacht. Es werden schon jetzt Vorschläge zur Verbesserung einiger dieser Normen gemacht, insbesondere hinsichtlich der Struktur von DIN EN ISO 9241, aber auch des Inhalts.

Schlagwörter: Gebrauchstauglichkeit, Nutzungsqualität, Usability-Normen, Anforderungsanalyse, Normkonformität, Qualitätsprüfung, Bildschirmarbeitsplatz

Spécifier les exigences et les critères de test de l'utilité pratique (usability) de systèmes interactifs

Résumé

A l'occasion d'une revision de la norme ISO 9241-10, les expériences qui ont été faites en appliquant, dans le cadre de plusieurs projets d'analyse, de design et de test, les parties de cette norme (ISO 9241) qui concernent le logiciel, sont reconsidérées et résumées dans ce rapport. En outre, des propositions pour améliorer les parties de ISO 9241 qui concernent le logiciel, sont présentées au comité international de standardisation en même temps qu'aux praticiens qui voudraient appliquer la norme à la conception des systèmes interactifs autant qu'à leur adaptation à l'utilité pratique. Termes techniques et définitions, comme par exemple principe de réalisation, exigence de l'utilisation et critère de test, sont formulés plus précisément pour mieux distinguer les conceptions de base, avec une prise en considération particulière de leur application lors de l'analyse des exigences et des tests de conformité. Le contenu des définitions adaptées a été harmonisé avec les termes établis dans l'analyse des exigences et dans le contrôle de qualité. Un cadre pour l'application est introduit, pour mieux aider les analystes des exigences et les ingénieurs de logiciel à appliquer les principes de réalisation du dialogue de la ISO 9241-10, surtout dans la vérification du niveau de qualité des exigences de l'utilisation liées à un certain contexte. Des malentendus potentiels sur la spécification des exigences de l'utilisation, exprimée dans le langage des activités des utilisateurs ont été éclairés. Les producteurs de systèmes interactifs y trouvent des instructions pratiques comment ils pourront spécifier les exigences de l'utilisation, avec une prise en considération particulière du contexte de l'utilisation (ISO 9241-10). Le cadre d'application servira aussi aux professionels en vérification de l'utilité pratique à spécifier leurs critères. Pour terminer, l'attention du lecteur sera tirée sur une révision de plusieurs autres normes de la série ISO 9241 à suivre. Ici dèjà certaines propositions sont faites pour améliorer plusieurs de ces normes, surtout en ce qui concerne la structure de ISO 9241, mais aussi quant-au contenu.

Mots clés: utilité pratique, qualité, ISO 9241-10, analyse des exigences, conformité à la norme, test de qualité, travail à l'écran

Table of contents

1 Introduction

By the end of the nineties the development of a series of International ergonomic standards (ISO 9241, parts 1 to 17) for computerized work places was completed. Since then a lot of experience has been gained with the application of these standards in various contexts, including development, procurement, customization and testing. This research report summarizes conclusions for supporting the review of the software parts of ISO 9241 and a more appropriate application of standards in analysis, design and test of software.

Experiences with applying usability standards have got along well by considering the tradition of thought in quality management and software technology. In the nineties best practices of software engineering have increasingly been put forth in software projects. Quality managers urged the projects to respect product standards and process standards, with some of the process standards having been transposed into so-called “key practices” or “base practices”, for instance in CMM (Paulk et al., 1994) and SPICE (El Emam et al., 1997). Although standards do not solve the chronical software crisis in industry in terms of meeting the deadlines or time to market, the standards are respected as a professional means of quality assurance. Complementary to software-technical standards are usability standards (ISO 9241), which became equally important, especially in those projects where usability engineering has been established as part of design as well as of analytical and constructive quality assurance. Consequently, usability standards are ready for becoming applied in analysis, design and testing of software. However, there are different types of usability standards. It happens that practitioners wonder which standard to apply when and how. The usability standards do not yet provide sufficient guidance.

The usability standards have been primarily applied in usability testing. Originally, the standards were also intended for usability design, because testing is just a measure of quality assurance. One cannot test usability into products. Reasons for neglecting the standards in design can be seen in a number of misunderstandings about characteristics of ergonomic standards, which are unique to the issue of

usability and have rarely been encountered in traditional technical standards. Moreover, the role of dialogue principles (ISO 9241-10) in requirements analysis and design was heavily disputed and needed to be studied in a research project. This research report outlines the results and specifies them in terms of an application framework for usability standards as well as recommendations for the future revision of the software parts of ISO 9241.

This report aims at supporting the national as well as the International working groups in the ongoing review of ISO 9241and subsequent standardization projects.

2 Types of standards for software in usability engineering

Standards for software in usability engineering can be put into categories as follows:

1. Proprietary standards (e.g., Windows style guide)
2. International usability standards (ISO 9241 parts 10-17)
   2.1 Generic standards (e.g., dialogue principles, ISO 9241-10)
   2.2 Standards for dialogue techniques (e.g., menu dialogues, ISO 9241-14).
3. Design process standards (e.g., user-centred design processes as specified in ISO 13407)

Two types of standards are applied in the quality assurance of usability. 1. proprietary user-interface standards, e.g. the Windows User Interface Style Guide, Microsoft Corporation (1999) and 2. International usability standards, ISO 9241 Parts 10 - 17. The first type is aimed at establishing a standard user interface (Stewart 1990) including a corporate design, such as the Microsoft interface. The second type provides guidelines for designing the user interface according to state-of-the-art knowledge in usability research and engineering. Both types of standards can be regarded as complementary sources of knowledge to support design and test activities. Style guides and associated supporting tools help develop a standard user interface, thereby suggesting the current technology and proven product attributes to be implemented. Corporate design and consistency of the user interface can be achieved with the aid of style guides. Complementary to style guides are the International usability standards, which typically do not provide guidance for implementing specific product attributes. For software designers this is an unexpected characteristic of standards and has caused some confusion or even rejection of the International usability standards as compared to the traditional technical standards.

Nevertheless, this research report clearly supports International usability standards and intends to remedy the prejudices by explaining the role of usability standards in software engineering. Let us assume, for instance, that an engineer has implemented a user-interface component that complies with a proprietary style guide. Conformity

with a style guide does not warrant that the user interface also complies with the International usability standards. Whereas the application of style guides helps to implement the interface attributes consistently, usability standards help to ensure that usability requirements are being fulfilled. In other words, it does not suffice to adhere to a style guide. Style guide adherence does not guarantee an efficient user task performance, which is merely enabled by the interface attributes. User-interface attributes as such do not warrant usability. Two studies provide evidence that consistency of interface attributes is just a small portion of a product's usability (Travis, 1997; Berry, 2000). Logically speaking, it may be possible that an interface is consistently cumbersome. Hence, usability is a comprehensive quality concept and goes far beyond consistency.

Another distinction between types of standards can be made regarding the quality of products versus the quality of processes. Style guides and the previously mentioned International usability standards (ISO 9241) contribute to the quality assurance of products. Activities of quality assurance within development projects are the subject of process standards (see e. g. ISO 13407). For a mature user-centred process it can be predicted that a design project will be capable of creating a usable product. Therefore, activities described in the product standards are also recommended in the process standards, for instance, context-of-use analyses and conformity tests. The most well known usability process standard, ISO 13407, however, has been developed from the perspective of project managers. Therefore, usability engineers will rarely find information in it that helps to establish a mature usability-engineering process. Recently, ISO 13407, the rather abstract process standard, has been transformed into more practical recommendations. There are two approaches, one of them focusing on so-called base practices of usability quality management (ISO TR 18529, 2002), the other one recommending process test criteria derived from best practices in usability engineering (DATech, 2003). DATech is the German accreditation body for IT test laboratories.

Another distinction between usability standards can be made within the software-related parts of ISO 9241 (parts 10-17) themself. Parts 10 and 12 are intended to provide generic design recommendations in terms of principles such as the dialogue principles (part 10) and the information design principles (part 12). Parts 14 - 17 are devoted to specific dialogue techniques such as menu dialogues (part 14) or form

filling dialogues (part 17). The distinction between generic standards and standards for dialogue techniques may be relevant for the software designer. Standards for dialogue techniques help to obtain specific design recommendations when they are appropriate for the design context. Generic standards should always be adhered to.

3 Requirements within usability standards

The notion of requirement was originally defined in ISO 8402:1994 and has been adopted within ISO 9000:2000 but has rarely been considered in usability design and testing. Instead, usability designers apply rules of thumb or heuristics for designing and testing user interfaces. For designing the interface of a software product the previously mentioned style guides are mainly applied. As a consequence, usability testing suffers from a severe lack of clearly specified usability requirements which a product can be tested against within its particular context of use. However, by adopting the International usability standards in design and testing, this situation does not seem to improve immediately, since the standards do not provide requirements. Therefore it is necessary to first study the established meaning of the term “requirement” and then to explain how usability requirements can be specified with the help of usability standards.

3.1 The established meaning of the term requirement

The original definition of the term requirement was given in ISO 8402:1994 (which has been superseded meanwhile by ISO 9000:2000):

> ”Expression of the needs or their translation into a set of quantitatively or qualitatively stated requirements for the characteristics of an entity to enable its realization and examination" (clause 2.3).
>
> An entity can be a product, an activity or a process, an organization, a system or a person or any combination thereof (clause 1.1).

As can be seen from the above, ISO 8402 defined the term requirement in terms of requirement, which is a tautology. However, it may be very insightful for understanding the original concept of requirement when we study why this mistake happened. In conventional requirements engineering a requirement is only expressed in terms of a product attribute. The crucial word in the definition, however, is the first "or". This word was intended as an "and/or", thereby allowing the requirements engineer to express a requirement in terms of a need (e. g. a user need) or to transform a need in terms of an attribute of a product or even to do both. In usability engineering, best practices of defining requirements have been established which require the specification of a usability requirement in terms of a

need and a translated product attribute as well. Typically, the need is a context-related requirement. It serves as a design rationale (Carroll, 1995) for the suggested product attribute. The need has to be validated to avoid arbitrary requirements specifications. The corresponding product attribute has to be checked for feasibility and appropriateness.

In the currently valid definition of the term “requirement”, given in ISO 9000:2000, clause 3.1.2, the original concept of requirement has been watered down:

> “Need or expectation that is stated, generally implied or obligatory.
>
> Note 1: ‘Generally implied’ means that it is custom or common practice for the organization (3.3.1), its customers (3.3.5) and other interested parties (3.3.7), that the need or expectation under consideration is implied.
>
> Note 2: A qualifier can be used to denote a specific type of requirement, e. g. product requirement, quality management requirement, customer requirement.
>
> Note 3: A specified requirement is one which is stated, for example, in a document (3.7.2).
>
> Note 4: Requirements can be generated by different interested parties."

The new definition does not explicitly say that qualified types of requirements should correspond to each other in such a way that one type is taken as a rationale for the other type (compare with Note 2). For example, a qualified user performance requirement provides a rationale for a product requirement, otherwise a product requirement cannot be validated. As a consequence, in usability requirements engineering, best practices of specifying the requirements must continue to adhere to the *concept* of requirement as defined in ISO 8402.

3.2 Why technical standards typically do contain requirements

Technical standards specify *product* requirements for an assumed or given context, and variations within the context do not affect the requirements significantly. For example, in many countries the format of a sheet of paper is standardized (e.g., in Germany, DIN A4). The standard precisely specifies the required length and width of a paper sheet. It is cost-effective to adhere to this standard, since it facilitates the construction of all kinds of machines in those contexts in which paper sheets are used, for instance, copy machines or fax machines. For this, the standard specifies

the required attributes of a DIN A4 paper sheet. Product attributes are typically specified in all technical standards. In designing a product the designer is instructed by the standard that provides the product requirements.

Technical standards also specify *interface* requirements for an assumed or given context. The context is expected to be adapted to the specified interface attributes. There are two types of interfaces. One type of interface couples components of machines. For example, the format of a data file needs to be specified to enable file transfer between computers. The other type of interface couples a human and a machine, also referred to as user interface. Standards for technical interfaces provide interface requirements, which can be used for implementation purposes. Usability standards, however, do not specify interface requirements in terms of interface attributes to be implemented. Even if the context of a human interface is known, the technical interface attributes do not get specified in a usability standard. One of the reasons for this is that one component of the context is the user. In contrast to the technical context of an interface, a user cannot be arbitrarily adapted to the attributes of an interface. Before explaining further why usability standards should not specify technical attributes, the notion of usability requirement needs to be studied.

3.3 The notion of usability requirement

Specified requirements have been recognized as a key factor of successful projects, with success being defined as

- valid specification of requirements and
- design proposals which hold the test against those requirements.

The notion of usability requirement is not yet widely applied in the community of usability practitioners. The tradition of thought once established in requirements engineering and quality management has not yet been adopted in the community of usability practitioners.

A usability requirement is specified in terms of a *user performance* to be conducted at the user interface according to the implied needs in the product's context of use.

The concept of performance has a threefold meaning. Firstly, performance measures a behaviour of an object (e. g. the selected menu option is echoed at the display by highlighting); secondly, it means a process conducted by the user (e. g.

discriminating selected and unselected menu options), thirdly, performance indicates the quality or quantity of a certain success which a user should achieve in interaction with the system (e. g. a user succeeds to select the intended menu option within a menu panel).

It is up to the requirements engineer to decide for one of the meanings of performance. A usability requirement should be specified in terms of user performance. However, for specifying a usability requirement it is recommended to avoid specifying the behaviour of an object. The previously given examples of performance indicate that the behaviour of an interface object can also be expressed in terms of a user behaviour ("discriminating", "succeeds to select"). Reasons for preferring a specification of usability requirements in terms of user behaviour (user performance) are summarized in section 3.4 of this report.

Even in the domain of professional requirements engineering, the notion of usability requirement is not yet widely applied. Traditionally, any usability-related requirements are classified as so-called “non-functional” requirements. Functional requirements define the fundamental actions that must be executed by the software. Non-functional refers to the manner in which such functions are performed, for example, software performance requirements, software external interface requirements, software design constraints, and software quality attributes (c.f. the corresponding IEEE definition). A non-functional requirement can be, for instance: "95% of the transactions shall be processed in less than 1 second." Only very few requirements engineers would specify a non-functional requirement as follows: "A user shall not have to wait for the transaction to complete." Though not yet established in conventional requirements engineering, the specification of this requirement fits perfectly well with a usability requirement. A user performance is sufficiently specified, the achievement of which can be verified.

The comparison of such an unusual non-functional requirement with a well specified usability requirement may help illustrate, why requirements engineers stemming from different disciplines may run into a dispute. Requirements engineers specifying usability requirements run the risk of falling between two stools. The wording of usability requirements is neither accepted by many traditional quality managers nor by those usability specialists who only rely on heuristic evaluation, user testing or usability metrics.

Nevertheless, such disputes or misunderstandings can be solved. When quality management standards are carefully considered in specifying a requirement it can easily be shown that the constituting elements of a requirement are considered in both kinds of statements previously given as examples of non-functional requirements.

In many so-called “user-centred” design and test processes, the products are not tested against usability requirements, but get evaluated, for instance, with the aid of so-called heuristics (Nielsen, 1992) or checklists. Heuristics and checklists, however, do not represent requirements, since the context to which they are related is not specified. It may well be that a checklist is indeed related to the specific context where the checklist was applied the first time. However, within another evaluation case, the context might no longer be the same. Consequently, the checklist items cannot be regarded as a valid list of evaluation criteria, unless they get related to the actual context of use.

In so-called “user testing”, the notion of requirement is also not yet established. Current practice in user testing is to ask a sample of users to accomplish a specific task with a system, thereby recording troublesome dialogue steps, which may indicate defects of the system. User testing is usually done by the manufacturer of a product in order to improve the system or check its acceptance before it gets delivered to the customer. In user testing, the critical issue is the number of subjects to be tested, whereas in usability quality assurance, the critical issue is the test criterion and its rationale, which is a validated, context-related usability requirement. When user testing is conducted with representative users at their work places, context-related usability requirements should have been identified before the test starts. Typically, however, these requirements are not stated explicitly before running the test. Therefore, user testing often fails as an appropriate measure in usability quality assurance. This is the reason why the annex of the DATech test handbook for usability testing (DATech-Prüfhandbuch Gebrauchstauglichkeit, 2002) does not include user testing as a method. Instead of user testing, the so-called method of “participatory observation” has been preferred. Moreover, the term “user testing” may be misleading, as it is not the user who is being tested but the product, and the product is not tested against the user but against a requirement (which, however, is typically not clearly specified in user testing).

Instead of user testing, participatory observation has increasingly been introduced in usability quality assurance. Participatory observation can be applied even if a usability requirement has not yet been specified; during the observation, the requirement gets explicated and the deviation from the requirement is evaluated with regard to its impact on user performance. Participatory observation focuses on the source of trouble with a system, which in most cases is a deviation from a usability requirement derived from both the context of use and a usability standard. A usability requirement which has been explicated in view of the product's context of use can be further verified by means of a usability standard thus specifying the requirement in terms of a test criterion.

3.4 Why usability standards typically do not contain requirements

In contrast to technical standards, the usability standards (ISO 9241 parts 10-17) do mostly not contain requirements in terms of required product attributes. As previously outlined (section 3.2), the technical attributes of a user interface are not specified in a usability standard even when the context of the interface is known. One reason already mentioned is that users cannot be arbitrarily adapted to predefined technical attributes of interfaces. The exact opposite is necessary. For ensuring the efficiency of user performance, the interface needs to be adapted to the user. Generally speaking, attributes of the user interface depend on the product's context of use, which the user is part of.

Of course, there are exceptions from this rule. Some interface attributes are specified in the standards for dialogue techniques (ISO 9241 parts 14-17). For example, the menu standard (ISO 9241-14) recommends that if an option name in a menu panel is to represent an action (e.g., delete), it should be stated as a verb, unless this is unnatural for the language used (c.f. recommendation 8.3.5). The specification of this interface attribute rests on empirical evidence. There is no doubt that this interface attribute is a valid usability requirement regardless of the context of a product and its interface.

In most cases, however, the specification of a user interface attribute is determined by the product's context of use. As previously outlined, a user interface attribute can be expressed in terms of physical characteristics as well as user performance characteristics. For example, the above-mentioned requirement concerning the

option name in a menu panel could be expressed as follows: "The user should be enabled to immediately understand that a selectable menu option allows an activity to be performed." In order to apply this usability requirement, the context of use must be analyzed. The activity a user needs to carry out (e. g., a task) may reveal the information necessary for specifying a usability requirement. Once a context-related usability requirement is given, for instance, "data must be deletable", the menu standard can be consulted for further specifying the requirement in line with the standard. To sum up, usability standards typically do not contain requirements in terms of required product attributes, because the context of use determines these requirements and the standard itself must be suitable for all likely contexts of use.
Nevertheless, usability standards contain statements in terms of recommendations, with many of them being qualified as conditional recommendations. A recommendation is specified in terms of a "should", whereas a requirement is a much stronger request, typically being specified in terms of a "shall". A conditional recommendation indicates that the conditions (i. e. the context of use) for applying the recommendation need to be analyzed before applying a recommendation in analysis and design.

Recommendations as well as conditional recommendations can be stated in a usability standard, which are based on consensus within a standardization project. Consensus can be achieved about generally valid recommendations irrespective of any context. However, a usability requirement that is context-related can only be specified with respect to a specific context of use. A standardization committee would be overwhelmed with endless discussions about the specifics of contexts and is therefore not a proper place for specifying detailed usability requirements. Hardly any usability requirement is context-independent. Therefore standardization committees should be relieved from the burden of over-specifying the standards.

Consequently, there is a division of effort in specifying usability requirements during analysis within development or procurement situations. The standards provide principles and recommendations. These principles and recommendations given in the usability standards can be taken as a guidance for recognizing and specifying usability requirements in a specific context of use which is subject to analysis. In order to guide the requirements engineer, an application framework should be provided in the standards. Some guidance is already given in ISO 9241-11. A

context-of-use analysis is suggested as a prerequisite for specifying usability requirements. More advice should be given in the other parts of the standards. It should be emphasized that the data acquired from the context of use provide the rationale for specifying context-related usability requirements. These requirements, typically being specified in terms of user performance, are suitable for bridging the gap between the implied needs in the context and the design solution that should meet the implied needs.

3.5 Principles, requirements, criteria – clarifying the difference

Usability engineering is an interdisciplinary endeavour. People from disciplines such as software engineering, psychology and design work together with people from various application fields, for instance, business engineering, marketing or education. It is indispensable to clarify terms and definitions in the very beginning of a project. Section 6 of this report clarifies some of the most frequently used terms in usability engineering. In the past, some terms caused confusion between engineers and psychologists. This section particularly deals with principles, requirements and criteria in order to get rid of the confusion, hopefully once and forever.

To begin with, principles are not criteria. A principle is a general guidance that tells how to treat a certain issue in daily life, in design, in our society, etc. For example, the principle of equity has been introduced in most of the legal systems of western countries to prevent minorities from being discriminated. An advantage of this principle is that the legislative system is relieved from the burden of specifying a number of specific laws all of which are dealing with the nitty gritty troubles of discrimination in our life. When a case of discrimination is encountered, it is more efficient for the people involved to go back to the principle of equity. Therefore, many of the established principles are stated in the constitutional law of a country, for instance, in the German “Grundgesetz”.

In analogy to legal principles, the principles of ergonomic design have been established, with some of them even being mentioned in a European Council Directive (EU Council Directive, 1990). Ergonomic design principles have been established when the originally specified dimensions of user-perceived quality (Dzida et al., 1978) became a German DIN standard. Standard design principles are no legal principles; however, there are many similarities between a standard and a law.

For example, standards and laws both rest on consensus and are elaborated according to rules of conduct within a democratic process of negotiation. Just as the principle of equity prevents from discrimination, so does the ergonomic principle of error tolerance (ISO 9241-10) prevent users from being blamed on the occasion of a mismatch between user and system.

A principle is not a requirement. Of course, a principle may generally require things to be treated in a certain way. However, it is up to an explicit requirement to specify treatment and circumstances in precise terms. Typically, a requirement is generated by various interested parties. Types of requirements indicate what should be done, who is expected to do it or what should be the result.

A criterion is neither a principle nor a requirement. A criterion serves the purpose of comparison, only. Of course, a principle or a requirement can be taken as a criterion when appropriate for comparing things. For example, a quality test is always a comparison between a given quality and a required quality. A requirement, stated in terms of quality, can be taken as a criterion for evaluating the given quality relative to the required one. In most cases, a criterion is more precisely specified than a requirement, otherwise the comparison cannot be made. In a test situation, requirements and criteria can be identical, if sufficiently specified to make the meaning testable.

The notion of criterion has also been used in so-called “quality models”, for instance, in the first quality model of software-technical quality characteristics (McCall et al., 1977). In this model, the term “criterion” is related to a metric, which is by definition a measure for comparing things on a scale. For example, the efficiency of a program can be measured on a scale, and a criterion can be set to be compared with the minimum execution time required for the program to be installed in a fire alarm station. Consequently, the term “criterion” should be applied in cases when a comparison needs to be made for preparing a decision, for instance, when selecting a product for procurement or when deciding about improving the quality of a product.

3.6 Qualitative requirements versus quantitative requirements

A frequently discussed issue is the level of quality specified in a usability requirement. Can a qualitatively described user performance be taken as a

sufficiently specified requirement? For example, does it suffice to require: "A user shall not have to wait for the transaction to complete." Or is it obligatory to specify: "A user shall not have to wait more than 1 second for the transaction to complete."

The answer is: "It depends on the product's context of use." A qualitatively specified level of user performance may suffice in an ordinary office context where users form expectations about response times of transactions. In principle, the required response time should meet user expectations (ISO 9241-10). In such a case, users do not consider themselves to wait. They might get nervous about system response time when it deviates from the time expected. However, in the context of a fire alarm station, the level of quality to be specified for a required user performance can be different. A response time of 1 second can be regarded as a critical level, whereas in the case of an ordinary office, a rationale for this quantitative level can hardly be given. One might ask, for instance: "Why not one and a half seconds?" Generally, the risk with quantitatively specified levels of user performance is that there is no rationale. Of course, there are exceptions from this rule. For example, when entering characters, the echo displayed on the screen should appear within less than 500 msec. There is empirical evidence for this level of quality of an echo on displays, regardless of any context. Notably, in this case a product attribute is specified, not a user performance.

Specifying a quantitative level of user performance can even be misleading. The required efficiency of user performance can be interpreted as a productivity measure. This is due to the threefold meaning of performance (compare section 3.3). Although there is a relationship between user efficiency in ergonomic terms and user productivity in economic terms, both concepts should not be confused. This happened, for example, in ISO 9241-11. Some of the criterion levels for efficiency specified in the examples of ISO 9241-11 are arbitrary, especially when being stated in terms of time. For example, the criterion for the efficiency of an installation task is specified as "within 10 min" (annex C of ISO 9241-11). Why not 11 minutes? In the context of use, an implied need for productivity may have been recognized; however, this need should not be taken directly as a usability requirement, but rather as an organizational requirement. When productivity and user efficiency is compared, it should be considered that user efficiency is not a linear function. Often, a medium

level of efficiency is an appropriate level of user effort, whereas extreme high or low levels are questionable.

When conducting a dialogue with the system, the efficiency of user performance can be classified according to a number of factors that help to structure the main aspects of efficiency. In the following, some examples are given to describe factors of efficiency.

A user performance expended in the dialogue with the system is efficient

- when the user can focus on the original task to be done and need not bother with extra dialogue steps which are solely due to the technology being used;
- when the user immediately finds all required information and understands what the displayed information is meaning for accomplishing the next dialogue step, thus preventing the user from undue effort in gaining an understanding;
- when the user can rely on the experiences made during system use and the experiences made in completing tasks in a specific context, thus preventing the user, for instance, from the burden of dealing with an unexpected reaction of the dialogue system or an inconsistent dialogue flow;
- when the user can organise the flow of dialogue steps in such a way as the original task intrinsically requires it, thus preventing the user from undue effort to keep in control of the dialogue;
- when the user can recover from an erroneous dialogue step with minimum corrective action;
- when the user does not need to spend undue effort in learning how the dialogue is to be conducted;
- when the user can adapt the dialogue to justified personal needs, thus avoiding unnecessary effort to overcome barriers of performance.

Seven dialogue principles have been defined to classify the above listed aspects of efficient user performance in a concise way (ISO 9241-10).

At the time when the usability standards (ISO 9241 parts 10-17) were published, the application of the standards in requirements engineering was not yet fully understood. A typical indicator is a statement as follows: "ISO 9241-11 does not and

cannot provide guidance with respect to particular criterion levels" (ISO 9241-11, p. 16). This statement is no longer valid. An application framework is presented in this report that shall assist clear specifications of usability requirements as well as test criteria.

4 The role of principles presented in ISO 9241-10 in analysis, design and test

ISO 9241-10 is an International standard that lays down principles of dialogue design for interactive systems and corresponding recommendations. Of course, the set of principles is not the only existing one. A lot of such principles have been published with many of them being adopted in user-interface evaluation, for instance, Scapin and Bastien (1997), Shneiderman (1998), Dix, Abowd, Beale, and Finlay (1998). The most prominent example is the set of so-called heuristics (Nielsen, 1992, 1994). The earliest source of a set of empirically based principles is Dzida, Herda, and Itzfeldt (1978), which was taken as the basis for developing the German standard DIN 66234-8 (1988).

There is no longer any controversy about the need for principles in designing and assessing user interfaces. However, the need for a standard set of principles was a highly disputed issue in the late seventies when principles of dialogue design were published in the community of software engineering for designing the first interactive systems. The controversy continued when the dialogue principles were published as a German standard (DIN 66234-8, 1988). With the advent of a standard of so-called quality characteristics for software evaluation (ISO/IEC 9126, 1991), the discussion about the need of standard dialogue principles came to an end. The community of software engineers respected the existence of software-technical quality objectives as well as the corresponding principles of user-interface design. The German standard was submitted to ISO committee TC159/SC4/WG5 and was updated in an International consensus process, which resulted in ISO 9241-10 (1996).

Since then ISO 9241-10 has been extensively applied in various contexts. However, many software designers and usability testers complained that the dialogue principles and allocated generic requirements in Part 10 do not provide specific test criteria. This is why many people expected the standard to provide checklist items for designing or testing products. The revised version of ISO 9241-10 should therefore include an application framework for the standard, to guide the reader when applying the standard in designing and testing products. This report is aimed at explaining the framework. Lessons learned through the application of ISO 9241-10 are summarized,

so as to provide a concise commentary on the roots, the state of the art in usability requirements analysis and testing, and perspectives of further improvements of the multipart standard ISO 9241.

4.1 A general framework for applying ISO 9241-10

The application framework outlined in this section emphasizes the need for specifying a usability requirement in terms of both the product's context of use (ISO 9241-11) and the relevant dialogue principle. Three main target groups use ISO 9241-10 and need to be considered when describing the application framework: 1. requirements engineers, 2. software designers, and 3. quality testers. These target groups should benefit from the standard and the application framework, in order to cooperate more effectively in ongoing projects.

A mature usability engineering process starts with the preparation of a software project. In many conventional projects, however, the usability specialists come into play relatively late, typically when “user testing needs to be scheduled”. As outlined in section 3.3, user testing is no longer seen as the central measure of quality assurance, because one cannot test usability into a product. In the very beginning of a software project, usability requirements need to be specified, which provide a valid basis for usability design and test. One should not test a product against users but against requirements. For supporting the requirements engineer's activities, it is necessary to analyze the context of use of the product to come. Usability objectives are then derived from this analysis in order to describe a user-oriented concept of system use, which is complementary to the envisaged system concept. Two standards are relevant for organizing the process of analysis and specification of quality objectives. ISO 13407 describes the analysis process from a management point of view. ISO 9241-11 provides guidance on the quality concept of usability and how to take care of context-of-use data.

4.1.1 Construction of usability requirements – the role of ISO 9241-10

For every main target group of ISO 9241-10, an application framework should be provided in the revised version of ISO 9241-10. Figure 1 describes the analysis case which a usability requirements engineer is dealing with, figure 2 provides an example of this case. Figure 3 describes the design case of applying the standard, with figure

4 providing an example. Figures 5 and 6 deal with the conformity test case (see below).

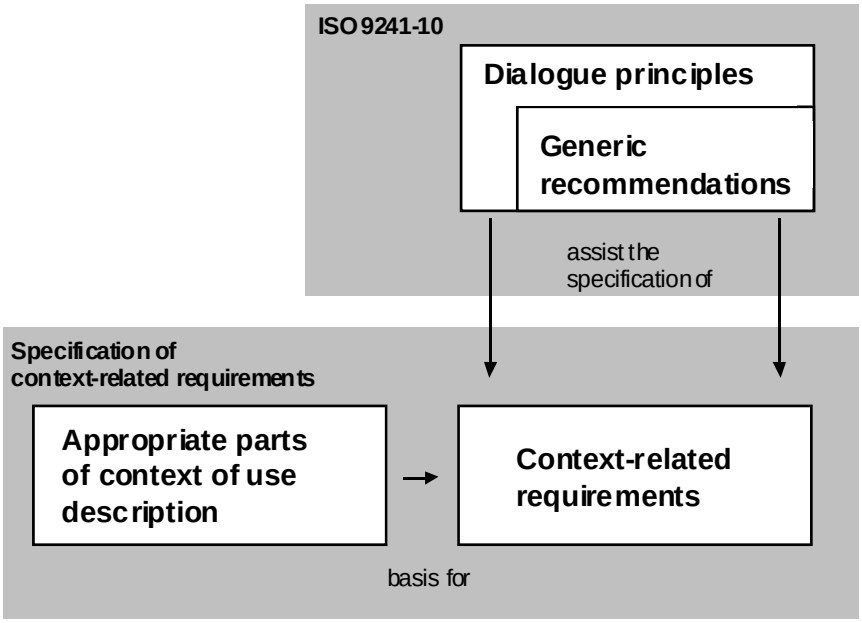

FIGURE 1 Analysis case: the application of ISO 9241-10 in the specification of context-related requirements

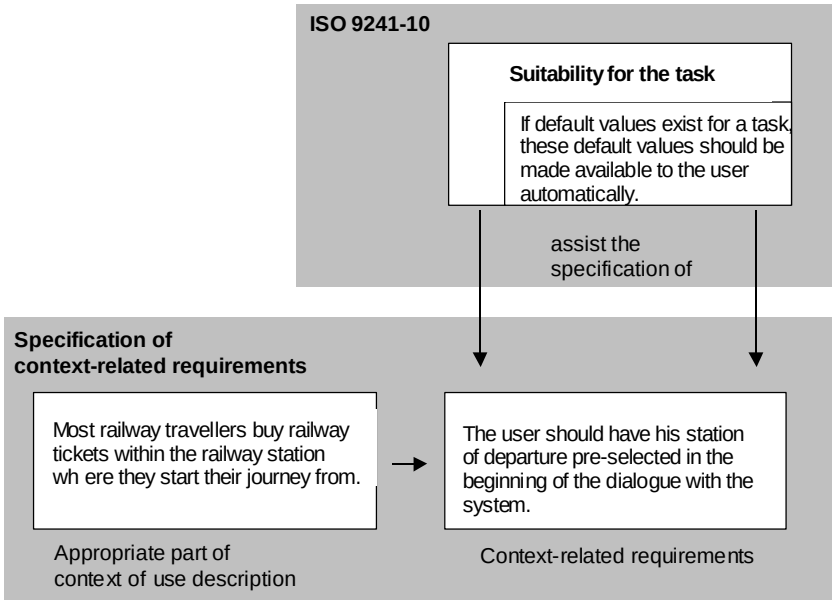

FIGURE 2 Example analysis case: the application of ISO 9241-10 in the specification of context-related requirements

In practice, the aim of applying the principles or the corresponding generic recommendations is to assist the analysis of the context of use for identifying or

recognizing context-related usability requirements (compare with figure 1 and the example provided in figure 2). ISO 9241-10 primarily serves the purpose of specifying context-related requirements in line with the standard. This can be done by selecting an appropriate dialogue principle and/or a corresponding generic recommendation and applying it to an appropriate part of the context of use description to determine a context-related requirement. The list of generic recommendations allocated to each of the dialogue principles is not complete, it represents the state of consensus. If the requirements engineer does not find an appropriate recommendation then the most fitting dialogue principle should assist the specification of a usability requirement.

The dialogue principles or the associated generic recommendations can also be applied for constructing usable design solutions. Figure 3 describes the role of ISO 9241-10 in combination with those usability standards which are devoted to dialogue techniques (ISO 9241 parts 14-17). Although the generic recommendations of ISO 9241-10 are not specific to any dialogue technique, they may correspond to a specific recommendation given in a dialogue-technique specific standard. Figures 3 and 4 suggest the designer to pick up an applicable recommendation in one of the specific standards for dialogue techniques and set this recommendation relative to both the context of use and a design principle (or generic recommendation). This approach helps the designer to create a design solution in line with the recommendations given in usability standards. Although the usability standards do not provide a design proposal, the designer is well advised in creating a solution that does not violate the standard recommendations.

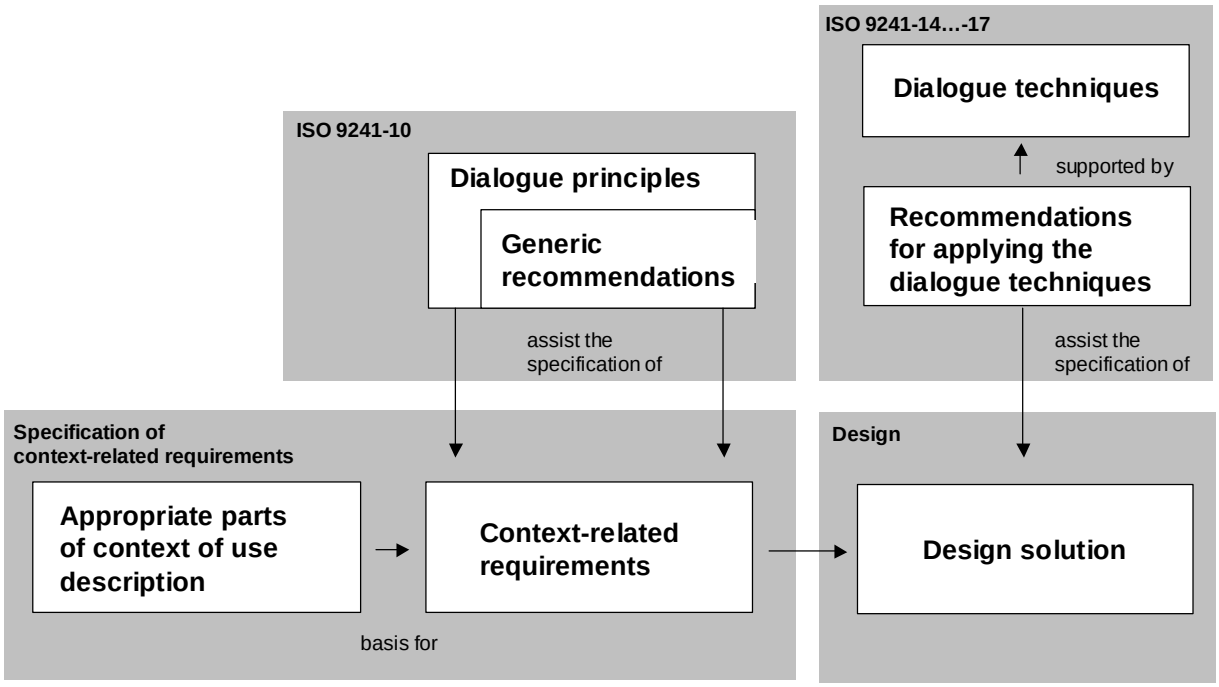

FIGURE 3 Design case: the application of ISO 9241-10 in association with the dialogue techniques (ISO 9241 parts 14-17)

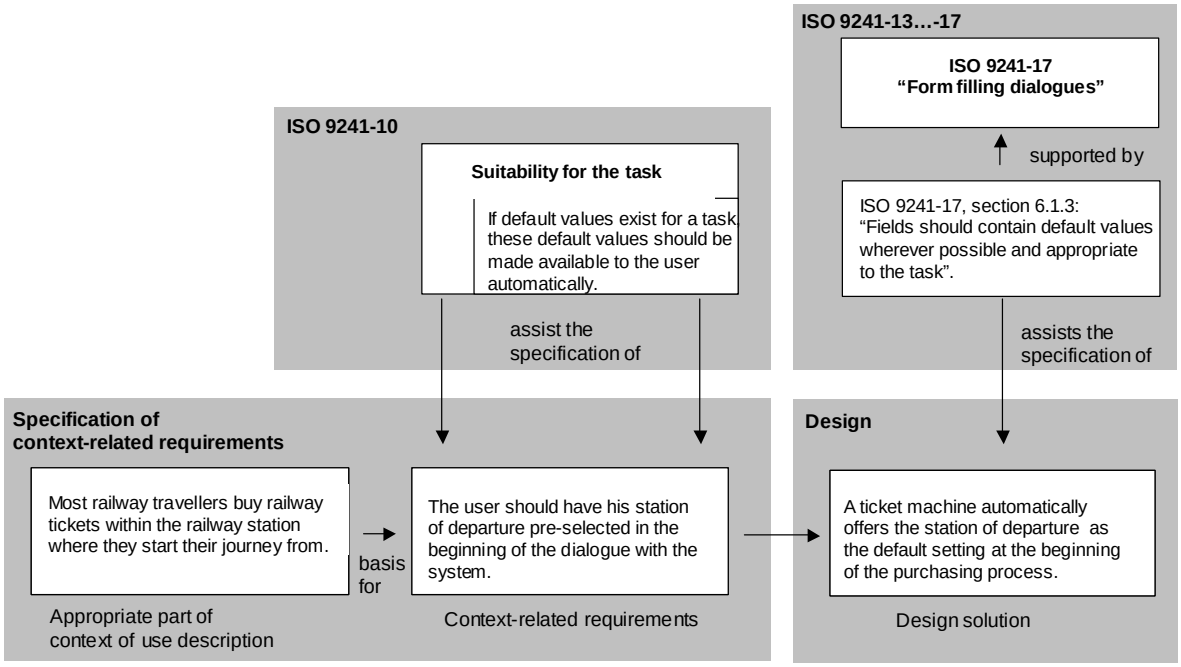

FIGURE 4 Example design case: the application of ISO 9241-10 in association with the dialogue techniques (ISO 9241 parts 14-17)

Eventually, testing the design solution for conformity with usability standards is necessary to check the design for its level of quality. Figures 5 and 6 describe the case of a conformity test with ISO 9241-10. A test for conformity requires a test

criterion to be specified (compare with section 3.5). The test criterion represents the required minimum level of usability. Of course, a context-related requirement and the design solution can (and often should) exceed this level. Nevertheless, a test against the test criterion is indispensable, even if the context-related requirement and the design solution depict a higher quality level than the test criterion. The example in figure 5 describes a test case with a test criterion that is already sufficiently specified in the context-related requirement (and therefore identical).

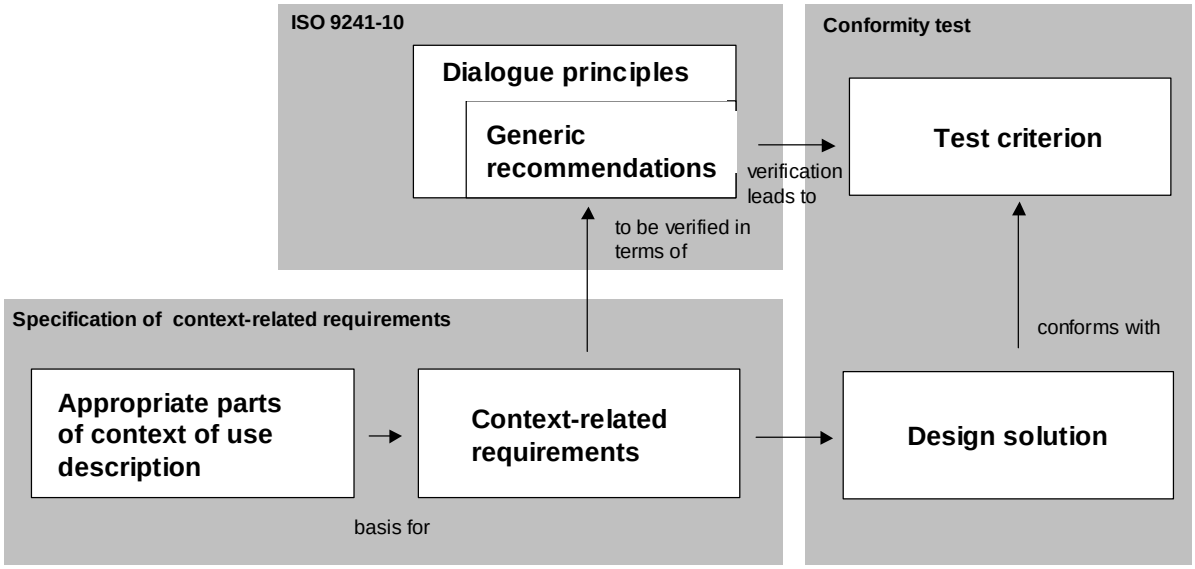

FIGURE 5 Conformity test case: the application of ISO 9241-10 in conformity testing

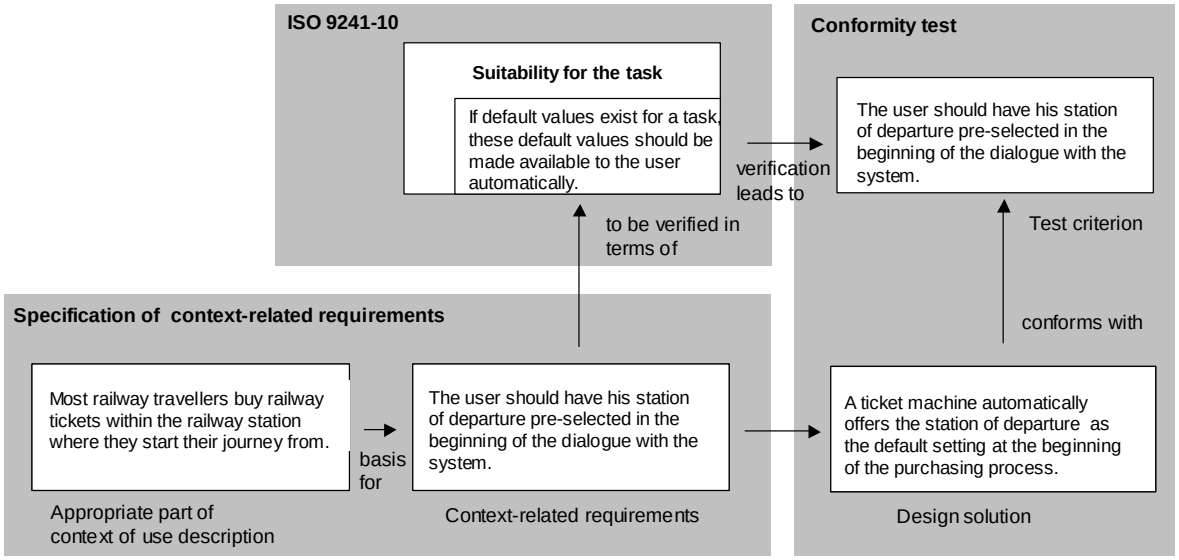

FIGURE 6 Example conformity test case: the application of ISO 9241-10 in conformity testing

4.1.2 How manufacturers can specify usability requirements in line with ISO 9241-10

Usability requirements cannot simply be analyzed. Although users are willing to spell their requirements, they are typically not able to do so. Even worse, what a user wants is not always what the user really needs. Therefore, usability requirements need to be developed by an experienced requirements engineer in cooperation and in consensus with users. This is a process of helping users to figure out what they want to do and how they want to use a product.

Software designers are willing to design for usability, but on their own they are not able to specify valid usability requirements. Typically, software designers anticipate user needs. They hope to meet the users' intentions but are often faced with mismatches and misunderstandings.

Although it is indispensable to ask users what and how they want to do something, it can be dangerous to develop products only based on what users say they want. Usability requirements cannot simply be analyzed by “capturing” the features, functions and solutions which users ask for. Before a requirements engineer accepts what a user asks for, the requirements engineer needs to understand what the user is aiming to achieve. Increasing the user’s understanding of the potential of a technology is one aspect of the dialogue between requirements engineer and user. Even more important, however, is increasing the designer's understanding of what the user needs to achieve and what the user wants to get improved. The requirements engineer should listen to the user, but should know what to listen for and what to ignore. Users are consulted as experts in their work, not as experts in designing software. Users tend to compare among products, thereby getting impressed by the latest advances in features and functions. It is tempting to assess them and we should not prohibit users to do so. Nevertheless, it is of vital importance to ask users why they appreciate or reject product attributes. We want to understand the user's needs in terms of appropriate user performance.

For example, we know that many users like to give suggestions for designing colour combinations. Well, as far as colour is a matter of taste, the user is always right. However, if colour is a matter of usability (i. e. legibility of characters or discriminability of items) then the user is respected as a judge of whether something is legible or discriminable for him, but not as an expert on colour design.

In the past, the usability specialist’s common phrase was: "Know the user". However, this turned out as too psychological a focus. While establishing usability research in the eighties, the focus expanded towards: "Know the user's task". However, even this focus was still too narrow, since the initial understanding of a task was that of the computer user's actual flow of dialogue steps. This focus caused a methodological bias also referred to as “immunization trap”. The definition of the user's task requirements was influenced by existing system features. Usability requirements and product evaluations thus included circular statements. In response to this mistake, the focus was expanded from the user's task to the system's context of use. Context-of-use analysis became a state of the art concept in usability engineering (ISO 9241-11). Nowadays, the saying is: "Know the product's context of use."

An unbiased tool for describing the context of use is the language of the user, also referred to as "context scenario" (Dzida and Freitag, 1998, DATech, 2002). A context scenario describes the user's work situation as an episode including the objectives, the key tasks to achieve them, means and prerequisites for conducting tasks, and also including obstacles or shortcomings as well as the user's vision on how to improve the situation. Ultimately, a context scenario provides a structured story of the user's daily work episodes, which are easy to read for both the user and the designer.

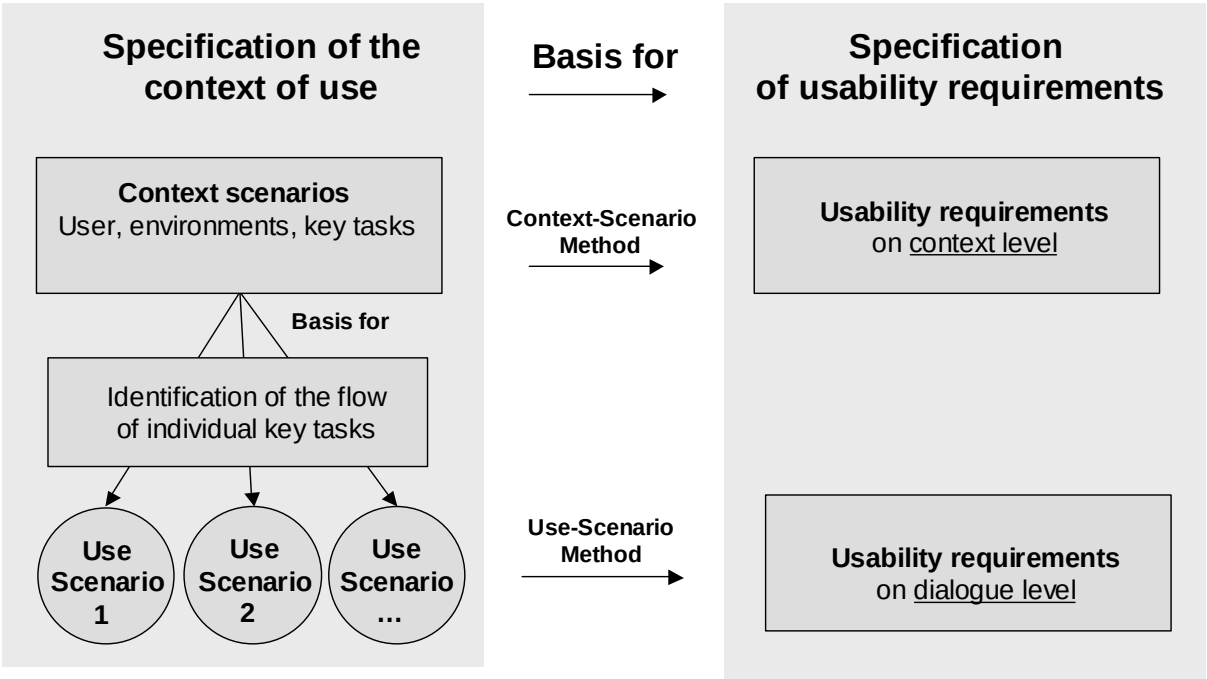

FIGURE 7 Activities and results in specifying usability requirements

Figure 7 describes the activities in usability requirements engineering (Dzida and Geis, 2003). For every key task which the user mentioned in the context scenario, a use scenario is described to capture the user's knowledge about the conduct of a task or the way of performance. As a result, two sources of information are exploited for specifying usability requirements:

1. the product's context of use, and
2. the user's task performance.

Use scenarios are extended use cases (Jacobson, 1995). It is important that both context scenarios and use scenarios get validated by the users concerned. The methodical procedure is laid down in the DATech Test handbook Usability (DATech, 2002).

Most important is a valid identification of contextual facts described in the context scenario. Once context scenarios have been acquired (and validated by the interviewed users), they need to be further processed in order to develop usability requirements. The schema presented in table 1 describes a way for developing usability requirements in a systematic way. This schema applies to context-related usability requirements. A validated context scenario is a prerequisite for the specification of context-related usability requirements. A key methodical approach to engender objective requirements is the second step, depicted by column two of the

schema in table 1. An implied need is a self-evident demand that results from the context data and is not disputed among people who are representative for the context. If there is still any dispute about an implied need, then an objective understanding has not yet been achieved. Dialogue principles of ISO 9241-10 are a good tool to derive usability requirements for interactive product use.

TABLE 1 Transforming context data into usability requirements

Context of use	Implied need	Dialogue principle ISO 9241-10	**Usability requirement**	Design proposal
→				→
Context description validated by users.	What is undoubtedly necessary in this case?	Which principle fits well with the implied need?	What shall the user be enabled to do in interaction with the system?	Which product attribute is proposed to support this?

Principles are, for instance, suitability for the task, self-descriptiveness, controllability, and error tolerance. They define the scope of dialogue design by setting the lower boundary which a solution proposal should not violate. Consequently, the resulting specification of usability requirements is based on valid needs, objective performance requirements and solution proposals which are pre-checked against a quality standard.

Further usability requirements are devoted to the structuring of information to be presented on the display, with many of them being also covered by style guides. However, user performance requirements which are related to an interactive conduct of tasks have to be specified first. For example, before designing the layout of a series of online forms, it is necessary to determine the flow of work, i. e. the sequence of online forms. This example can be taken as a paradigm for the development of usability requirements in general. First of all, a schema for deriving usability requirements is needed (for instance the DATech schema, table 1). Then the context-related requirements need to be determined. Finally, the information design requirements have to be specified. In practice, one can observe that

designers start too early with the specification of information design requirements, since it is easy to start with them due to the easily available style guide.

TABLE 2 Example of systematic extraction of a usability requirement

Context of use	Implied need	Dialogue principle ISO 9241-10	**Usability requirement**	Design proposal
→				→
When Mr. Miller cancels meetings, unfortunately he tends to forget to also cancel the reservation of the meeting room.	Mr. Miller needs to be reminded to cancel room reservations as soon as he cancels meetings.	Error tolerance	A software supporting the scheduling of meetings must at least remind the user to cancel the room reservation once the meeting has been cancelled.	The software supporting the scheduling of meetings for team members opens a window “Would you like to cancel the reserved meeting room, too?” as soon as the inviting team member cancels the meeting through the software

Nevertheless, it may occur that despite a long discussion people do not agree upon an implied need. This fact is captured and must be pursued during the subsequent development of usability requirements. It may turn out later that an agreement or compromise can only be achieved in view of an implemented interactive prototype. It is vital to emphasize that the endeavour of coping with mismatches in the process of constructing requirements is a basic approach to quality assurance in usability engineering. It may be costly to deal with these mismatches, but it is far more expensive to reconcile them during implementation or maintenance.

The example presented in table 2 shows a fragment of a scenario in which the context of an office clerk is described. Based on objectively stated implied needs these context-related demands are transformed into demands regarding interactive product use. One step within this approach is the application of dialogue principles. The dialogue principles help to clarify the scope of usability requirements. Given the implied need and the context data, it is obvious that "error tolerance" (ISO 9241-10)

can be taken as the most suitable design principle. Context data as well as the implied need provide the rationale for adopting this principle. Apparently, Mr. Miller needs to be reminded to cancel room reservations. One might argue that the user should not have to cancel the room reservation at all and that the system should perform this step automatically. This might well work in certain contexts; however, the minimum level of usability required in view of the implied need is “to be reminded” and therefore satisfied. The dialogue principle error-tolerance points to cope with contextual demands during task performance. When looking into the standard (ISO 9241-10), the requirements engineer can confine the reading of the standard to only one dialogue principle. By interpreting context data in the light of the generic recommendations given in the standard, the requirements engineer will specify an appropriate usability requirement. The more precise the key requirements are formulated in terms of the context data, the better is the usability engineer's understanding of what the user really should be able to do in interaction with the product.

4.2 Overlap of dialogue principles in test and design

Originally, the dialogue principles have been claimed as being statistically independent dimensions (Dzida et al., 1978). As factors of quality, the dialogue principles are independent. Every principle can be applied for assessing a quality issue independently from any other principle, which may also apply to this issue. However, the dialogue principles are no longer independent when being applied as design principles. Notably, the empirical basis of the principles does not stem from design but from judgement.

Nevertheless, it is possible to adopt the principles in dialogue design. Design for controllability, however, is not independent from design for error tolerance or learning. For example, a ticket machine should be simple as regards to user control. Otherwise the initial learning effort for casual users might be too high. When a ticket machine becomes increasingly complex, the requirements for controllability become also complex, and design for error tolerance, too.

Sometimes a usability problem does not correspond to only one recommendation or one dialogue principle but to more than one. Generally, the more design principles are addressed within a problem description, the more complex is the usability problem. For example, a user applies "copy and paste" to insert a graphic from one file into another one. This sequence of commands is expected to be applicable for text and graphics. After having edited further text, the graphic suddenly vanishes and gets substituted by a big red "X". The user does not know why this happened and what to do next. This defect violates the principle of conformity with user expectations. Additionally, either the principle of error tolerance is violated or the principle of self-descriptiveness.

When more than one dialogue principle is addressed by a usability problem, it happens that one or more of the principles account(s) for the symptom while another one accounts for the source of trouble. For example, a windows user may receive an e-mail with an attached file from a Macintosh user and does not succeed to open the file by clicking on it. At first sight, the symptom could be assigned to a lack of "conformity with user expectations". However, the real source of the trouble is that under Windows, the type of the file (e.g. a doc file) needs to be specified via its file name whereas on the Macintosh, the file name can be arbitrary (and the file type is specified in a manner that is invisible to the user, i.e. via the so-called "resource fork"). Therefore, the source of the usability problem is that the e-mail program of the receiver violates the principle of "suitability for the task" as it knows (from the so-called MIME type of the attachment) the file type of the attachment and could easily adjust the file name extension to mark the file type according to the local operating system conventions. In this way, the user would not be bothered with activities that are alien to the original user task and which the system can take over itself.

When more than one dialogue principle is concerned in the design of a dialogue step, the overlap of dialogue principles can have negative consequences if the designer does not trade off the benefits of one principle against others. The previously described assessment of symptom and source of a usability problem is not a matter of overlapping dialogue principles, since the symptom can be independently assessed from the source. However, in designing a dialogue, an overlap of design principles can have negative consequences. Designing is compromising (Winograd,

1998), meaning that in usability engineering there is rarely a right or wrong design solution but only a more or less appropriate one. Let us assume that a user fails to casually buy a special ticket at the prototype of a ticket machine. For improving this user task, the designer has to trade off the controllability of the dialogue against the self-descriptiveness of the entire user interface. As a compromise, it may be more appropriate for the design of the ticket machine to offer regular tickets differently than special tickets. Generally, it can be stated that negative consequences from overlapping design principles arise from lacking user participation in the design process.

5 The future role of the software parts within the ISO 9241 series

A standard is not a textbook. Nevertheless, practitioners should be enabled to better learn the application of standards with the aid of a guiding framework for analysis, design and testing. Frameworks should be provided by the standards. Fragments of guiding information are already included in some of the standards, for instance, in the amendment of ISO 9241-1 and in ISO 9241-11.

In addition to the application frameworks, it is indispensable for the practitioner to gain a clear overview of the content of ISO 9241 and its structure. Also, some guidance is necessary for reading the standard. A simple structure may help to rapidly find the appropriate recommendations and select the information needed.

The future role of usability standards is to assist the usable design of interactive products. Hardware and software are components of the products. Designers should be guided by cross-references to consider both software and hardware standards. Action items can be suggested to the further review of these standards. Before doing that, the current structure of the multipart standard needs to be reflected.

5.1 Overview on existing structure of ISO 9241

The list of the multipart standard ISO 9241 does not depict a clear structure. Considering the concept of usability as defined in ISO 9241-11, it may be possible to draw a simple structure. Therefore, it is necessary to keep in mind that the usability of a product is context-dependent. Hence, it is appropriate to distinguish between context and product. As regards the product, hardware and software can be distinguished. Accordingly, the series of standards can be grouped in hardware and software related as well as those, which are related to the product's context of use.

In table 3, the usability standards dealing with hardware are parts 3 and 4 as well as 7, 8, and 9. Software usability standards are parts 10 - 17. Standards dealing with the ergonomics of the context are parts 2, 5, and 6.

TABLE 3 Parts of ISO 9241 (May 2003)

Part	Title
1	General Introduction
2	Guidance on task requirements
3	Visual display requirements
4	Keyboard requirements
5	Workstation layout and postural requirements
6	Guidance on the work environment
7	Requirements for display with reflections
8	Requirements for displayed colours
9	Requirements for non-keyboard input devices
10	Dialogue principles
11	Guidance on usability
12	Presentation of information
13	User guidance
14	Menu dialogues
15	Command dialogues
16	Direct manipulation dialogues
17	Form filling dialogues

Table 3 provides a list of standards under the title “Ergonomic requirements for office work with visual display terminals (VDTs)”. Parts 10-17 are software usability standards.

Note: The ISO 9241 standards series is in the process of getting a new title “Ergonomics of Human System Interaction” in order to take off the limitation to office work, as much of its content is also applicable to other dialogue systems.

The concept of usability applies to the hardware and software of a product. An attempt has been made to also apply this concept to the quality of the product's context of use as well as to the entire work system that comprises product and context of use. This attempt failed, since the quality of an entity is always relative to contextual requirements. It is questionable to set the quality of a context relative to the requirements of its context. Of course, elements of a context do have quality, for instance, the user's task may have shortcomings. From discussing this issue, an interesting conclusion can be drawn. The concept of usability always goes along with the concept of interface. Hence, designing for usability means designing interfaces,

for instance, hardware and software interfaces to users or organizational interfaces between users.

Some of the standards are basic. If elements of a product's context are of poor quality then there is no chance for a high-quality product to become usable in the sense of ISO 9241-11. For example, shortcomings in the design of tasks, organization of work or workstation layout cannot be compensated by a high-quality user interface of the product. In other words, designing the user interface can result in a partial optimization of the interface, if the quality of the context of use is ignored. For example, the screen layout may be excellent but other parts of the user interface such as the dialogue may be cumbersome. A well-designed user task is a sound basis for designing a usable dialogue with the application program that should assist the task.

Figure 8 illustrates the structure of the usability parts of ISO 9241 in terms of a matrix. Task requirements are the basis from where to start in analysis and design. The requirements engineer analyzes the key tasks described in a context scenario. The ergonomic quality of tasks and organizational conditions should be evaluated by the users. Users are willing and able to envision what to improve in the current context situation, thereby providing valuable information about the potential of innovation in an application domain of software. The requirements engineer can obtain guidance on task design from ISO 9241-2. This standard is a source of recommendations, which should be considered in the interviews with users, so as to figure out the user needs, which are implied in the context. Based on well-designed user tasks, the development of usable software products can be prepared. Task requirements get transformed into dialogue requirements (compare with the schema in section 4.1.2, figure 7 as well as tables 1 and 2).

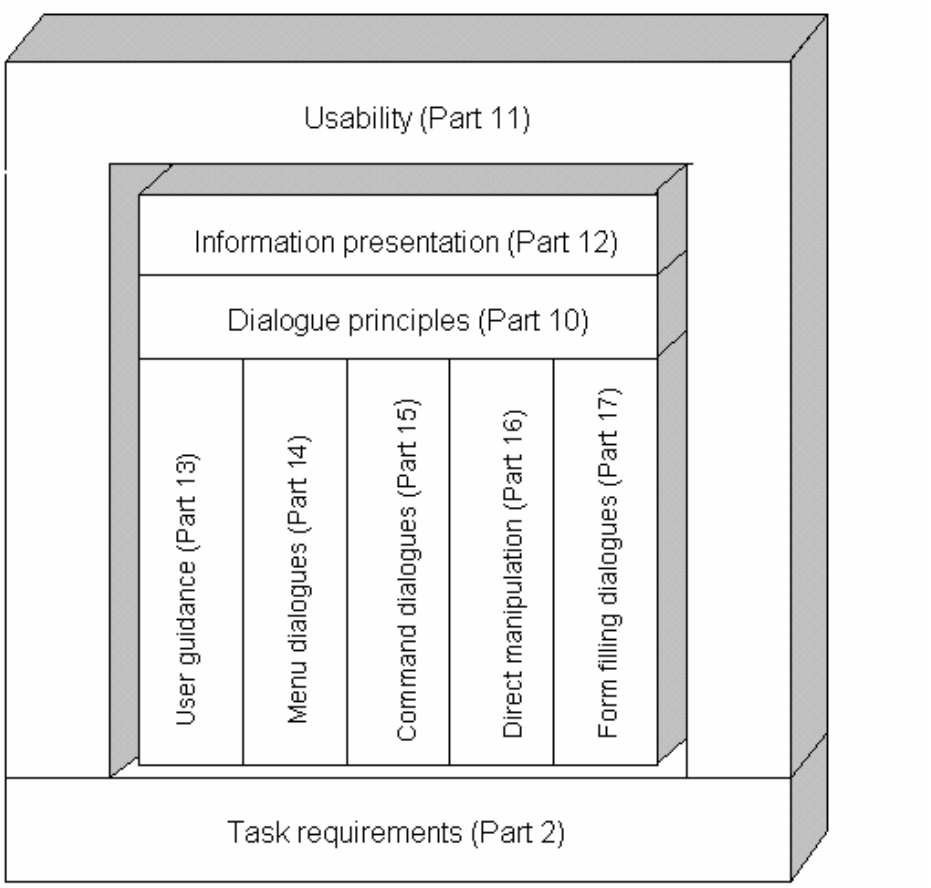

FIGURE 8 A structure of the usability parts of ISO 9241 (Dzida, 1995)

Figure 8 also provides a structure of the software usability standards to be applied once the context of use has been analyzed. The ultimate quality objective is usability as defined in ISO 9241-11. In the matrix of figure 8, part 11 frames all the software standards in order to illustrate that other parts of the standard help to achieve this objective. Within the software usability standards, there are two standards at a general level (ISO 9241-10, dialogue principles, and ISO 9241-12, information presentation principles). As outlined in section 2, these standards are neutral towards any dialogue techniques. The design principles, however, can be recognized when reading the specific recommendations given in the standards for dialogue techniques. For example, part 14 (menu dialogues) provides recommendations for menu presentation, which relate to the principles of part 12, whereas recommendations for navigation within menus pertain to the principles of part 10. Having the principles in mind, it is quite easy for the reader of the standards (parts 14 - 17) to find the appropriate section. This experience with the reading of standards suggests not to start with the specific standards on dialogue techniques. It may be tempting to do

that, especially when the reader expects to obtain advice for design proposals. However, design is much better guided by design principles than by some sort of specific checklist.

Figure 9 recommends to first read the generic standards, so as to prevent the reader from not seeing the forest for the trees. The most general standard defines the quality concept of usability (ISO 9241-11). The reader is informed that a product's usability depends on the context in which the product is being used. In other words, there is no such thing like THE usability of a product. Principles for the design of user interfaces are given in two standards, one devoted to the dynamic aspect of interactive system usage, so-called dialogue principles (ISO 9241-10), the other one devoted to the static aspect, often referred to as information design principles (ISO 9241-12). Standards devoted to specific dialogue techniques (Parts 14 - 17) provide recommendations which should be considered if applicable. However, there is freedom of design. The interface designer can develop a design solution for a specific dialogue technique (e. g. menu dialogue) that deviates from a recommendation of the respective standard (ISO 9241-14). Nevertheless, the design solution must comply with the design principles laid down in Parts 10 and 12.

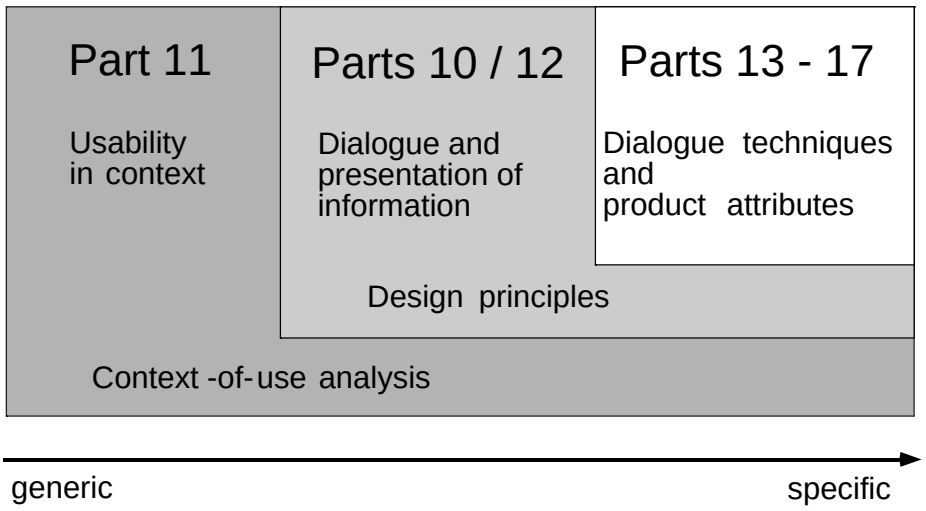

FIGURE 9 Software usability standards (ISO 9241) structured from generic to specific ones

Figure 10 provides an example application of the standards, starting with an analysis of the context of use (Part 11), then deriving test criteria for the solution proposals by

applying, for instance, ISO 9241-10 and finally considering the recommendations given in the specific standards for the dialogue techniques (e. g. ISO 9241-14). The following example on purpose does not use a software product, to demonstrate the extended scope of future releases of ISO 9241, which will apply to all kinds of interactive products. The example can be read as follows: from the context of use we know that Mrs. Miller usually boils eggs for the family, however, the requested softness of eggs varies among family members. The user interface of an interactively usable egg boiler should allow Mrs. Miller to keep in control of how soft the eggs be boiled. The designer takes this requirement and proposes a horizontal menu bar with options of softness. ISO 9241-14 recommends how to present options in a horizontal menu. However, if the designer suggests a solution other than a menu and if this solution is not covered in the standards, the designer may do so, but the design principle of controllability (ISO 9241-10) still must be fulfilled.

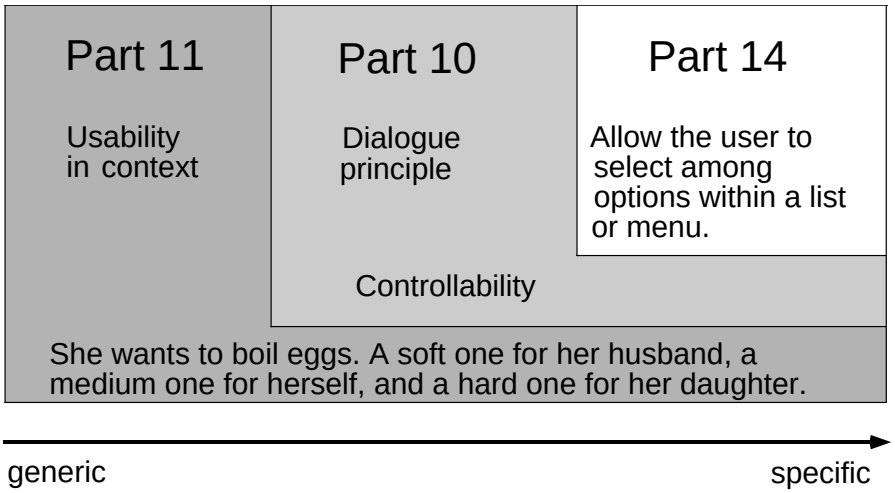

FIGURE 10 Applying usability standards to the design of an interactive egg boiler

5.2 Recommendations for further reviews of ISO 9241

With the review of ISO 9241-10, a number of experiences have been discussed so that proposals can be made for the reviews of other parts of ISO 9241.

5.2.1 ISO 9241-11 needs to become ISO 9241-1

The ultimate quality objective of software and hardware of products is usability. According to this quality model, the current series of ISO 9241 standards should be restructured, starting with guidance on the concept of usability, which is currently ISO

9241-11. Restructuring the standards may entail renumbering. Advantages and disadvantages of renumbering should be considered. A disadvantage is that many people who are already familiar with the ISO 9241 series might become irritated. However, in particular those people who are well trained in the application of ISO 9241 will certainly not protest against the renumbering of the current part 11 standard, since the current number 11 is arbitrary and there are not many references to ISO 9241-11 from literature outside of ISO 9241. The contents of the current parts 1 and 11 could be merged, so as to provide an introductory standard containing usability as a leading quality concept for all kinds of interactively usable products, a usability quality model, and guidance on how to apply both the concept and the model.

5.2.2 User interface is the subject of usability, not “software” and “hardware”

The distinction between hardware and software is not that important as has formerly been assumed in the usability design of interactively used products. Product attributes of hardware and software components need to be combined in such a way that a partial optimization is avoided. Users of interactive products do not differentiate between software and hardware. All elements of the interactive products that allow manipulation and/or provide feedback are part of the interaction and dialogue, independent of them being hardware or software.

An issue of optimization, however, is the user interface, which is both hardware and software in relation to the contextual needs. As it is intended to extend the scope of application of ISO 9241 beyond computerized office tools to all kinds of interactive products, the key issue of design will be the user interface, which should become the subject matter of ISO 9241. ISO 9241 as a multi-part standard for user interfaces of interactive systems should take into account that it may be one thing to e. g. determine the minimum size of displayed characters; however, it may be very important to also consider another thing, which is the legibility of characters under given contextual conditions. Hence, the user interface designer should be familiar with both hardware and software standards.

ISO/TC159/SC4/WG5 should claim the responsibility for any standardization of user interfaces of interactive products. Otherwise there is the risk of being faced with

competitive and perhaps immature standards developed outside of SC4 (e. g., ISO WD 2882, design for “every day products”).

5.2.3 Syntactic versus semantic recommendations

A clarification within the hardware parts of ISO 9241 is required concerning the fact that hardware-related standardization typically focuses on form (i. e. syntactic design recommendations), whereas the software parts also cover meaning (compare with the linguistic model below, section 6.7, the notion of interaction).

For example, recommendation 5.21 within ISO 9241-3 (“Areas coded by luminance only shall differ in display luminance with respect to each other by a ratio of at least 1,5:1”) can be associated to the principle of “Discriminability” of ISO 9241-12 which is on a syntactical level (see section 6.7).

As far as syntax (form) of information is concerned, principles of information design as provided in ISO 9241-12 need to be considered in hardware-related standards of ISO 9241. The recommendations in these standards should point to the design principles of ISO 9241-12, in order to direct the designer of hardware attributes to the underlying design principles, thereby making aware of the contextual needs to be considered when applying design principles (compare with figure 1, section 4.1.1, which also holds for the information design principles given in part 12).

5.2.4 Action items for the revision of ISO 9241-12

As previously mentioned, there is a need to provide cross-references between hardware and software standards of ISO 9241. Currently, ISO 9241-12 contains a number of static user interface recommendations which typically can be found in style guides. Just as the hardware attributes of a display (ISO 9241-3) should be related to the underlying design principles of part 12, so should the typical style guide design items of part 12 do themselves. The designer should always be made aware of information design principles when selecting a recommended display attribute from the standards. Thus the framework for an application of ISO 9241-10 can also be adopted in part 12.

Moreover, the design of display attributes should become more related to the implied needs within the context of use. Designers who are familiar with checklist items of

style guides should be advised to adapt the design process in such a way as to consider first the user's flow of task and other contextual needs before creating the appropriate information items for the display. Prior to information design is task and dialogue design. Adhering to this process will guide the designer to first consider dialogue principles such as self-descriptiveness or conformity with user expectations. Subsequently, the designer should be guided to relate the dialogue principles to specific information design principles, e. g., conciseness or discriminability. These design principles are currently named “characteristics of presented information”, which is misleading, since a characteristic of information coding may be a colour that contributes to the design principle of discriminability. To sum up, principles of dialogue design (ISO 9241-10) should become more clearly related to the principles of information design (ISO 9241-12). In doing so, it can be suddenly recognized that “comprehensibility” should be removed from the current list of information design principles, since comprehensibility covers semantic aspects, whereas other principles in part 12 cover syntactic aspects only. Comprehensibility is already included in the principle of “self-descriptiveness” in part 10 and therefore is redundant in part 12.

5.2.5 Action items for the revision of ISO 9241-13

ISO 9241-13 in its current form has shown to be a misleading standard. It defines “user guidance” as some extra information outside the regular user-computer dialogue, thus suggesting that some extra guidance should be provided to the user, especially when the user is stuck or got lost in the dialogue. However, if an interactive product complies with the design principles laid down in ISO 9241-10 and 12, there is no need for any extra user guidance beyond the dialogue required by these principles.

When the principles of information design (ISO 9241-12) are satisfied, the user will obtain sufficient action guiding information, also referred to as affordance (Norman, 1988). Guiding information should be inherently given to the user, thus all user guidance should be a natural part of the user interface.

When principles of dialogue design (ISO 9241-10) are satisfied, the user is allowed to request additional information when the information presented is not intuitively understandable. In particular, when a product is self-descriptive and conforms with users' expectations, then every user will be guided efficiently.

Of course, it may happen that a user encounters a troublesome situation, which is caused by a mismatch of intentions between user and designer. Even then, however, the principles of suitability for learning and error tolerance (ISO 9241-10) cover all kinds of additional dialogue steps in order for the user to manage the situation or to be effectively prevented from making errors.

The danger of the current ISO 9241-13 is that it can be easily misunderstood as a “checklist” of recommendations to be used to compensate for problems in dialogue design. If applied without consideration for the dialogue principles, the recommendations of ISO 9241-13 can lead to “partial optimizations” that actually reduce the overall usability of a product. Therefore, designers should not be encouraged to design for extra user guidance but for self-descriptiveness. In the worst case, designing for extra user guidance will result in a talkative user interface, which hinders the user from working efficiently. Consequently, extra use guidance is not an appropriate substitute for poor user interface design.

Additionally, the current title “User guidance” suggests that ISO 9241-13 is the main usability standard regarding user guidance. Therefore, designers might prefer to adhere to this standard rather than to ISO 9241-10, which certainly results in poor ergonomic design.

During revision of ISO 9241-13, three aspects should be fixed:

1. The title should be changed to reflect the true content of the standard. A title like “Common aspects of user guidance” would communicate better that ISO 9241-13 contains those aspects of user guidance that are more specific than principles (ISO 9241-10 and -12) but not necessarily tied to a specific dialogue technique (ISO 9241-14 to -17).
2. The definition of “user guidance” should clarify that user guidance is a normal part of the user-computer dialogue and that therefore all recommendations in ISO 9241-13 have to be related to the dialogue principles of ISO 9241-10.
3. The individual recommendations should be related to the appropriate dialogue principles so that the designer is always made aware of the higher

level of the dialogue principles when selecting a recommendation from ISO 9241-13. This avoids user guidance to become a “game of its own” where extra guidance can e. g. increase the number of task steps, thus violating the dialogue principle “suitability for the task” (ISO 9241-10).

Recommendation of ISO 9241-13	Corresponding principle of ISO 9241-10 or ISO 9241-12
5.2.2 If system-initiated user guidance messages are no longer applicable to the current system state or user actions, the information should be removed from the display.	Part 10: Suitability for the task
5.2.3 User-initiated guidance should stay under the control of the user.	Part 10: Controllability
5.2.6 Distinctive message or coding techniques should be consistently used to alert users to conditions that require special attention.	Part 12: Detectability
5.3.7 User guidance should use terminology that the user population typically uses to perform its tasks.	Part 10: Self-descriptiveness
6.2.1 Prompts should indicate implicitly or explicitly the types of input that will be accepted by the dialogue system.	Part 10: Self-descriptiveness
9.2.3 If system failures can be anticipated, an indication of the potential problem should be provided before the failure occurs.	Part 10: Error tolerance
9.5.7 The system should allow users to move error messages, if they are likely to obscure relevant task information.	Part 10: Controllability
9.5.3 Error messages should convey what is wrong, what corrective actions can be taken, and the cause of the error.	Part 10: Error tolerance
9.5.8 Error messages should be presented as quickly as possible after a task-related unit of input has been entered.	Part 10: Conformity with user expectations
10.2.4 System-initiated on-line help should be non-intrusive.	Part 12 Conciseness

TABLE 4 Example recommendations of Part 13 assigned to principles

5.2.6 Action items for the revision of ISO 9241 parts 14 - 17

The standards devoted to the design of dialogue techniques (ISO 9241 parts 14 - 17) are regarded as well laid down and should be kept in their current form and content. As previously recommended, the application framework for ISO 9241-10 needs to be inserted into ISO 9241 parts 14 - 17. The application framework explains the relationship between a specific recommendation pertaining to a dialogue technique and a generic recommendation (see figures 3 and 4). Moreover, the framework explains how to consider the conditions given in the context of use. The currently used concept of "conditional" recommendations can then be dropped, since almost every specific recommendation given in the standards needs to be checked for applicability. As the application framework suggests, the applicability depends on both the implied needs in the context of use and the conformity with an appropriate design principle.

6 Required terms and definitions

Usability engineering is an interdisciplinary approach. To facilitate communication among practitioners, it is indispensable to agree upon the definition of terms. Standards are an excellent means to establish definitions. However, for understanding the definitions it might be necessary to know the concepts which the definitions rely on.
From experiences with projects in usability engineering, we know that the following terms are important candidates:

1. Context of use
2. Usability
3. Dialogue
4. Dialogue step
5. Dialogue principle
6. Generic recommendation
7. Interaction
8. Interactive system
9. User interface
10. Usability requirement

Definitions will be introduced for each of these terms. Additionally, each definition will be explained in view of the underlying concept to explain the content of the definition. Notes will be provided to give further advice for the application of the term.

6.1 Context of use

Users, tasks, equipment (hardware, software and materials), and the physical and social environments in which a product is used (ISO 9241-11:1998).

Underlying concept
In the past, the usability specialist's common phrase was: "Know the user". However, this turned out as too psychological a focus. While establishing usability research in the eighties, the focus expanded towards: "Know the user's task." However, even this

focus was still too narrow, since the initial understanding of a task was that of the computer user's actual flow of dialogue steps with the system (Card et al, 1983). This focus caused a methodological bias referred to as “immunization trap”. Existing system features determined the specification of the user's task requirements. Usability requirements and product evaluations thus included circular statements. In response to this mistake, the focus was expanded from the user's task to the system's context of use. The quality of the product was set relative to the context requirements. Context-of-use analysis became a state of the art concept in usability engineering (ISO 9241-11, 1998). Nowadays, the saying is: "Know the product's context of use." The user is part of this context as well as the user's task. An unbiased tool for describing the context of use is the language of the user, also referred to as "context scenario" (Dzida and Freitag, 1998, DATech 2002).

Note

A context-of-use analysis according to ISO 9241-11 should not be confused with a business specification. An analysis of the context of use is aimed at understanding the implied needs in the context, so as to derive valid usability requirements. A business specification is aimed at a data model, a business model, a functional model and organizational issues to be dealt with in system design. A business specification is very useful *before* analyzing the context of use.

6.2 Usability

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).

Underlying concept

The efficiency of user performance is taken as an indicator of product quality. Efficiency is a measure of user effort spent to achieve a result effectively. The effectiveness of user performance is enabled by the product’s functionality and its non-functional quality characteristics such as reliability, maintainability, etc. From this concept, it follows that usability is the ultimate quality objective in a quality model (Dzida, 2003).

Note

It occurs that the efficiency of user performance is measured in terms of time. To avoid confusion with productivity measures, efficiency should be measured in terms of user effort, for instance, the number of dialogue steps necessary to prepare the execution of a task or the number of dialogue steps needed for error recovery, etc. These indicators of user efficiency also hold for measuring specific dimensions of user efficiency, for instance, suitability for the task or error tolerance (see also dialogue principles).

6.3 Dialogue

Actions exchanged between user and system as a sequence of input and output which embody the meaning (semantics) of information to achieve a goal.

Underlying concept

In the late seventies, it was a technologically innovative property of a system to become controlled by a user in terms of a language (programming or command language). It was the first time in the history of human being's tools that the interaction between a user and a tool became dialogue-like. Hence, there was an urgent need to understand the concept of dialogue, especially, its constituting dimensions (see dialogue principles).

Note

The terms “dialogue” and “interaction” should be distinguished. For more information, see the concept underlying the definition of “interaction”.

6.4 Dialogue step

A user action aimed at achieving a task result beginning with an evaluation of the current system status, continuing with preparatory steps necessary to be ready for data input or command input, followed by the data input or command input. and ending with evaluating the response of the system after an execution of the command by the system.

Note

In case of a mismatch, a dialogue step may come to an end before submitting a command to the system. System response is then evaluated by the user as an unexpected system status or an error message, or system break down, etc. Therefore, the definition of dialogue step holds for error free dialogues, but may also hold for a user's task aimed at error recovery or mismatch management.

Underlying concept

In an error-free dialogue, the user achieves the required task result in one or more dialogue steps. Generally, the user starts with an evaluation of the current status of the system before going on to prepare the conduct of the task. If the current system status corresponds with the user's expectations, the user can go on planning the dialogue step. An evaluation is also necessary after having executed a task or a task step. Therefore, evaluation is an element of the user's behaviour that is modelled as an intersection of two adjacent dialogue steps (figure 11). - When the user does not exactly know how to continue the dialogue, he typically selects the most plausible information for going on. Such a trial attempt can be called mismatch management, since there is a mismatch between the user's intention to do something and the information given by the system. This kind of behaviour can be observed when the user is going to prepare the execution of a task. An incorrect execution causes a so-called error. The model of the dialogue step (figure 11) also holds as a description of special classes of dialogue steps, which are the user's mismatch management or error recovery behaviour.

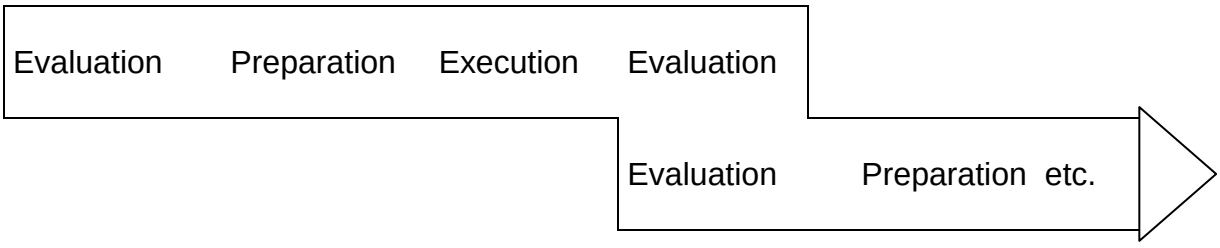

FIGURE 11 Model of the user's task flow within a dialogue step

6.5 Dialogue principle

A fundamental guidance that a design of dialogue or information should generally adhere to regardless of the design technique (ISO 9241 Parts 14-17) applied.

Underlying concept

In the late seventies, an empirical analysis of important properties of dialogue systems was aimed at clarifying the concept of dialogue. The empirical data for generating the dimensions were based on judgements of users who set values on rating scales for estimating the importance of statements about properties of dialogue systems. The principles derived from these empirical dimensions became well-recognized categories of ergonomic quality of interactive systems. Originally, the dialogue principles have been claimed as being statistically independent dimensions (Dzida et al., 1978). It is vital to understand what the dimensions semantically represent before the frequently asked question concerning the overlap of dimensions can be answered.

Note

Dialogue principles are also referred to as ergonomic design principles or principles of software ergonomics (EU Council Directive 90/270/EEC, 1990). To avoid confusion, the principles should not be called criteria. In usability quality assurance, the notion of criterion is for specifying a test criterion.

6.6 Generic recommendation

A rule associated with a dialogue principle to express a usability requirement in an abstract manner independent of a specific context and dialogue technique.

Underlying concept

To prevent the dialogue principles in ISO 9241-10 from causing an arbitrary proliferation of dialogue requirements, the generic recommendations of the standard are specified in a way as to cover a variety of products and contexts of use as well. This property of the generic recommendations, however, entails the burden of applying the recommendations with regard to context-related requirements (see the application framework in section 4.1).

Note

A generic recommendation is not a usability requirement. For specifying a usability requirement, the context of use (ISO 9241-11) needs to be analyzed. A generic recommendation may help to recognize a usability requirement in the context of use (context-related usability requirement).

6.7 Interaction

Actions exchanged between user and system as a sequence of input and output which embody the form (syntax) of information to achieve a goal.

Underlying concept

Practitioners recognized that some recommendations of ISO 9241 parts 13-17 do not correspond directly to dialogue principles. The reason is because some recommendations of parts 13-17 address the form of communication rather than the meaning. According to the linguistic model of user interfaces (figure 12), dialogue principles only cover the meaning whereas principles of information design (ISO 9241-12) cover the form.

<table>
<tr><td rowspan="2">Meaning</td><td>Conceptual level</td><td rowspan="2">Semantic level</td><td rowspan="2">Dialogue</td></tr>
<tr><td>Functional level</td></tr>
<tr><td rowspan="2">Form</td><td>Sequencing</td><td>Syntactical level</td><td>Interaction</td></tr>
<tr><td>Binding</td><td>Lexical level</td><td>Echoing</td></tr>
</table>

FIGURE 12 Organization of the user-interface layers into the linguistic model

Example of meaning: The user should be able to obtain on-line help related to prompts that the user does not understand (ISO 9241-13, rec. 6.2.4). This recommendation pertains to the dialogue principle of self-descriptiveness, i. e. explanation is given to the user on demand.

Example of form: Prompts for data entry should be displayed in a standard location next to the entry field (ISO 9241-13, rec. 6.2.6). This recommendation pertains to the information design principle of consistency, i. e. the placement of information is adjacent to the entry field throughout the application.

Note

Considering the linguistic model of user interfaces may also help to clarify the relationship between ISO 9241-10 and ISO 9241-12.

6.8 Interactive system

Combination of hardware and software components that receive input from, and communicate output to, a human user in order to support his or her performance of a task (ISO 13407:1999).

Underlying concept

The computerization of offices started with the advent of dialogue systems in the late seventies. Visual display units appeared on the desktops of office clerks who increasingly used application programs such as text processors or databases. Today's consumers, however, are additionally faced with various innovative products which are interactive in the sense of being controllable or even programmable by means of buttons and LCD displays. Such interfaces allow the user to do more than just turning a device on or off as with many traditional products. For example, an electric whiteboard allows printing texts written by a marker, or electric coffee makers allow users to program the number of cups to be made at a certain hour according to the habits of the family in the morning. Looking at this trend from a usability perspective, it is appealing that interactive products provide more comfort in accomplishing tasks while some of them even enable innovative behaviours which were not possible or were difficult to accomplish before. However, it can easily be shown that with every button and display added to a product, the risk increases that the users might have trouble to operate the product according to their needs. Interactive products are not only software products or consumer hardware products. Even for the design of machinery, it is important to design for usability, since unexpected operation of machinery can cause severe accidents.

Note

The formerly used term “dialogue system” will be replaced by the more appropriate term “interactive system” in order to cover all kinds of products which can be used with the help of a user interface.

6.9 User interface

All attributes of an interactive system (software or hardware) that contain necessary information and/or need to be manipulated by the user according to conventions of use in order to accomplish specific tasks with the interactive system.

Underlying concept

A number of human-computer interaction reference models have been published; for a survey see Spring et al. (1991). Some of the reference models are devoted to a functional specification, some describe system architectures, and others provide conceptual models (Norman 1988) for users. Conceptual models include a specific interface model, layer models (Fähnrich and Ziegler 1985) and linguistic models of interaction (Moran 1981; Myers 1989; Marcus van Dam 1991). The interface model was suggested by the International Federation for Information Processing [IFIP] Working Group 6.5 (message handling systems); for a formalized description see Dzida (1987, 1988). The user's interaction with a system is enabled by an interface which has been structured into three interface components, each representing a separate aspect of interaction: input/output; the conduct of dialogue; and access to tools, services or data. This structure can be amended by a fourth component for involving a specific type of interaction: information exchange in an organization or within a computer network (see the organizational interface).

Note

Information designers of user interfaces tend to confine their concept of a user interface to only one component (i. e. the input/output interface), which is characterized by the surface of an interactive system. Surface characteristics are facilities of data input as well as attributes of data presentation (such as grouping and coding of data, or echoing keystrokes and mouse clicks). The software designers' conceptual models, e. g. the MVC model (Goldberg 1990) or the PAC model (Coutaz 1987) do

not necessarily fit well with the conceptual model of a user interface in usability engineering.

6.10 Usability requirement

A required user performance specified in terms of the product's context of use and the implied needs in the flow of the user's task.

Note: When specifying a usability requirement, a generic recommendation derived from a dialogue principle can be applied to assist the specification process, where appropriate.

7 References

Berry, D. 2000. The user experience. The iceberg analogy of usability. IBM home page. http://www-106.ibm.com/developerworks/library/w-berry/

Card, S.T., Moran, T.P., & Newell, A. (1983). The Psychology of Human-Computer Interaction. Hillsdale, NJ: Erlbaum.

Coutaz, J. (1987). PAC, an object oriented model for dialog design. In INTERACT'87 Conf. Proc. Human-Computer Interaction, H.J. Bullinger and B. Shackel, eds., pp. 431-436. Elsevier, Amsterdam.

DATech (2002). Prüfhandbuch Gebrauchstauglichkeit. Leitfaden für die Evaluierung von interaktiven Produkten auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11. http://www.datech.de/download/

DATech (2003). Prüfbaustein Usability-Engineering-Prozess. Leitfaden für die Evaluierung des Usability-Engineering-Prozesses bei der Herstellung und Pflege von Produkten auf der Grundlage von DIN EN ISO 13407. http://www.datech.de/download/

DIN 66234, Teil 8 (1988). Bildschirmarbeitsplätze - Grundsätze ergonomischer Dialoggestaltung, Beuth-Verlag, Berlin.

Dix, A., Abowed, G., Beale, R., and Finlay, J. (1998). Human-Computer Interaction. Prentice-Hall, Englewood Cliffs, N.J.

Dzida, W. (1987). On tools and interfaces. In Psychological Issues of Human Computer Interaction in the Work Place, M. Frese, E. Ulich, and W. Dzida, eds., pp. 339-355. North-Holland, Amsterdam.

Dzida, W. (1988). Modellierung und Bewertung von Benutzerschnittstellen. Software Kurier 1:13-28.

Dzida, W. (1995). Standards for user interfaces. Computer Standards & Interfaces 17:89-97.

Dzida, W. (2003). Applying International usability standards. In A.B. Tucker (ed.), The Computer Science and Engineering Handbook. CRC Press, Boca Raton, in press.

Dzida, W., Herda, S., and Itzfeldt, W.-D. (1978). User perceived quality of interactive systems. IEEE Transactions on Software Engineering, SE4(4):270-276.

Dzida, W., & Freitag, R. (1998). Making use of scenarios for validating analysis and design. IEEE Transaction on Software Engineering, 24(12), 1182-1196.

Dzida, W.; Hofmann, B.; Freitag, R.; Redtenbacher, W.; Baggen, R.; Zurheiden, C.; Geis, T.; Beimel, J.; Hartwig, R.; Hampe-Neteler, W. und Peters, H. (2000). Gebrauchstauglichkeit von Software. ErgoNorm: Ein Verfahren zur Konformitätsprüfung von Software auf der Grundlage von DIN EN ISO 9241 Teile 10 und 11. Schriftenreihe der Bundesanstalt für Arbeitsschutz und Arbeitsmedizin, Forschung F 1693, Dortmund.

Dzida, W. and Geis, T. (2003). User requirements - What they are and how to specify them. In: H. Bons and S. Salmann (eds.), Proceedings of the 8th Congress on Software Quality Management, section requirements engineering, track 1.3. Cologne, April 2-4, 2003.

El Emam, K., Drouin, J.-N., Melo, W. (1997). SPICE **-** The Theory and Practice of Software Process Improvement and Capability Determination. New York: Wiley.

EU Council Directive (1990). Council directive on the minimum safety and health requirements for work with display screen equipment (fifth individual Directive within the meaning of Article 16(1) of Directive 87/391/EEC). Number: 90/270/EEC of 29 May 1990. Official Journal of the European Communities, No L 156, p.14 - 18.

Fähnrich, K.-P. and Ziegler, J. (1985). Workstations using direct manipulation as interaction mode - aspects of design, application and evaluation. In INTERACT'84 Human-Computer Interaction. B. Shackel, ed., pp. 693-698. Elsevier, Amsterdam.

Goldberg, A. (1990). Information models, views, and controllers. Dr. Dobb's J. 7:54-61, 106-107.

ISO/IEC 9126 (1991). Information technology - Software product evaluation - Quality characteristics and guidelines for their use.

ISO 9241-10 (1996). Ergonomic requirements for office work with visual display terminals (VDTs): Dialogue principles.

ISO 9241-11 (1998). Ergonomic requirements for office work with visual display terminals (VDTs): Guidance on usability.

ISO 9241-12 (1998). Ergonomic requirements for office work with visual display terminals (VDTs): Presentation of information.

ISO 9241-13 (1998). Ergonomic requirements for office work with visual display terminals (VDTs): User guidance.

ISO 9241-14 (1997). Ergonomic requirements for office work with visual display terminals (VDTs): Menu dialogues.

ISO 9241-15 (1997). Ergonomic requirements for office work with visual display terminals (VDTs): Command dialogues.

ISO 9241-16 (1999). Ergonomic requirements for office work with visual display terminals (VDTs): Direct manipulation dialogues.

ISO 9241-17 (1998). Ergonomic requirements for office work with visual display terminals (VDTs): Form filling dialogues.

ISO 13407 (1999). Human-centred design processes for interactive systems.

ISO TR 18529 (2002). Human-centred lifecycle process descriptions.

Jacobson, I. (1995). The use-case construct in object-oriented software engineering. In J.M. Carroll (Ed.): Scenario-Based Design (pp. 309-336). New York: John Wiley.

Marcus, A. and van Dam, A. (1991). User-interface developments for the nineties. Computer, 24(9):49-57.

McCall, J.A., Richards, P.K., and Walters, G.F. (1977). Factors in software quality, Vols. I,II,III. U.S. Rome Air Development Center Rep. NTIS AD/A - 049014, 015, 055.

Microsoft Corporation (1999). Microsoft Windows User Experience. Microsoft Press, Richmond, WA.

Moran, T.P. (1981). The command language grammar: a representation for the user interface of interactive computer systems. Int. J. Man-Machine Stud. 15:3-50.

Myers, B.A. (1989). Encapsulating interactive behaviours. In Proc. CHI'89 Conf. pp. 319-324. ACM Press, New York.

Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In Proc. CHI'92 Conf. J. Bennett and G. Lynch, eds., pp. 373-380. ACM Press, New York.

Nielsen, J. and Mack, R.L. eds. (1994). Usability Inspection Methods. John Wiley and Sons, New York.

Norman, D.A. 1988. The Psychology of Everyday Things. Basic Books, New York.

Paulk, M.C., Weber, C.V., Curtis, B., and Chrissis, M.B. (1994). The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley, Reading, MA.

Scapin, D.L. and Bastien, J.M.C (1997). Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behaviour & Information Technology, 16(4/5), 220-231.

Shneiderman, B. (1998). Designing the User Interface. Addison-Wesley, Reading, MA.

Spring, M.B., Jamison, W., Fithen, K.T., Thomas, P.M, and Pavol, R.A. (1993). Models for a human-computer interaction. In Encyclopedia of Microcomputers, A. Kent, J.G.Williams, eds. Vol 11, pp.189-218. Marcel Dekker, New York.

Stewart, T.F.M. (1990). SIOIS - standard interface or interface standards. In D. Diaper et al, (eds.): INTERACT'90 Human-Computer Interaction, pp. xxix-xxxiv, Elsevier, Amsterdam.

Travis, D. 1997. Why GUIs fail. Home page of System-Concepts Ltd., http://www.system-concepts.com/articles/gui.html