NOMADIC MEDIA

(ITEA if02019)

User-Centred Design

Guidelines for Methods and Tools

VTT Information Technology

University of Oulu, Dept. of Information processing science

Philips Research

Philips Applied Technologies

Authors

- VTT Information Technology: Juhani Heinilä (Ed.), Hanna Strömberg, Jaana Leikas, Veikko Ikonen
- University of Oulu, Dept of Information processing science: Netta Iivari, Timo Jokela, Kari-Pekka Aikio, Ilari Jounila
- Philips Research: Jettie Hoonhout
- Philips Applied Technologies: Natasja Leurs

Contact information

Juhani Heinilä, (Ed.)
VTT Information Technology
P.O. Box 1300,Tampere
Sinitaival 6,
FIN-33101 Tampere
FINLAND
Tel. +358 20 722 111, fax +358 20 722 3380
Email: juhani.heinila@vtt.fi

Executive summary

The vision of the Nomadic Media project was to combine people, technologies and services in a way that allows consumers to enjoy content and interactive services at the times and in the places they prefer, using the devices that best suit their context, at home or on the move. This challenging vision was tackled by the Nomadic Media project consortium[1] of several European companies, research institutes and universities within the ITEA programme[2] (ITEA = Information Technology for European Advancement).

The purpose of User-Centred Design (UCD) is to involve end users in the development process of the product or system in a way that the prototypes and designs, and finally the products or systems would meet the needs and requirements of the users as well as possible. The UCD work is multidisciplinary, joining together the expertise of different stakeholders, such as UCD experts, engineers and graphical designers. In the Nomadic Media project, different organisations participated in UCD. These were VTT Information Technology, Department of Information Processing Science of Oulu University, Philips Research and Philips Applied Technologies. By focusing on the needs of the technical work packages in the project this UCD team carried out various UCD activities during the project. The authors believe that this work has proved to have a positive effect both on the designs and demonstrations of the project and on the knowledge and understanding of the importance of UCD among the technical people working in the project.

This guide is the final deliverable of UCD work package (WP4) in the Nomadic Media project. The report is produced to be utilised beyond the Nomadic Media project by the product or system development teams of European ICT industries intending to reach an enhanced level of usability and user acceptance of the nomadic interactive systems and devices to be developed. The characteristic features for those kinds of projects include issues such as context awareness and development of multi user applications for various device platforms, multi location based systems and systems supporting privacy, security and portability aspects.

The general structure of this guide follows step by step the general UCD process model. In each phase, the purpose of the step and the commonly used methods are discussed. Furthermore, related to these steps the experience and lessons learnt in the Nomadic Media project are given. Each step ends up with a relevant reference list concerning the step.

The guide at hand consists of seven chapters which are summarized as follows:

Chapter 2: User-Centred Design deals with the concept of UCD, user experience and scenario-based design. The purpose is to clarify what UCD is and why it is important. Usability and other user experience attributes are presented. Principles and different phases of scenario-based design are highlighted.

Chapter 3: The iterative User-Centred Design Process is the main chapter of the guide. First, it deals with the general UCD process model. Secondly, the process is divided into six steps, each of them is analysed and the appropriate methods to be used in each step are discussed.

[1] http://www.hitech-projects.com/euprojects/nomadic-media/index.htm;

[2] http://www.itea-office.org/index.php;

Chapter 4: Summary of methods is presented in a matrix format including the comments on the phasing of each method in the UCD process lifecycle and principal references.

Chapter 5: User-Centred Design in Organisational Context describes the problems faced when initiating UCD practices into organisational level among industry. Furthermore, an overview of UCD literature providing guidance for introducing UCD in organisations is given.

Chapter 6: A reference list including relevant UCD articles and websites is presented.

Key Words user-centred design (methods), UCD process, usability, user requirements, scenarios, scenario-based design, visualisation;

Contents

List of abbreviations

ANSI	The American National Standards Institute
FPP	Full Project Proposal
HCD	Human-Centred Design
HCI	Human-Computer Interaction
IEC	The International Electrotechnical Commission
ISO	International Standardization Organization
ITEA	Information Technology for European Advancement
LAN	Local Area Network
POC	Proof-Of-Concept
SBD	Scenario-Based Design
UCD	User-Centred Design
QOC	Question, Options, Criteria - method for initial requirements capture and prioritising
WP	Workpackage

1 Introduction

User-Centred Design (UCD) is a design approach, adopted by many industries in order to develop products and services that will meet the needs and expectations of the end users. However, implementing a UCD approach is not trivial, and many industries still appear to experience difficulties in effectively applying UCD methods in their development processes. Traditionally, the focus of applying UCD in industry has been on the development of systems to be used in business and occupational settings instead of the development of products and services directed to be used by consumers. This situation appears to be changing together with the development of ubiquitous products and services, and therefore there is a need to clarify the UCD process and make the UCD methods more accessible for companies, especially when designing systems and products for the consumers.

This document provides a summary of a simplified set of methods and tools for implementing UCD for the companies working in the Nomadic Media type of projects. The document has been realised by combining the experience and research results gained in the Nomadic Media project, expertise of the specialists in the project, literature reviews, relevant reference lists and different web sites.

2 User-centred design

What is user-centred design?

User-centred design (UCD) is an approach to interactive system development that focuses specifically on making systems or applications easy to use (ISO/IEC 1999). Adopting this approach we end up with products that

- help users achieve their goals,
- people want to use,
- people can understand how to use,
- can be successfully used,
- will satisfy the users, and
- will prevent users from making errors.

User-centred design aims at understanding the needs of users early in the design and development process, providing guidance for designing product that will meet user needs, and assuring that the product will be accepted by the users. The process of user-centred design is composed of a number of key activities (ISO/IEC 1999; McClelland et al. 2005):

- understanding and specifying the user requirements,
- understanding and specifying the context of use,
- iteration of design solutions, and
- evaluating the solutions with the users against requirements.

On the other hand, UCD is not a self-contained process. It must be considered in a broader organisational picture fitted into business and technical development processes. Thus, in addition to the end-users the opinions, e.g., of the user organisation, the service providers and the society as a whole has to be taken into account.

The disadvantages of failing to be user-centred include:

- unwanted, underused products,
- confused and frustrated users,
- angry customers,
- damage to brand,
- lost sales, and
- high user support costs.

What is user experience?

User experience is a holistic term used to describe the overall experience a user has when using a product or a system. The user experience research focuses on the interactions between people and products/services, and the experience resulting from the interaction. The experience of even simple artefacts does not exist in a vacuum but, rather, in dynamic relationship with other people, places and objects (Buchenau et al. 2000). Defining the user experience can extend to concern all aspects of experiencing the product or service, including physical, sensitive, cognitive, emotional, and aesthetic relations (Kuniavsky 2003).

User experience strengthens the role of time, context, meaning and emotions in using products or services. Also shared experiences (e.g. co-experience, Battarbee 2005) and social interaction are lately taken into the discussion (Leikas et al. 2005; Leikas et al. 2006).

Different user experience attributes are presented in the following.

Usability

Usability can be defined as the measure of the quality of a user’s experience when interacting with a product or service. According to the international standard ISO 9241-11 (Guidance on usability, ISO/IEC 1998) usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.

The key terms are further defined as follows:

- effectiveness: the accuracy and completeness with which users achieve specified goals;
- efficiency: the resources expended in relation to the accuracy and completeness with which users achieve goals;
- satisfaction: freedom from discomfort, and positive attitude to the use of the product;
- context of use: characteristics of the users, tasks and the organizational and physical environments.

Usability is recognized as one of the most important characteristics of systems and products. Usable systems are easy to learn, efficient to use, not error-prone, and satisfactory in use (Nielsen 1993). High usability brings many benefits for the user and possible other stakeholders of the product. It means, e.g., “ increased productivity, enhanced quality of work, improved user satisfaction, as well as reductions in support and training costs." (ISO/IEC 1999).

In fact, usability can not be defined in absolute terms since it always depends on who the users are, what are their goals and the environment of use. Nowadays, when the focus is shifting more and more from traditional desktop computing to ubiquitous computing applications (that are available everywhere and all the time) a very heterogeneous group of users’ everyday practises have to be understood and supported in the product design (Abowd et al. 2002). Especially concerning consumer and entertainment products and applications the definition of usability often expands to cover also other factors of user experience to be discussed in the following.

User satisfaction and user acceptance

User satisfaction may not always be dependent on the usability of a product. Sometimes users do not mind even major usability problems; in other cases usability problems may lead to severe user dissatisfaction. In some cases it might be necessary to include all the features in the product that the user hopes for, in order to avoid user dissatisfaction and to gain user acceptance (Jokela 2004).

Easy to learn

At its best a product is so intuitive to use that it requires no training to use it. However, this is usually difficult to achieve in the product design. In any case the product should be so easy to learn that the resources to use the product effectively and efficiently are minimal.

Enjoyable

Besides usable, many product and devices should be enjoyable, entertaining and engaging. This concerns especially home electronic devices and entertainment products. Apart from wanting to provide users an enjoyable experience, another reason to make the interaction enjoyable can be to motivate users to explore the capabilities of (complex) devices, and thus in a playful way teach the user to learn how to use the system.

Transparency

The interaction with devices (e.g. home electronics) should be transparent. This means that users are able to perceive the most effective and efficient ways to complete their tasks successfully at a glance without having to consult procedural instructions. Transparency is the ideal relationship between the user and a disappearing tool (or product) (Rutkowsky 1982).

User in control

Probably many scenarios will feature some kind of automated behaviour. Concerns about the amount of control the user will have over the system should be taken into account. Users should be provided with appropriate feedback about system changes that will give a feeling of being in control. Certainly, at least for a novice user, it might be helpful if initially the system takes control, since the user will not have sufficient knowledge of the functionality of the system and how to operate it. However, the amount of perceived control of the system should be alterable by shaping the way of interaction, either by letting the system or the user take the initiative. For many users it is important to have control over what is happening when they are using a product or a system.

Scenario-based design

Originally scenario as a term can be traced to the early theatre and film terminology. In this context a scenario is a synonym to the screenplay, manuscript, copy or a script. The elements of a scenario are virtually the same both in the original notion and in the scenario definition applied as a method in user-centred product development process. These basic elements include: the actors (users), the scene (context) and the scheme (the story including the background, tasks, goals and action). The scenarios that include these elements and are used in user-centred design, can be more specifically defined as user scenarios, use scenarios, usage scenarios or interaction scenarios. In the following we call them as ‘scenarios’.

Scenarios as design method can be used to:

- describe problems to be solved;
- catalyze interaction within a design team, thus improving team-building;
- facilitate user involvement in the early design;
- support collaborative design where all the participants don’t need to know the technology;
- help in transferring and explaining design ideas;

Advantages of scenarios:

- scenarios help express the requirements of the different stakeholders in a format that can be easily understood by the other stakeholders;
- scenarios encourage the partners in a project to consider the characteristics of the intended users, their tasks and their environment early in the project; also, usability issues can be explored at a very early stage in the design process (before a commitment to code has been made);
- scenarios are the goals against which prototype design and development process are measured;
- scenarios help the persons involved in the process to explore and subsequently determine the problem space and the solution space (including prioritising the problem areas or potential solutions that will be addressed);
- scenarios can become the conceptual description for a demo or prototype system;
- scenarios can also be used to generate contexts for evaluation studies;
- only minimal resources are required to generate scenarios;
- scenarios can be written by participants with little or no prior experience, although the next steps in the scenario usage process (e.g. analysing) will require expertise.

In software engineering, the term "use case" is often used in a similar sense as "scenario" and these terms can easily be confused (see chapter 3 Step 2). A use case specifies the sequences of actions that a system or a subsystem performs when interacting with outside actor(s). Use cases help identify and clarify the functional requirements of the system to be designed whereas scenarios focus on identifying user and contextual requirements. Thus scenarios tend to describe actions on a higher level, not focusing on details of actions like use cases. In user-centred design, scenarios are usually defined in connection to user requirements specification, and they are refined to use cases during the functional requirements definition phase.

As design instruments, scenarios are stories about people and their activities in a particular situations and environments (contexts). They can be textual, illustrated (e.g. picture books or comic strips), acted (e.g. dramatised usage situation) or even filmed (e.g. videos) descriptions of usage situations. They describe current or envisioned work practises or tasks of the users and thus help different stakeholders (including the users themselves) understand the tasks in their contexts, evaluate the practises and suggest changes to these practises in parallel to designing new tools.

Scenario generating aims to predict how people could act in particular situations. That is why it is well suited for designing new product concepts, when the context of use and the intended users are not strictly defined. Scenario building is a flexible and cost-effective way to generate design ideas for new products and to identify the potential user groups and contexts of use for the product. It is desirable to develop and compare several concepts. The most feasible concepts can then be selected to be further elaborated towards user and application requirements specification.

It is often easier and more fruitful to generate scenarios with a group of people rather than individually. That is why the group methods, e.g., brainstorming and group discussions are well suited for scenario building (see chapter 3 Step 1). Optimally the group should include people with different expertise, e.g. designers, end users, application field experts, marketing people and usability experts. Involving end users in the scenario generating and refining process is vital to enable genuine user feedback for the system development as early as possible. However, scenario generation can also start without authentic user participation for instance when the project group first needs to agree on their common vision of the goals of the design.

Start with creating initial scenarios

The value of scenarios is that they make ideas more concrete describing complicated and rich situations and behaviours in meaningful and accessible terms. In this way ideas are easier to analyse and communicate. Scenarios help different stakeholders understand better the implications of particular design solutions especially from the user's point of view.

The main elements a scenario should contain are:

- user group: the characteristics of those who will use, install and buy the devices;
- contextual setting: where and how current or expected conditions could influence device use;
- circumstance: the trigger of using the device (the why of using a device by whom, where and when);
- goals (or motivations) of users and critical success factors;
- artefact: devices that might be used, how these might work, their appearance, and possibly their requirements;

v time frame: when, how long, at what frequency, and in which life time period;

v qualities: the physical form factor of possible devices (e.g., portability, weight).

Scenarios describe individual users in individual usage situations, and are not meant to describe the whole functionality of a system. The value of scenarios is that they concretise something for the purpose of analysis and communication. The concreteness enables designers and users to deal with complicated and rich situations and behaviours in meaningful terms, and to understand better the implications of particular design solutions for performing realistic tasks (Carroll 1995).

Input for the creation of scenarios can come from market research, studying users interacting with existing appliances in their current environments, technology innovations; also, various brainstorming type of activities can provide input for the actual writing of scenarios.

Scenarios are treated (and modified) as a tool for specific context of use analysis, user requirements capture, concept definition and overall user involvement.

Scenario building is a widely accepted way to incorporate and generate design ideas for new products and to identify the possible users and contexts of use for the product to be. It is well suited to the design of new product concepts and to the design of consumer products, where the context of use may vary a lot. Descriptions of people using technology help different participants in discussing and analysing how novel technologies, applications and services could be influencing everyday life of individuals, communities and society. (Rosson et al. 2002)

Analyse your scenarios

After the illustrated scenarios are evaluated with end users the design team (and project partners) will agree upon the application scenarios that will be considered to be implemented in the project.

Then the scenarios will be revised (rewritten) into usage scenarios keeping in mind the functional features that are planned to be introduced in the application or service. These usage scenarios represent a sequence or flow of user actions required achieving a specific task as well as a set of features with relevance to the project's key technology.

After this initial pre-selection process of the scenarios, the scenarios will have to be analysed in more depth, in order to determine user requirements and technical requirements. For this more detailed analysis process, various techniques are available. Both qualitative and quantitative methods can be used, depending on the goals of the evaluation. Qualitative methods give concrete feedback on the user actions presented in the scenarios whereas quantitative methods can be used to rank different scenarios e.g. in terms of credibility or acceptability.

The usage scenarios and the context of use can be analysed together with end users, specifying the tasks to be achieved. The aim of task analysis is to find answers to the following questions: Who are the users? What do they do? Where? Why? For each scenario a set of properties and functionalities that the system should present for the scenario to be feasible will be identified. These elements are task-specific requirements in the scenarios that have to be refined and generalised later in the process. The design should identify all the tasks to be carried out and define which parts of the tasks are taken care of by technology and which parts are the user’s responsibilities. The human functions should form a meaningful set of tasks.

The identified task-specific requirements will now be divided into functional and non-functional requirements. (See chapter 3, Step 4 for more information).

The scenario itself does not (and should not) comment upon the specifications of how a product or service operates, for this would steer the discussions in a particular direction too soon. Instead, the analysis of a scenario provides room for discussion and aligns expectations and mindsets of the different disciplines and stakeholders within a project. The purpose of such an analysis is to reveal how the scenario should be interpreted, resulting in user requirements and technical requirements. The user requirements specify the criteria that should be met by the project result (product and/or service) to make it usable and useful, and generally meet the requirements and needs of the intended users. The technology requirements specify the technological topics that will be worked on (and the technological topics that will not be worked on) to realise parts of the scenario.

QOC method for analysing scenarios

QOC method - Questions, Options and Criteria - is one approach to systematically analyse scenarios, in order to come up with as many different design solutions as possible (this to prevent that the first design solution that comes to mind is chosen, without further exploring alternative solutions that might turn out to be better). QOC is not just about coming up with more than one design solution. The idea is that in addition to that each possible solution is evaluated using a set of criteria. These criteria can be based on accepted usability criteria, on other user requirements, and on technical requirements. QOC could then be the next step, after the scenarios have been selected.

The QOC method has been introduced as a semiformal notation to represent the design space around an artefact (MacLean et al. 1991). The questions in the method highlight key issues in the design while options are the answers to these questions, weighted by design criteria. The user and technology requirements that stem from the scenarios are used to generate design criteria. Options are technological solutions to these questions and are weighted by the design criteria. As a method, QOC supports the generation of design options based on a design rationale. The rationale stems from the comparison of alternative options.

Case Nomadic Media: lessons learnt

Scenario-based design was the key method for gathering user requirements and context of use information in the Nomadic Media project. The participants were instructed to create their scenarios on a uniform scenario template. The scenarios were based on the basic principles, goals and scope of the project. These were (Nomadic Media 2003):

> *Nomadic Media will address the intersection between (1) the enabling and infrastructure technologies, (2) the services and content, and (3) people using entertainment/leisure content and related services in their 'networked homes'. In this context 'home' includes a persons 'physical home' and the 'virtual home' - the personal world a person carries around with them as they move from place to place.*
>
> *Nomadic Media wants to find ways of enhancing the flexibility with which consumers can access services and content where they wish, to use services and content in personalised and configurable ways, and to move content between devices according to need and circumstance. The 'enabling and infrastructure technologies' includes the architecture of*

> *enabling technologies, standards (international and industry) for interconnectivity (wired, wireless, and private LAN), interaction technologies, and access points to entertainment/leisure content and related services. The 'enabling and infrastructure technologies' also includes the use of stationary, portable and mobile devices in both private and shared locations."*

After a combination of numerous scenarios and a rough analysis phase carried out in a small team of stakeholders the initial scenarios were revised and the final combined scenario was written, where the focus was clearly put on four subjects (called clusters): ‘Home’, ‘Airport’, ‘Ski-slope/On the go’ and ‘Health care/Pharmacy’. In the next phase the scenarios were divided into scenario elements from technical and user interaction basis and the elements were assigned to partners according to their technical interests. From now on, the scenario elements were used as a source of innovations and as communication tools in design teams and between the partners when capturing functional and non-functional requirements and specifying the system architectures of the concepts to be created. These goals were supported, e.g., by applying QOC in the scenario analysis and by preparing visualisations of the scenario.

QOC: The benefit of applying QOC (questions, options, criteria) in the scenario analysis is to produce many opinions to the problem at hand, create criteria for prioritising of options, and document this thinking. QOC is a kind of a "structured brainstorming" method, aiming at getting input from many points of view and generating as many potential solutions as possible. In the beginning of the QOC process numerous requirements and open questions and choices raised by the different participants exist. The more choices are made, the more the procedure will lighten as the number of open options will decrease.

The choice of the relevant criteria in the QOC analysis is critical. This clearly affects the results achieved through the QOC analysis. Therefore, it is important to consider how many criteria deal with usability/user experience versus technical feasibility/technical goals of the project.

The QOC method was applied in design team before the user evaluations in order to go thoroughly through the Nomadic Media scenarios. QOC was found to be useful since the questions arisen from QOC method were utilised when designing the user evaluations of the scenario. At this stage we concentrated on questions that were found important to the scenario refinement or from the user interaction point of view.

The QOC analysis succeeded in highlighting unclear issues in the scenario. It revealed the unresolved issues related to users’ actions in the scenario and forced the analysis team to consider these issues, i.e. how a user is supposed to carry out a particular task in detail or how a user is supposed to interact with the device/service in question in a particular situation.

Although the method turned to be rather time consuming, we suggest using QOC as a tool for analysing the scenario and in detail to reveal unclear issues before gathering user feedback of the scenario.

Visualisation of scenarios: The analysis and visualisation of scenarios aim at presenting the scenarios in such a way that the end users can easily understand the scenario contents. In the Nomadic Media project, the analysis before visualisation was carried out with QOC method. The visualisation methods, including an MMV method (MMV=match-stick man visualisation), see chapter 3, Step 2.

Suggested resources for additional reading:

Abowd, G., Mynatt, E, and Rodden, T. (2002): The Human Experience, Pervasive Computing, Vol. 1(1), 48-57.

Battarbee, K. (2004): Co-Experience - Understanding user experiences in social interaction. Academic Dissertation. Publication series of the University of Art and Design. Helsinki A 51.

Buchenau, M. and Fulton Suri, J. (2000): Experience Prototyping. In Proceedings of DIS´00. ACM, Brooklyn, New York.

Carrol, J. M. (Ed.) (1995): Scenario-Based Design. London: John Wiley & Sons.

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

ISO/IEC (1999): 13407 Human-Centred Design Processes for Interactive Systems, ISO/IEC 13407: 1999 (E), 1999.

Jokela, T. (2004): When Good Things Happen to Bad Products: Where arethe Benefits of Usability in the Consumer Appliance Market? ACM Interactions XI.6: pp 28-35.

Kuniavsky, M. (2003): Observing the User Experience: A Practioner's Guide to User Research. Morgan Kaufmann.

Leikas, J. and Strömberg, H. (2005): Novel User Interfaces for Promoting Social Interaction in Multiplayer Games. ENACTIVE/05, November 17-18, Genoa, Italy.

Leikas, J., Strömberg, H., Ikonen, V., Suomela, R. and Heinilä, J. (2006): Multi-user Mobile Applications and a Public Display: Novel Ways for Social Interaction. (Accepted). Pervasive Computing and Communications, Pisa, Italy, 13-17 March 2006.

McClelland, I. and Fulton Suri, J. (2005): Involving people in design. (Chapter in 'Evaluating Human Work 3rd edition'). Editors: J Wilson and N Corlett, Taylor and Francis, London.

MacLean, A., Young, R.M., Bellotti, V.M.E., and Moran T.P. (1991): Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6(3&4), 201-250

Nielsen, J. and Sano, D. (1994). SunWeb: User Interface Design for Sun Microsystem's Internal Web.

Nielsen, J. (1993): Usability Engineering. San Diego, Academic Press, Inc.

Nomadic Media (2003): FPP-Final Project Proposal.

Rosson, M. B. and J. M. Carroll (2002): Usability Engineering. Scenario-Based Development of Human-Computer Interaction, Morgan Kaufmann Publishers.

Rutkowski. C. (1982): An introduction to the Human Applications Standard Computer Interface, Part I: Theory and principles. Byte, 7 (10), October 1982, 291-310.

3 The iterative user-centred design process

Different people may speak about UCD with somewhat different understandings on its exact goals and contents. Widely accepted main principles of UCD are user involvement throughout the process, iteration of design solutions, and cross-functional teamwork, mentioned, e.g., in (ISO/IEC 1999). In the Nomadic Media project, we have illustrated the UCD process with in the Figure 3.1.

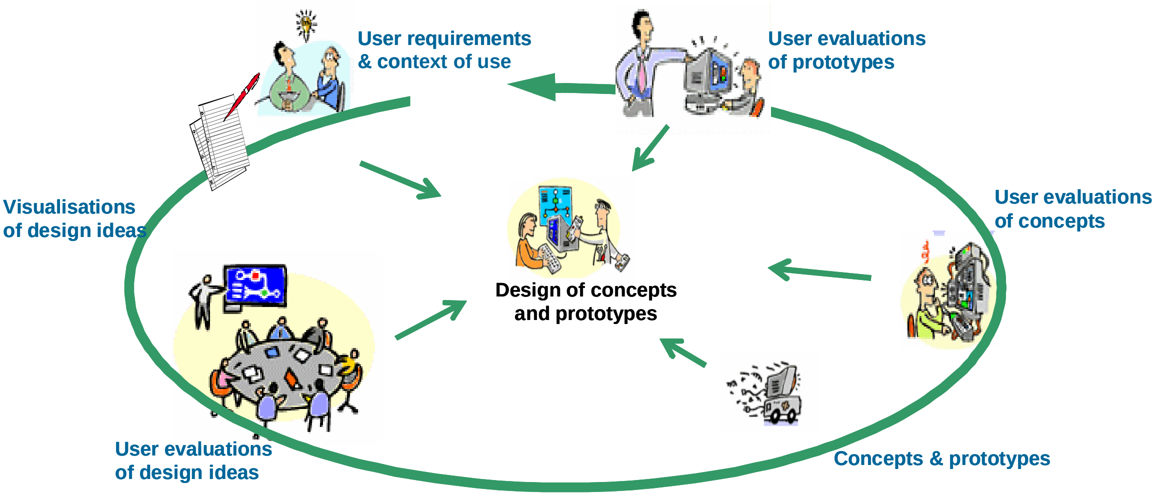

User requirements & context of use
User evaluations of prototypes
Visualisations of design ideas
User evaluations of concepts
Design of concepts and prototypes
User evaluations of design ideas
Concepts & prototypes

Figure 3.1. The iterative User-Centred Design process

User-centred design in the whole product development cycle is an iterative process where feedback from end users is brought into design in different stages of the development cycle. In the following we will describe the different steps of the iterative UCD process. These steps are:

- Step 1: Define user requirements and context of use;
- Step 2: Visualise design ideas;
- Step 3: Carry out user evaluations of design ideas;
- Step 4: Create concepts and prototypes;
- Step 5: Evaluate concepts and prototypes.

Iterative design should be employed throughout the entire product development cycle, as well as within each step of the UCD process. This principle recommends that requirements are collected and the product is designed, modified and tested repeatedly. Usually it is neither possible nor meaningful to go through the development cycle once; one has to continue to iterate and fine-tune with each cycle until the final user requirements are specified with the help of different methods in different steps of the product development process.

For example, interviews may be carried out in the step1, gathering phase of initial requirements. They may be used to begin the user requirements collection. They may, however, also open up new questions that must be answered by running a follow-up activity. This can be e.g. task analysis. The results of the analysis can then be used to go back and revise and refine or iterate the user requirements document (e.g. a design rationale) based on the new data. (Courage et al. 2005)

Requirements specification

The initial user requirements are refined during the design process on the basis of continuous feedback from the users. In most cases, the user requirements cannot all be fixed at the beginning of the project. You have to be prepared to identify new requirements and to refine or reject the existing ones throughout the design process.

The requirements from the user's point of view are specified including the functions required to support the user tasks, the user-system interfaces, user support required, physical and organisational requirements, equipment and hardware. The requirements include usability and quality goals that must be achieved. In practice, the requirements evolve iteratively with steps 2 to 5 during the process (see Figure 3.1).

In user-centred design the user and organisational requirements should:

- define the range of relevant users;
- set user-centred design goals;
- define priorities of the different requirements;
- provide measurable usability criteria;
- be confirmed by the users or their representatives;
- include statutory or legislative requirements;
- be adequately documented;
- be a working document, which will be updated throughout the design process (a design rationale).

When defining the user requirements, the end-users are often not the only actors whose requirements should be taken into account. In addition to the actual end-users you may have to take into account the points of view of the user organisation, the service providers and society as a whole. There will also be business and technical requirements for the system, which will be identified and developed in parallel with the user and organisational requirements.

The functional requirements represent the main features of the future product. All of them may not have an immediate user action but they have an influence on the usage as a whole. The non-functional requirements are goals for quality of use. They are requirements that make the usage of the system easier and more attractive for the user. In other words, the functional requirements indicate what is needed, and the non-functional requirements explain why the requirement is important, how the specific feature is presented to the user or how it should respond to the user's action.

Traditional approaches to requirements engineering concentrate on identifying functional requirements and validating that the developed product meets these requirements. Other non-functional requirements (e.g. efficiency, reliability, usability, maintainability and portability) have had less importance. Yet, from the user's perspective, non-functional requirements may be critical to successful implementation of a new system. A user-centred approach emphasises the importance of obtaining a complete understanding of users' needs and of validating the emerging requirements against potential scenarios of usage. The user-centred methods will help identify and prioritise requirements from the user's perspective.

As the crucial phase in the requirements definition is the transformation of informal descriptions (e.g. scenarios) to formal application requirements, and finally to the features of a system, it is essential to collect the user requirements in to a form that is understandable for all members in the design team. The design team should keep, e.g., a diary of the design decisions, a design rationale, where the requirements are updated based on requirements capture work and feedback from user trials. The design rationale may contain the requirements that have to be taken into account in order to develop a successful product or service with high usability (Leikas et al. 2005).

The design rationale can also be a change history illustrating all the thinking and iteration behind each design decision. It may cover what was decided, by whom and when in the design. Thus it is possible to follow, e.g., why a specific requirement or a feature was left out in the implementation although it was at first taken into consideration. As the end-user opinions about the product or system often evolve during the course of the design, the decisions should be made accordingly. The decisions of keeping specific requirements in the design and decisions and reasons of dropping some others out should all be documented in the design rationale.

Suggested resources for additional reading:

Courage C. & Baxter K. (2005): A practical guide to user requirements. Methods, Tools & Techniques. Morgan Kaufman Publishers, San Francisco).

ISO/IEC (1999): 13407 Human-Centred Design Processes for Interactive Systems, ISO/IEC 13407: 1999 (E), 1999

Leikas, J., Väätänen, A., Virtanen, T. & Suihkonen, R. (2005): Design rationale - Bridging the Gap between user requirements and implementation. HCI International 2005 Conference, HCII 2005. Las Vegas, 22 -27 July 2005. HCII. USA (2005).

Step 1: Define user requirements and context of use

Creating user friendly products starts from a thorough understanding of the users. Who are they? What are their goals? What problems do they need to solve? In what kind of environment are they living and working? What is their context of use?

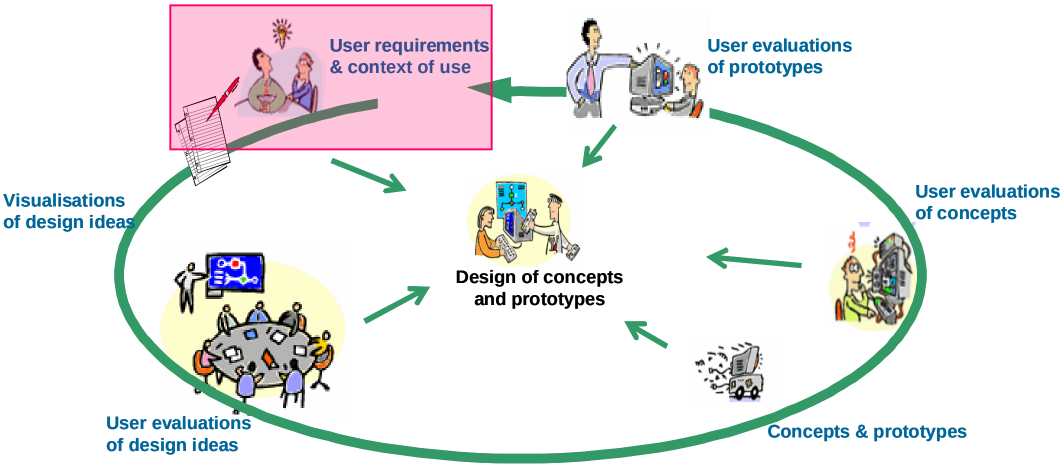

User requirements & context of use
User evaluations of prototypes
User evaluations of concepts
Concepts & prototypes
User evaluations of design ideas
Visualisations of design ideas
Design of concepts and prototypes

Figure 3.2. Step 1: Define user requirements

Defining the context of use is the starting point of an iterative user-centred design process. According to ISO standard 9241-11:1999 (Guidance on Usability), context of use consists of users, tasks, equipment and the physical and social environments in which a product is used. This information, together with a review of the current processes and a review of similar systems or products on the market (if available), is used to identify an initial statement of requirements. At the end of this stage, a list of initial design ideas or a design concept can be produced and evaluated by the intended users.

The description of the context of use specified during this phase of the design process should:

- specify the users, tasks and environments in sufficient details to support design;
- be confirmed by the users or their representatives;
- be adequately documented;
- be made available to the design team;
- be a working document, which will be updated throughout the design process.

In order to collect information about the context of use, it is first necessary to describe the goals and scope of the project. It is best to describe a project in terms of a ‘problem’ in need of a solution; a user-centred design process will then determine the exact solution. This approach has a better chance of achieving a successful outcome than one that begins with pre-conceived solutions.

The characteristics of the *intended users* that should be defined include knowledge and skills of the users, their experience, education, physical attributes, habits, preferences, and capabilities. If the users have different roles, these should also be described (e.g., purchaser, secretary, financial manager, inventory personnel).

The description of the *user tasks* should describe the tasks, the goals of the user, the frequency of the tasks and their importance to the users. If there are different user groups, the task descriptions should be connected to the appropriate user groups. At this phase of the design process one should describe user tasks, not functions of the system to be designed. However, the task descriptions can give initial ideas about which parts of the tasks could be allocated to the technology to be designed. The current processes for achieving user goals should also be reviewed, and possible problems documented. It is also useful in this stage to review existing systems and products that are aimed at meeting the same user/task goals, in order to identify the good features that could be incorporated into the new system or possible bad ones that should be avoided.

The *environment* description defines the other systems (hardware, software, networks) that affect the use of the planned system. This description also defines the physical environment (home, office, car), the legislative environment (laws and directives), and the social and cultural environment (organisational structure, work practises, target cultures, etc.)

During the iterative design process, the context of use can be refined. For instance, new user groups can be identified or the characteristics of the present user groups can be refined. Often in the evaluation stage the users will identify new tasks that they would like to perform with the system. These new features may again generate new user groups or changes to the planned physical or social environment.

It is not possible to understand what people want, need, can do and will do without talking to them yourself. Neither can one rely only on descriptive data. In the project team, it is essential that the designers have direct contact with the users. The usability experts in the design team facilitate the communication between users and designers, but cannot take care of the communication on behalf of the designers. Direct contact helps reduce the often-mentioned problem that the users and the designers do not speak the same language. You may not have any idea about what you would like to know about the users until you meet them personally in their own environment.

Commonly used methods and tools

The following methods present different ways of collecting information on the users and the context of use and defining initial user requirements.

Interviews

Interviewing is still the most widely used method of learning about the users and what they want. There are 3 types of interviews: unstructured interviews, semi-structured interviews and structured interviews. The type, detail and validity of data gathered vary with the type of interview and the experience of the interviewer. Interviews can be used in all phases of the design process: context of use analysis, definition of user requirements and evaluation at different phases.

Unstructured interviewing is characterised by an unconstrained attitude to the agenda and is a technique that can be conducted in practically any human endeavour.

Semi-structured interviewing is useful in situations where broad issues may be understood, but the range of respondents' reactions to these issues is not known or suspected to be incomplete.

Structured interviewing should only be carried out in situations where the respondents' range of replies is already well known and there is a need to measure the strength of each shade of opinion (Maguire 1998; Preece et al. 1994).

In each category, the interviews may be carried out individually (one-on-one interviews, see Step 3, case Nomadic Media) or in groups (e.g., in focus groups).

Observation

Observation is a method where an investigator views users as they work and takes notes on the activities that take place. Observation may be either direct, where the investigator is actually present during the task, or indirect, where the task is viewed by some other means such as video.

Observation is useful for studying currently executed tasks and processes in an organisation, and is commonly used for that purpose (Maguire 1998).

Contextual Inquiry

Contextual Inquiry is based on ethnography and the sociological research tradition where the researcher/observer visits the users' own environment. The observer not only observes the user but also asks questions about events that are not obvious as such. Work products like data sheets and notes can also be collected for later reference. The observation session can be videotaped for later reference.

This method can be used for task analysis and especially to gain fresh insights and to create a basis for design ideas (Beyer et al. 1998).

Literature study

The user requirements definition should analyse what has already been found out and what kind of user requirements can be identified on the basis of previous studies. Therefore, in most cases it is recommendable to carry out literature studies on user requirements as well as market surveys on products that relate to the concept or product at hand.

Brainstorming

The idea of brainstorming is to let people come together and inspire to generate new ideas by opening the mind to any idea that is suggested, allowing freedom for creativity. Brainstorming can be used for rapid generation of ideas or identification of problems in a specific domain, and is focused on the quantity of the responses.

It is typically used early in the development phase when little of the actual design is known, and there is a need for new ideas (Maguire 1998).

Evaluation of the usage of corresponding products

Sometimes a good source of information for user requirements is the evaluation of previous products. By studying how people deal with these products you can identify problems with current solutions, needs for new features and ideas for further development. The evaluation can be carried out, e.g., as usability testing in laboratory (see Step 5). The methods should be chosen according to the type of the product and/or available resources and time.

Card Sort

Card Sort is a common usability technique often used to discover users' mental model of an information space. A typical application of card sorting is to get ideas for menu structures by asking users to sort cards with the command names: commands that get sorted together often should probably be in the same menu. A limitation of the method is that users may not always have optimal models. Card sorting (or other similarity measures) is often used to assess the difference between the way novice and expert users understand a system.

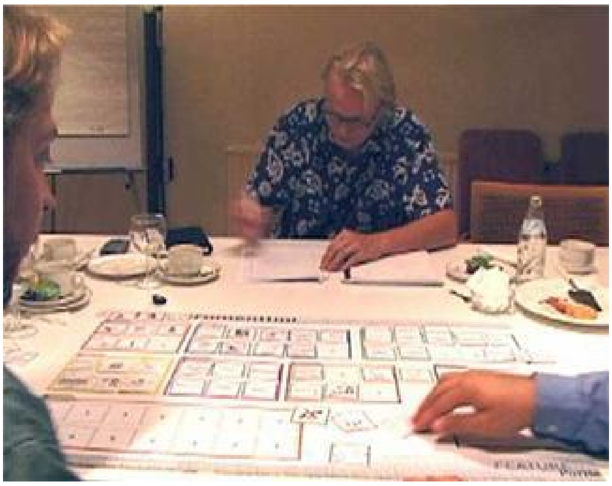


Figure 3.3. Card Sort

A typical Card Sort session proceeds in the following way (Nielsen et al. 1994):

- One note card is prepared for each concept in the application. The concepts are found beforehand, e.g., by performing a task analysis and interviewing users.
- Each note card is given to a user in a random order.

- The user sorts the cards on the basis of similarity. The user does not get any detailed instructions, but rather is just told to put the cards into piles that she/he thinks are similar.
- The user then groups the piles into larger groups. This could continue until a tree hierarchy is established, but can be stopped at a certain level.
- The user names the groups, e.g., using post-it notes. The verbal disagreement phenomenon (i.e. different people call the same thing by different names) is a great way to get suggestions for topic names.

Task Analysis

The aim of task analysis is to find answers to the following questions: Who are the users? What do they do? Where? Why? - Task analysis covers different sub-methods, e.g., focus groups, surveys, interviews, observations and diaries to collect information. Task analysis can be defined as the study of what a user is required to do, in terms of actions and/or cognitive processes, to achieve a certain goal.

Task analysis is best suited to situations where you have a well-defined user group with well-defined tasks. Usually such contexts are working places where different groups of employees have well-defined responsibilities.

Task analysis has classically been used in areas where the users' actions at the computer can be defined in terms of specific goals. For situations where the users' interest is likely to be unfocused, such as browsing or searching, task analysis may still be useful, but yields less tangible results (Kirwan et al. 1992).

Scenarios

A commonly used method in defining the initial user requirements and context of use is based on creating scenarios. See chapter 2 (Scenario based design).

QOC

See chapter 2 (Scenario based design).

Group discussions

See Step 3.

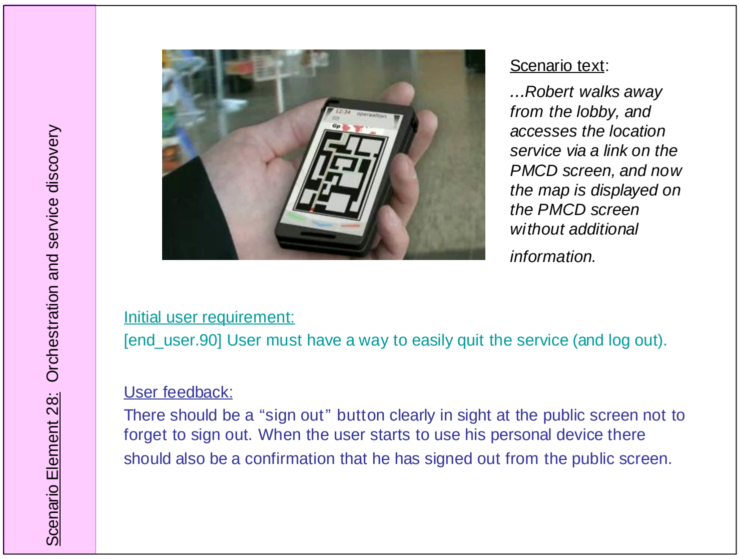
Use Case

See Step 2.

Case Nomadic Media: lessons learnt

Scenario based methods were used to gather user requirements in the Nomadic Media project. The approach on how to proceed with the scenarios has been presented in chapter 2 (Scenario based design).

An example of requirements specification: Figure 3.4 shows an example of a way to carry out the requirements specification discussed above. The process was documented with Power Point slides and used in a common workshop between the project partners in the design team.



Scenario Element 28: Orchestration and service discovery

Scenario text:

...Robert walks away from the lobby, and accesses the location service via a link on the PMCD screen, and now the map is displayed on the PMCD screen without additional information.

Initial user requirement:

[end_user.90] User must have a way to easily quit the service (and log out).

User feedback:

There should be a “sign out” button clearly in sight at the public screen not to forget to sign out. When the user starts to use his personal device there should also be a confirmation that he has signed out from the public screen.

Figure 3.4. Specifying user requirements.

The Nomadic Media scenario text is given on the top of the slide (Figure 3.4). First, the text was analysed and the scenario elements were recognised and documented as seen on the left of the slide. The technical members of the design team produced the initial functional and non-functional requirements from their point of view (given in Figure 3.4 with green text) of the specific scenario elements. These requirements were then cross-referenced with the user feedback (blue text) received from the scenario evaluation in focus groups carried out by the UCD team. The results were analysed in a common workshop. This kind of a requirements specification procedure enables to analyse and update the technical and quality requirements at the same time.

Suggested resources for additional reading:

Beyer, H. and K. Holtzblatt (1998): Contextual Design: Defining Customer-Centered Systems. San Francisco, Morgan Kaufmann Publishers.

Brassard, M. (1989): The Memory Jogger Plus: Featuring The Seven Management And Planning Tools Methuen MA., GOAL/QPC.

Carrol, J. M. (Ed.) (1995): Scenario-Based Design. London: John Wiley & Sons.

Drury, C.G. (1995): Methods for direct observation of performance, in J. Wilson & E.N. Corlett, Evaluation of human work: A practical ergonomics methodology, Taylor & Frances.

ISO/IEC (1998): 9241-11 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability. ISO/IEC 9241-11: 1998 (E).

Jones, J.C., Design Methods: Seeds of Human Futures, 1980 edition, John Wiley & Sons, (1980).

Kirwan, B and Ainsworth, L.K. (Ed.) (1992): A Guide to Task Analysis. London: Taylor and Francis.

Maguire, M. (Ed.) (1998): RESPECT User-Centred Requirements Handbook, version 3.2 June 1998.

Nielsen, J. (1993): Usability Engineering. Boston, MA: Academic Press.

Preece, J. et al. (1994): Human-Computer Interaction. Wokingham. Addison-Wesley.

Rosson, M.B. & Carroll, J.M. (2002): Usability engineering. Scenario-based development of human-computer interaction. San Francisco, CA: Morgan Kaufmann Publishers.

Rubin, J. (1994): Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. New York: Wiley & Sons.

Step 2: Visualise design ideas

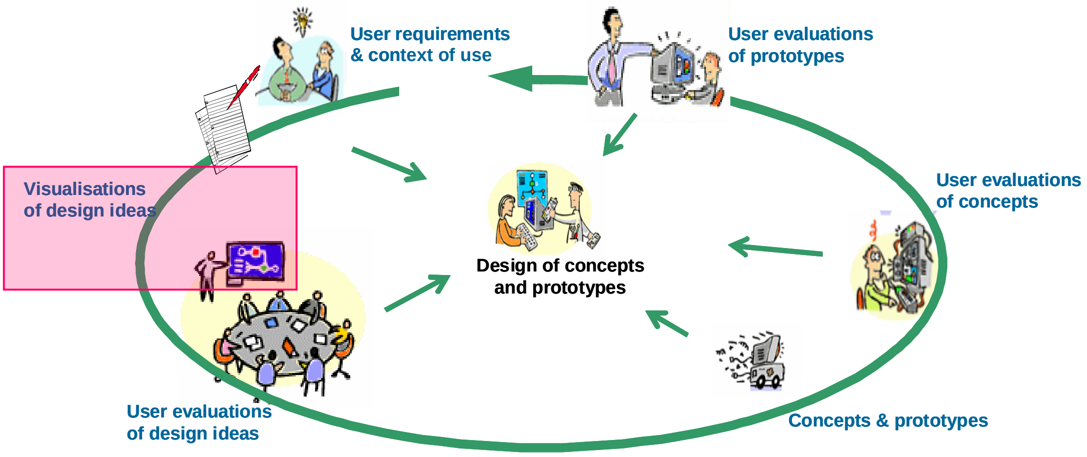

User requirements & context of use
User evaluations of prototypes
User evaluations of concepts
Visualisations of design ideas
Design of concepts and prototypes
User evaluations of design ideas
Concepts & prototypes

Figure 3.5. Step 2: Visualisation of design ideas

Illustrating potential design solutions in the early phases of the design allows the designers to communicate more effectively with the users and reduces the need for and costs of reworking later on. The potential design solutions should be based on the initial user requirements gathered, established state of the art as well as the experience and knowledge of the project participants. User interface style guides, knowledge of previous products and marketing knowledge can be used to support the initial design. Generic human factors and ergonomic design guidelines and standards should also be used.

The benefits of early, concrete illustrations include:

- Design decisions become more explicit and easier to comment on
- Designers can explore several design concepts before selecting one
- User feedback can be gathered early in the design
- Several iterations of a design can be evaluated and alternative designs compared
- Illustrations help in evaluating and completing the functional specification

Commonly used methods and tools

Visualisation of scenarios

In the beginning of the project, several scenarios may exist. Therefore, some rough analysis and first selection by prioritising and combining the set of scenarios for further refinement in the design team should be made. The criteria for this may consist of checking, whether each of the scenarios are complete, whether they have business potential and whether they meet the key objectives of the project. Also the scenarios should meet the aims and goals of the different partners. This refinement process leads to more sophisticated set of scenarios.

Why to visualise scenarios?

Scenarios are usually text-based descriptions including detailed information about users, contexts and devices. The amount of written text of scenarios increases with the level of detail causing the need to find alternative ways to present them.

It takes a lot of time to read through detailed descriptions on some device. On the other hand, the end-users should be able to understand the scenario in a relatively limited amount of time. Therefore, scenario content should be presented in an easily approachable and not much time consuming form.

Visualisation of a scenario produces a concrete representation showing how a scenario is to be played out. When you have a visual representation, you have a visual tool for communicating the essential aspects of the scenario to other people. A language barrier might prevent the message being delivered, but by visualising we avoid the pitfalls of written text.

Finally, the aim of visualising the scenarios is to modify the scenarios into such a form that they become more understandable to the end users. By visualising scenarios the possible conflicting views of the content of the scenarios can be avoided and misunderstandings revealed.

How to visualise scenarios?

There are many different ways to visualise scenarios, such as drawings, photos, storyboading, use cases, video, animation and acting. Some of them suffer from excessive complexity when prepared and modified and some turn to be time-consuming in use. The proper method should be selected on the basis of time and cost effectiveness and on special demands and features of the individual case (Carrol 1995; Androile 1991; Dumas et al. 1993; Maguire 1998).

An innovative technique for visualising scenarios was studied in the Nomadic Media project (see case Nomadic Media, match-stick man visualisation, MMV). The technique is based on the combination of storyboards, mock-ups, drawing and photos. The method is designed for people working with development and evaluation of scenarios and their contents. This technique is designed to be applied in a situation in which a scenario is to be communicated to different kinds of stakeholder groups in product development (e.g. users, other designers) in order to evaluate the scenario. These stakeholder groups do not necessarily have the knowledge and understanding of what technical terms mean, may not even speak the same language and may not have much time to spend evaluating the scenario. In such a case it is meaningful to provide an easily approachable form of representation, such as a matchstick man visualisation, to the stakeholders.

Before applying the matchstick-men technique, it is assumed, that the creator of the visualisation should have a basic idea of what he/she wishes to visualise. The scenario could be available on paper as a written text, or it could exist only as an abstract idea. In the latter case, it would be beneficial to be able to state the content of the future visualisation very briefly using one sentence. An example of a statement could be as follows; “At the airport the user buys music from an on-line music-shop using his new mobile communication device”.

Match-stick Man Visualisation technique (MMV)

The match-stick man visualisation (MMV) technique consists of five steps:

1. Identifying visualisation elements: Visualisation starts by analysing the content of the scenario, in order to identify all the necessary visualisation elements. MMV has five different types of visualisation elements:

- Matchstick-men characters that represent the users;
- Context-elements that represent the context of use;
- Artefact-elements that represent the systems, devices and other concrete artefacts;
- Interaction-elements that represent user-device-system interaction;
- Information-elements (think-, speech bubbles and written text) represent scenario content in written text.

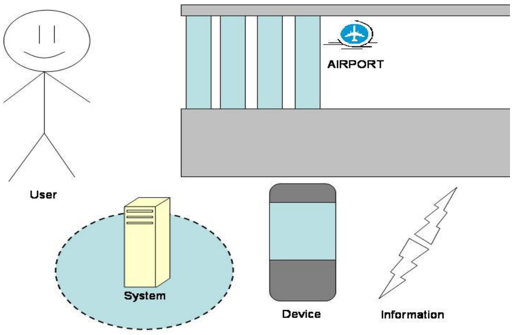


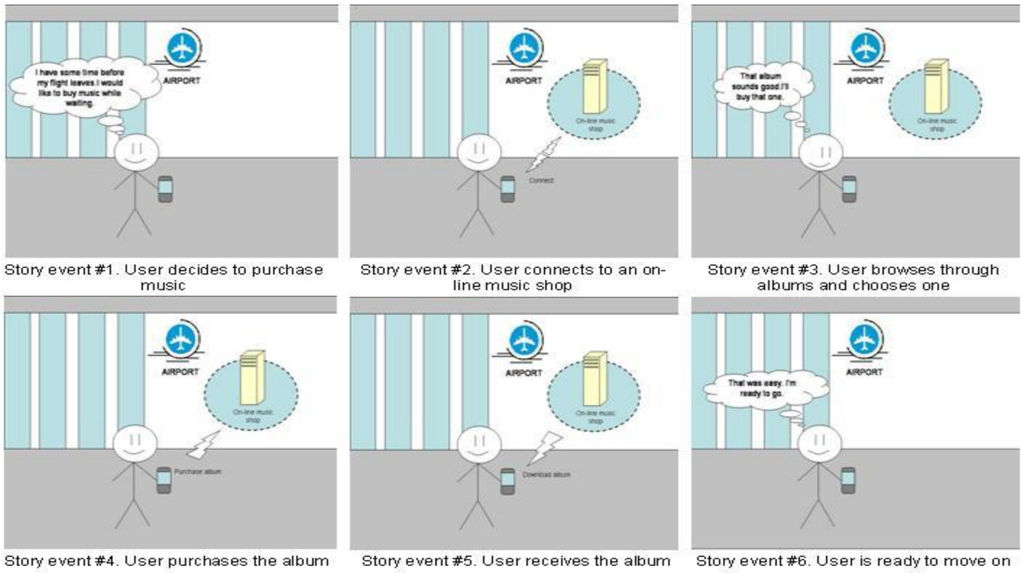
Figure 3.6. Visual elements in the visual element container.

The visual elements are shown in Figure 3.6, which is called the *visual element container*. An example of identifying the visualisation elements is demonstrated using the sentence: *"At the airport the user buys music from an on-line music-shop using his new mobile communication device"*. This sentence, when analysed, identifies at least four aspects that are relevant for creating a visualisation: the context (airport), the user (currently unknown user), the systems involved (on-line music shop and a mobile communication device) and the interaction between the system and the device. The sentence does not imply the use of information-elements such as think balloons, but they can be used to clarify the content of the scenario if necessary.

2. Identifying the story events: The designer should go through the scenario, make notes of the events that point to story events regarding the scenario as a whole. A story event is a concrete happening in the scenario, such as arriving at the context, operating the device, operating the system. Interaction flow between these elements can also be considered as story events for visualisation. For clarification, story events can be stated using text (see Figure 3.7). The number of identified story events affects the amount of visualisation work. At the beginning, one should try to identify the basic events and cover the whole scenario content.

3. Visualising story events: After identifying the story events, copy-paste the needed visualisation elements from the visual element container to the target slides. It is better to visualise one story-event / slide because the modifiability and maintainability decreases rapidly if many events are visualised in the same slide. The first slide in Figure 3.7 gives an example of how to visualise a single story event.

4. Combining visualised story events into a scenario visualisation: The slide content and the number of slides depend on how many essential story events the designer is able to identify. In addition, the detail of the visualisation affects the number of slides; a more detailed visualisation of the interaction means more slides. The layout depends on the applied media and the intended use. If you are presenting the visualisation digitally, you could have one story event / slide. Printable visualisation can have a cartoon-like layout to save space. Figure 3.7 illustrates the cartoon-like layout of the story events.


I have some time before my flight leaves.I would like to buy music while waiting.
AIRPORT
AIRPORT
On-line music shop
Connect
That album sounds good.I'll buy that one.
AIRPORT
On-line music shop
AIRPORT
On-line music shop
Purchase album
AIRPORT
On-line music shop
Download album
That was easy. I'm ready to go.
AIRPORT

Story event #1. User decides to purchase music

Story event #2. User connects to an on-line music shop

Story event #3. User browses through albums and chooses one

Story event #4. User purchases the album

Story event #5. User receives the album

Story event #6. User is ready to move on

Figure 3.7. Cartoon-like layout of the story events.

5. Evaluate the visualisation and iterate the steps if necessary: After the first iteration is finished, it is recommended that the designer evaluates the visualisation with other stakeholders. Asking other stakeholders to read and give their own interpretations of the visualisation is one way of testing the visualisation. If the stakeholders' interpretations of the scenario differ significantly from the designer's communicational intentions (the desired interpretation), it is a clear sign that further iterations are necessary.

Use cases

Use cases were originally defined as a part of object-oriented software engineering. They aim to describe all the different ways that the users can interact with the system. In use cases the environment of the system is defined by describing different users, who represent different user groups. The representatives of the user groups are called actors. An actor can also be another system that will be connected to the system to be designed. The actors use the system through a number of use cases. The use case model of a system is an external view of the system. The object model is an internal view of the same system.

Use cases form a model of the whole structure of the system. A complete object model of a system is seen through a set of object model views - one per each use case. Use cases simplify the traceability between different object models. Later on in the design process, use cases can be used in evaluations of concepts and prototypes to ensure that the planned functionality has been implemented.

Use cases should be updated throughout the iterative design process.

Storyboarding

Storyboards are sequences of images, which demonstrate the relationship between individual displays and actions within a system. A typical storyboard contains a number of images depicting features such as menus, dialogue boxes and windows. A sequence of these screen representations conveys further information on the structure, functionality and navigation options available within an intended system. The storyboard can be shown to colleagues in a design team and to potential users. This allows users and design team members to offer critical feedback about the composition and scope of the intended interface.

Storyboarding can be used early in the design cycle, in which case it supports the exploration of design possibilities and the early verification of user requirements. The method is of general relevance, especially to products in which a complex structure of information is being developed (Androile 1991).

Multimedia based visualisation

A scenario can also be visualised by means of a video or a multimedia presentation or a combination of these two. By using video or animation the flow of events can be nicely presented from the users´ point of view. If needed, also e.g. the device can be highlighted when designing the presentation. First a storyboard (including e.g. drawings) based on the original scenario may be made and then the final presentation is created. The team may consist of graphic designers, sound designers, animators, cameramen etc. since diversified know-how is needed for this kind of high quality presentation.

Case Nomadic Media: lessons learnt

Sketches: Illustrations of the Nomadic Media scenarios with simple drawings were carried out. Then users were interviewed using textual illustrated scenarios and the drawings served to show the context of use and to give rough impression of functional properties to be discussed. Though, these sketches were found useful, discussing functions while using functional demos could help to specify certain user requirements more precisely.

Multimedia based visualisation: A part of the Nomadic Media scenario was visualised with an animated multimedia presentation. Also a storyboard of the scenario was used. Experience of using the multimedia based visualisation (flash animation) of the scenario was twofold: To produce a multimedia presentation is relatively laborious. Also, producing such kind of presentation requires expertise of video and audio technologies. For example, failures on the sound track balance and volume level can irritate the users and draw attention away from the actual concept. Also, modifications to multimedia presentation are difficult to make afterwards. For these reasons careful preparations and proper understanding and analysis of the scenario to be visualised is necessary.

On the other hand, multimedia based visualisations helped the focus groups evaluating the scenarios to understand the scenario content. The multimedia presentation was found entertaining but due to rather fast flow of events it was considered difficult to follow at times. According to some elderly focus group participant the rapidly changing pictures of the flash animation were somewhat annoying.

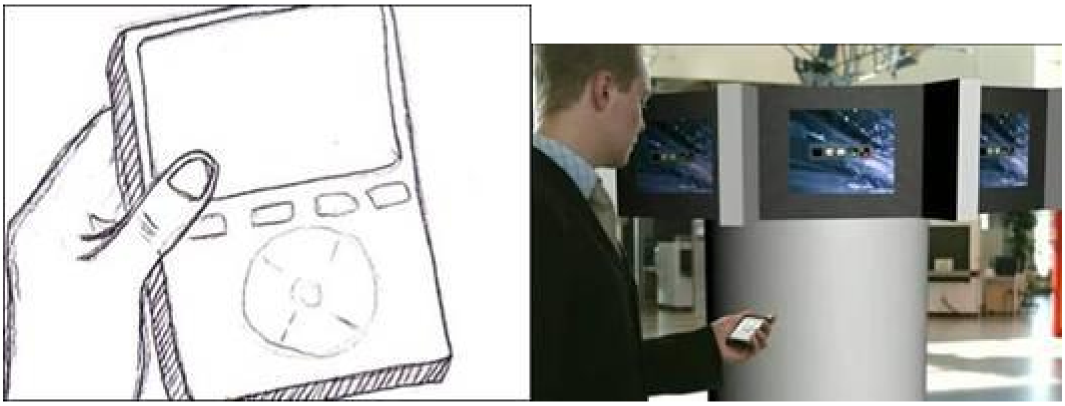


Figure 3.8. A picture of the storyboard (on the left) and of the multimedia (on the right)

MMV: The proposed matchstick-man visualisation technique supports rapid iterative prototyping and it can be used in the early phases of product development (Aikio et al. 2005; Aikio & Jounila 2005). It is intended to function as a visual aid for communication between different stakeholders (UCD designer/UCD designer, UCD designer/SW designer or UCD designer/user). This technique is designed to be applied in a situation in which a UCD-designer has to present a scenario to different stakeholder groups.

The MMV technique has several benefits: it can be implemented using a common tool (MS-PowerPoint), it does not require sophisticated graphical drawing skills, and it supports rapid low-fi prototyping of scenarios. Sequencing scenarios into story events and visualising them with a low-fi technique was received well. Using clear sequencing and low-fi technique supports easy creation, high modifiability and maintainability.

UCD-designers with some artistic skills are more likely to adopt the technique. Therefore, for the users of this technique some basic graphic design orientation is recommended.

Suggested resources for additional reading:

Androile, S. (1991): Storyboard Prototyping. Wellesley, MA: QED Information Sciences.

Aikio, K.-P. and I. Jounila (2005): Guide for visualizing scenarios. University of Oulu, Department of Computer Sciences (Master's graduate thesis: in Finnish)

Aikio, K.-P. – Jounila, I. – Jokela, T. (2005): Developing Techniques for Visualising Scenarios. HCII 2005 - 11th International Conference on Human-Computer Interaction. Las Vegas, USA, July 22nd -27th 2005.

Carrol, J. M. (Ed.) (1995): Scenario-Based Design. London: John Wiley & Sons

Dumas, J. S. and J. C. Redish (1993): A Practical Guide to Usability Testing. Norwood, Ablex Publishing Corporation.

Jacobson, I. (1995): The Use-Case Construct in Object-Oriented Software Engineering. In: J. Carroll (Ed.) Scenario-Based Design. Wiley & Sons.

Maguire, M. (Ed.) (1998): RESPECT User-Centred Requirements Handbook, version 3.2 June 1998.

Preece, J. et al. (1994): Human-Computer Interaction. Wokingham. Addison-Wesley.

Step 3: Carry out user evaluations of design ideas

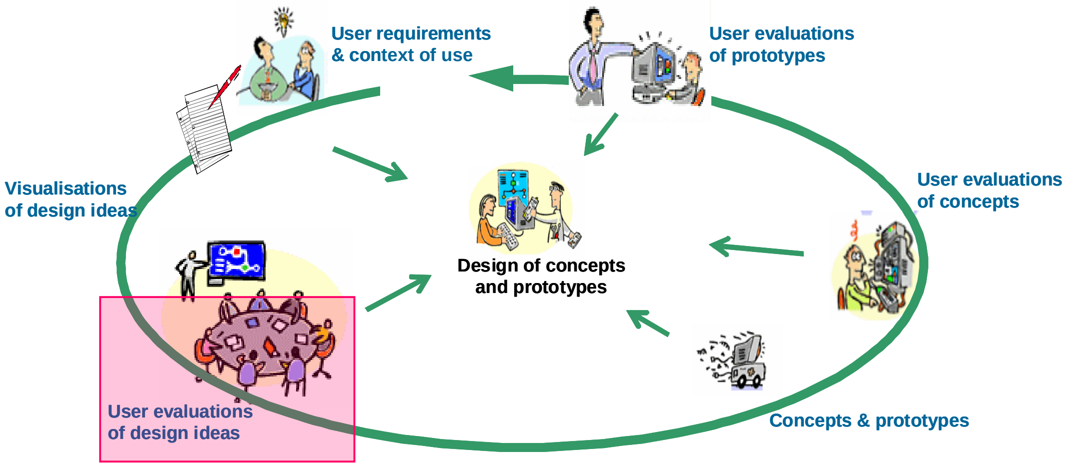

User requirements & context of use
User evaluations of prototypes
Visualisations of design ideas
User evaluations of concepts
Design of concepts and prototypes
User evaluations of design ideas
Concepts & prototypes

Figure 3.9. Step 3: User evaluations of design ideas

Once the context-of-use information has been collected, the design team can develop one or more ideas (or system concepts) for the new system. It is preferred to develop and compare several concepts. The most feasible concepts are then taken forward as part of the user requirements specification. The following techniques (besides scenario-based design) are useful for this type of early design concept generation.

Commonly used methods and tools

Focus groups

A focus group brings together a cross-section of interested parties in an informal discussion group format. The group should include users or their representatives. A facilitator elicits views on relevant topics. Meetings can be recorded for later analysis. Focus groups are useful early in the user requirements specification. They help to identify issues which may need to be tackled, and provide a multi-faceted perspective on them (Maguire 1998).

Group Discussions / Future Workshops

Group discussions help on summarising the ideas and information held by individual members. The group should include both designers and users or their representatives. New ideas, design options, costs and benefits, screen layouts, etc. are discussed within the design process. By discussion, a collective view is established from the individual parts.

Figure 3.10. Group Discussion

The 'future workshops' concept is designed specifically to allow all members to participate in the discussion. A future workshop goes through three phases:

- Critique: the participants voice current problems in a constructive manner;
- Fantasy: the participants generate visions of the ideal future scenario;
- Implementation: the ‘visions' are evaluated and a plan for future action is formulated.

Brainstorming

See Step 1 in this chapter.

Interviews

See Step 1 in this chapter.

Case Nomadic Media: lessons learnt

Focus groups: User evaluations and analysis of specific parts of the Nomadic Media scenario were aligned in several focus group sessions.

Steered focus group sessions seemed to be a reasonable way in involving different user groups into the development process and to gain their input for the scenario refinement and for revising the initial scenario.

It has been argued that focus groups do not necessarily provide valid results because it is difficult for people to reliably tell how they would behave in a hypothetical situation or would actually use/like a particular service. Our experience showed that focus groups at least succeeded in highlighting issues related to 'what not to build'. Related to certain aspects of the scenarios the focus groups highlighted reasons why users would surely not to use the device/service in question. Therefore, the focus groups succeeded in revealing important issues related to the contexts of use of the prospective devices/services.

The participants represented in focus groups must be selected with care from presumed end-users of the system or product or service to be developed (Ikonen et al. 2005). The participants may also have some other kinds of application related expertise to comment on the scenario.

For analysis purposes recording the sessions (video, audio) are recommended.

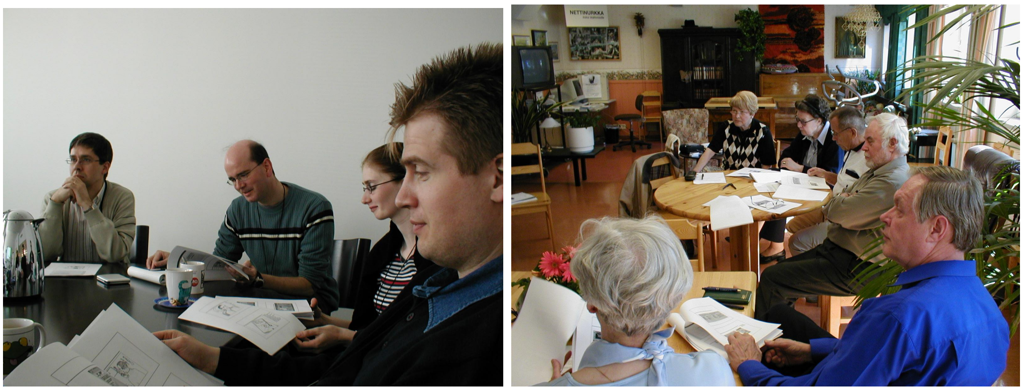


Figure 3.11. Two focus group sessions with people of different age and background

A set of general questions applicable to be used in focus groups:

v How realistic is the scenario? Which parts of the scenario can be considered as science fiction, which are found realistic (credibility)?

v What is the user experience of the services and products presented in the scenario (e.g. usefulness, ease of use, desirability, and enjoyability)?

v What kind of ethical issues are related to the scenario?

v If this kind of service/system or a product would be available, would users take it into use?

v How to improve the scenario(s)?

v How much would users be willing to pay of the service/system or a product?

v What are the new innovations and suggestions for the other user groups and context(s)?

v What are the user tasks related to the different parts of the scenario?

v What are the usability requirements for the user tasks?

Interviews (One on One interviews): All participants were asked the same questions in a structured interview (although not always in the same sequence).

First, each participant was asked to describe the current context of use in relation to the Nomadic Media scenario to be studied (routines, habits, what he likes/don't like, what they want to keep, what they want to change, why they did not change it yet, etc.).

In next phase the participants were taken step by step through the scenario of the subject. The participants had to respond immediately to each newly introduced function and the related usage activities. In the real life, a service system unfolds itself to the user in steps and one does not perceive the entire service right away as a whole. By using the system or device one can build the general view of it gradually. Therefore, the user responses of "top level" functions or properties were studied at first. Then the participants were taken to discuss specific functions in more detail. The challenge is to keep the users involved in case they do not see the advantage of top-level concept.

At the end, the participants were asked to elaborate on responses between their current way of doing things and the scenario (in case they seemed to be contradicting). For example, the expectations about the usefulness of certain functions based on the scenario may be different when compared to the habits of the users. This kind of interviewing seems to produce much useful information and the approach is useful in cases, where people have to imagine services and systems which are new to them.

Suggested resources for additional reading:

Caplan, S (1990): Using Focus Group methodology for ergonomic design. Ergonomics, 33.5, 527-537.

Ikonen, V., Leikas, J., Strömberg, H. (2005): Nomadic Elderly: design requirements for technologically enhanced traveling experience. 5th International Conference of the International Society for Gerontechnology, Nagoya, Japan, May 24-27, 2005.

Maguire, M. (Ed.) (1998): RESPECT User-Centred Requirements Handbook, version 3.2 June 1998.

Poulson, D., Ashby, M. and Richardson, S. (Eds) (1996): Userfit: A practical handbook on user-centred design for Assistive Technology. Brussels-Luxembourg: ECSC-EC-EAEC.

Step 4: Create concepts and prototypes

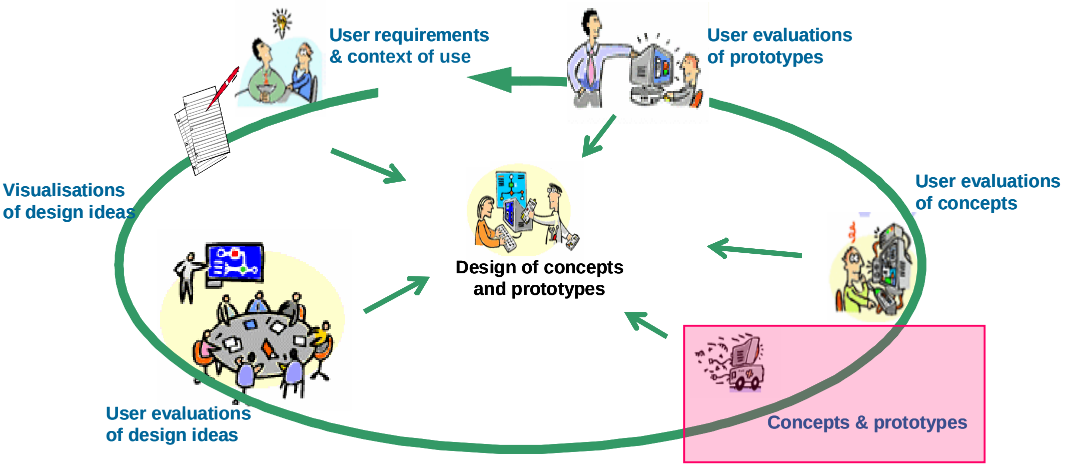

User requirements & context of use
User evaluations of prototypes
Visualisations of design ideas
User evaluations of concepts
Design of concepts and prototypes
User evaluations of design ideas
Concepts & prototypes

Figure 3.12. Step 4: Creation of concepts and prototypes

After received an understanding concerning the users, context of use and the user requirements the next step is to create the concepts and demonstrations. In the following, several alternatives commonly used for concept creation will be given.

A prototype is a representation of all or a part of a product or system that, although limited in some way, can be used for evaluation (ISO 13407:1999). In the user-centred design approach, prototypes are not simply demonstrations to show users a preview of the design, but they are developed to collect user feedback that is then used to drive the design process. A prototype can be as simple as a pencil and paper sketch or as complex as a computer-based simulation.

Simple prototypes are valuable in the early stages of design to explore alternative solutions. They should be kept simple enough and not too much design effort should be devoted to them. In this way it is easier for both the users and the designers to assess them freely. If the prototype looks too "finished", the users often tend to be too polite to criticise it, and the developers have invested too much effort to discard the prototype readily. In software design, prototyping typically starts with screen-view mock-ups and continues with simple prototypes which support only a subset of user tasks. Successive prototypes illustrate the implementation of support for various user tasks. The prototypes are updated on the basis of user feedback and are combined to cover more functionality.

The first prototypes can be evaluated in design walkthroughs. Functional prototypes can be evaluated in user evaluations under laboratory conditions. Later on, the trials can be done in more realistic contexts.

Commonly used methods and tools

Paper prototyping

This method features a paper-based simulation of an interface or system. Paper prototypes provide a valuable and cost-effective means of evaluating and iterating design options before deciding on one implementation. Interface elements such as menus, windows, dialogues and icons can be sketched on paper or created in advance using cards, pens, etc. The result is sometimes referred to as a low-fidelity prototype.

After the paper prototype has been prepared, a member of the design team sits in front of a user and plays the part of the computer by moving interface elements around in response to the user's actions. The user makes selections and activates interface elements, e.g., by using his/her finger as a pointing device and speaking out 'typed' input. Another person provides task instructions and encourages the user to express his/her thoughts and impressions. An observer may make notes and/or the session can be recorded on video for later analysis. The method is most suitable in contexts where it is easy to simulate system behaviour or when the evaluation of detailed screen elements is not required. Paper prototyping is appropriate for the early stages of the design cycle, where changes can be readily made before there is a commitment to one implementation.

Wizard of Oz Prototyping

This approach involves a user interacting with a computer system, which is actually operated by a hidden developer - referred to as the "wizard". The wizard processes input from a user and simulates system output. During this process the user is led to believe that she/he is interacting directly with the system. This form of prototyping is beneficial early on in the design cycle and provides a means of studying a user's expectations and requirements. The approach is particularly well suited to exploring design possibilities in systems which are demanding to implement. The approach is highly applicable to "intelligent interfaces" which feature agents, advisors and/or natural language processing (Maudsley et al. 1993).

Video Prototyping

This method allows designers to create a video-based simulation of interface functionality by using simple materials and equipment. Interface elements are created using paper, pens, acetates, etc. For example, a start state for the interface is recorded using a standard video camera. Stopping and starting the camera as interface elements are moved, taken away and added may then simulate the movements of the mouse pointer over menus. Users do not interact with the prototype directly although they can view and comment on the completed video-based simulation.

Video prototyping is particularly well suited to simulating interface functionality. A limitation of the method is that it must be possible to easily simulate the interface elements with basic materials. The method is relevant in the early stages of the design cycle to demonstrate design options and concepts (Vertelney 1989).

Rapid Prototyping

This method is concerned with developing different proposed concepts by evaluating software or hardware prototypes. The development of a simulation or prototype of the future system can be very helpful. It allows users to get an idea of the look and feel of the system and provide feedback on it. Thus it can be used to clarify user requirements options. Rapid prototyping is described as a computer-based method which aims to reduce the iterative development cycle. Interactive, quickly replaceable prototypes are developed in line with design feedback.

This feedback may be derived from users or colleagues as they carry out set tasks with the prototype. The prototypes exhibit a higher resemblance to the end product than in methods such as paper prototyping. The method requires more technical resources than prototyping methods which rely on paper materials. An additional cost is the human expertise required to master the development tools, along with the time necessary to implement a software prototype.

Proof-of-concept demonstrators

The term proof-of-concept is used often as a synonym for prototypes. A way of doing rapid prototyping is to prepare proof-of-concepts where the aim is to test the design idea in the early phase of the development with existing robust technology. In software engineering it may be a piece of code, while in product development early prototypes are proof-of-concepts that can be used to get a view of the design and interaction with the product to be developed. Proof-of-concepts are useful tools in the usability tests.

Case Nomadic Media: lessons learnt

Rapid prototyping: In the technical work packages of the Nomadic Media project a system enabling rapid prototyping was developed. The system allows to design and to implement adaptive, multi-channel and context sensitive web applications. This application platform makes use of formal task modelling which makes possible to design applications, e.g., to smart phones and desktop computers at the same time.

Prototypes of a service application based on the Nomadic Media scenario were created and adapted for laptop computer, iPAQ and mobile phone with this application for usability testing purposes.

In this case, rapid prototyping turned to be an efficient and useful way in getting user feedback in usability tests with a set of devices (see: Step 5).

Proof-of-concept demos (POC): During the development and implementation of the proof-of-concept, many trade-off decisions had to be made regarding technical and usability issues. Because one of the objectives of the project was also to develop and implement certain technologies, the focus in POC demos was put on such user issues where, e.g., usability had a clear impact on the technical development. Though the proof-of-concepts were not user friendly as a whole, in this phase it was more important that certain technical implementations could be tested with users.

Suggested resources for additional reading:

ISO/IEC (1999): 13407 Human-Centred Design Processes for Interactive Systems, ISO/IEC 13407: 1999 (E), 1999.

Maguire M. (1996): Prototyping and evaluation guide. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK. October.

Maudsley, Greenberg and Mander (1993): Prototyping an intelligent agent through Wizard of Oz. Interchi '93 Conference Proceedings, pp 277-284.

Vertelney, L. (1989): Using video to prototype user interfaces. SIGCHI Bulletin, 21, 2, 57-61.

Step 5: Evaluate concepts and prototypes

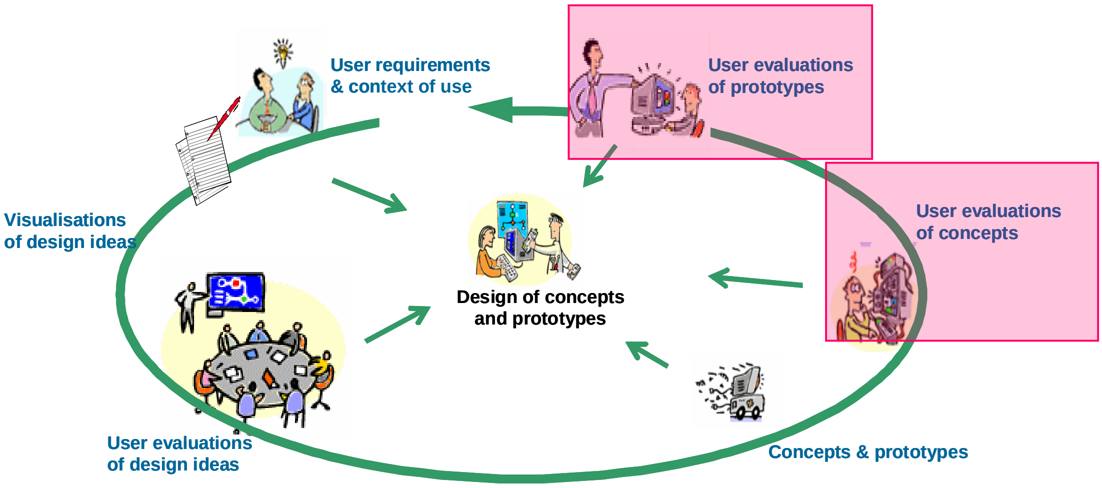

User requirements & context of use
User evaluations of prototypes
User evaluations of concepts
Visualisations of design ideas
Design of concepts and prototypes
User evaluations of design ideas
Concepts & prototypes

Figure 3.13. Step 5: User evaluations of concepts and prototypes

Evaluation of design solutions is an essential step of user-centred design and should be carried out at all stages in the system life cycle. Iteration of design solutions is an important principle of user-centred design. Therefore, evaluation of the design solutions should be started early in the development life cycle in order to iteratively improve the design. Evaluation should also be carried out to validate the user and organizational objectives have been met, i.e. when deciding when the system is 'good for the user'. (ISO/IEC 1999)

Altogether, there are two basic types of evaluation with different purposes (ISO/IEC 1999):

- Formative evaluation aims to provide feedback which can be used to improve the design and to generate new ideas;
- Summative evaluation aims to assess whether user and organisational usability objectives have been achieved.

Evaluations can be carried out for both products and prototypes (low- or high-fidelity prototypes), as well as for user task designs.

Evaluation should be started at as early stage as possible in order that the evaluation results could affect the design, and the changes are relatively inexpensive. In an early phase the emphasis should be on obtaining feedback, which can be used to improve the design and to generate new ideas. (ISO/IEC 1999; Rubin 1994)

During the early phases the goal is to evaluate the preliminary design concepts – the evaluation is related to the high-level aspects of the design (e.g. only main screens and navigation paths of the user interface are defined). Later the evaluation focuses on lower-level operations and aspects of the user interface, which is designed in more detail. (ISO/IEC 1999; Rubin 1994).

Comparative evaluations should be carried out for producing multiple design solutions and selecting the design among several alternatives (e.g. when comparing different user-interface styles or measuring the effectiveness of different elements of user-interface). (ISO/IEC 1999: Rubin 1994).

When a finalised prototype is available one should evaluate whether user and organizational objectives have been achieved. Final evaluation should be carried out to validate that the system meets the specified requirements. (ISO/IEC 1999: Rubin 1994).

Finally, evaluation of competitors or own products already in use should be carried out in order to derive feedback and requirements for next releases or similar types of products. (ISO/IEC 1999: Rubin 1994).

Long-term field trials can also be carried out to validate that the product responds to the requirements of the users, their tasks and the environment (field validation), to monitor the long-term use of the product or system, to measure user acceptance, to identify real contexts of use and usage cultures and to identify the key features of the system. (ISO/IEC 1999; Rubin 1994)

Commonly used methods and tools

Different methods serve different purposes, reveal different kinds of problems and require different kinds of expertise and resources. The methods complement each other, due to which it is suggested to use a combination of different usability evaluation methods. (Karat 1997; Mack et al. 1994; Karat 1994; Dumas et al. 1993)

Evaluation methods have been classified in many different ways. The most common way is to divide them into:

- empirical user testing methods and
- usability inspection methods.

They are separated due to the differences in the user involvement in the process. Empirical user testing includes representative users as participants. Usability inspection methods include methods that do not necessitate user involvement, but rely on inspectors' expertise and experience, and on guidelines, standards or heuristics. (Riihiaho 2000; Wixon et al. 1997)

Empirical user testing

It is essential to have real users evaluate the system. Empirical user testing is

- the best way to find out serious usability problems that are related to the user tasks and experiences;
- successful in identifying recurring usability problems and altogether the largest number of usability problems.

However, user testing with real users involved may be time consuming and expensive. (Dumas and Redish 1993; Karat 1994; Desurvire 1994; Karat 1997)

User testing can be used both for finding out usability problems to inform design (formative evaluation) and for measuring the system against the requirements (summative evaluation). It is more common to carry out tests to identify qualitative usability problems than to get quantitative measures (Wixon et al. 1997). Summative evaluation means setting usability requirements and measuring parameters to analyse the requirements. The measurable parameters can be objective (e.g. performance measures) or subjective (e.g. user preference measures). (Karat 1997)

Empirical user testing can be divided into methods in which testing is carried out in the field (natural setting) or in usability laboratory. Empirical user testing methods include, e.g., usability testing in the laboratory, field trials and usability walkthroughs. (Karat 1997)

Two empirical usability methods, i.e., *laboratory usability testing* and *field trials* will be presented in the following paragraphs.

Laboratory usability testing

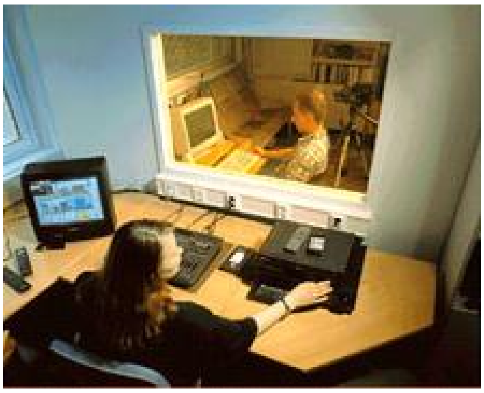


Figure 3.14. Usability testing in a laboratory

The most common and fundamental empirical user testing method is usability testing in which one or two representative users at a time do given tasks under observation in the usability laboratory. Users should be real users carrying out a set of real tasks. The sample match of users and tasks is important. Usually the users are asked to think aloud during the tests so that the observers can find out what the users are trying to do and why they make certain decisions. (Mayhew 1999; Karat 1997; Dumas et al. 1993; Nielsen 1993; Rubin 1994)

Primary goal is to find out usability problems to improve the design. Usually, there are more specific goals for each test (comparison to competitors, efficiency testing, etc.). Altogether, usability testing can be used both for formative evaluation and summative evaluation. However, it is more common to carry out usability tests to identify qualitative usability problems than to get quantitative measures. (Mayhew 1999; Karat 1997; Dumas et al. 1993; Rubin 1994; Wixon et al. 1997).

The usability testing process has three main phases: design and preparation of the test; conducting the tests and analysing and reporting the results. In the design and preparation phase a test plan is produced, the goals of the tests are defined, the user group for the tests is defined, representative users recruited, task scenarios and individual tasks to test defined, test material produced, test team prepared and a pilot test carried out. Typically all the tests are video taped. In the last phase the material gathered in the tests is analysed and usability problems identified and classified. Usually the scope and severity of the problems is used as classification criteria. A report is produced including the identified usability problems and recommendations for changes to fix them. Also a highlight tape (video tape) illustrating the most important findings (the most global or severe problems) from the tests should be produced. For the communication purposes, it is also recommended that the developers participate in the usability tests as observers. (Mayhew 1999; Dumas et al. 1993; Rubin 1994)

Field Trials

End-user involvement throughout the life cycle of the project enables evaluation of the usability of the implemented user applications in a real usage environment. The field trial focuses on evaluating the product in general as well as the specific features and functionality offered. In the trial, measurable criteria should be set to access how successfully these attributes and goals have been met.

The field trial can be either short-term or long-term. A short-term trial resembles laboratory usability testing, because it usually takes place within one session, but it is carried out "in the field", i.e. in the actual context the service is meant to be used in. The expert observes the use all the time and can ask appropriate questions during use.

Figure 3.15. Usability testing in real usage contexts

In a long-term field trial the user uses the device/service from several days to months as (s)he would use it in real life. In a long-term field trial the test users can freely use the prototype as a part of their normal life. This means that the actual use can only be observed during interviews. A long-term trial may also include interviews with observation that resemble the short-term trial. Usage information can also be gathered on log files for later analysis.

Data collection can take place in the form of automatic log files, observation, interviews, questionnaires, diaries, etc. Interviews and questionnaires can be used for gathering data about users’ experiences and preferences related to the system. Through these methods one can also study how users use the system and what features they like or dislike. Use data (e.g. related to errors, task completion times, general logging data) can be gathered through computer automatically collecting statistics of actual use (Nielsen 1993; Karat 1997). Contextual inquiry (Beyer et al. 1998) combines observation with interviews; the interviewer observes users at work and questions about the work.

The evaluation studies the usability of the product in the real context of use. Field trials require considerable amounts of resources and time. Thus it is not reasonable to test in the field things that you can test in a laboratory test. Before the field trial, the main usability problems should have been identified and fixed in a user evaluation under laboratory conditions. The field trial prototype should be technically reliable so that technical problems will not prevent use.

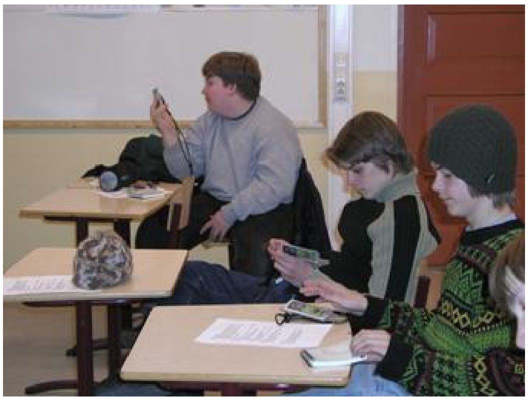


Figure 3.16 Evaluation of simple prototypes

The evaluation period can vary from a few days to several months, depending on

v the aim of the evaluation (field validation … long-term monitoring);

v the complexity of the prototype;

v available resources for the evaluation;

v the characteristics of the users (e.g. expert – novice);

v expected frequency of use.

Usability Inspection Methods

Usability inspection methods are non-empirical methods for evaluating user-interfaces. Usually they are not as labour intensive and expensive as empirical user testing methods, due to which they are more readily applicable - early and quickly - in the development. However, they do not provide as valid results of the usability of the system than the empirical user testing methods, and should not be seen as replacing empirical user testing. However, as an addition to cost effectiveness, inspection methods have also shown different problems than user testing methods, which implies that user testing and usability inspection methods should be used in combination. (Mack et al. 1994; Nielsen 1993; Desurvire 1994)

Usually usability inspection methods do not necessitate user involvement. They rely on the experiences and knowledge of the evaluators who inspect or examine user-interface using rules, guidelines or heuristics as a basis. (Mack et al. 1994; Virzi 1997).

Two usability inspection methods, i.e., *heuristic evaluation* and *cognitive walkthrough* will be introduced in the following paragraphs.

Heuristic Evaluation

The most common usability inspection method is heuristic evaluation (Nielsen 1993; Nielsen 1994) in which several evaluators check whether the system conforms to the given usability principles called heuristics. The evaluators may have different amounts of experience in usability evaluation. Generally, the more the evaluators have experience, the less evaluators are needed.

Heuristic evaluation proceeds as follows. First the evaluators carry out the evaluation individually. Each evaluator goes through the user-interface for several times and first focuses on the general flow, and afterwards inspects various dialogue elements and compares them with the list of heuristics. Afterwards the results are combined, and the severity of the usability problems estimated. Three factors affect the severity of the usability problems: frequency, impact and persistence of the problem. A result of heuristic evaluation is a list of usability problems with reference to the usability principles that were violated in each case. Fixes to the problems should also be recommended. (Nielsen 1993; Nielsen 1994.

Below is an example list of usability heuristics that should be followed by a user-interface designer (Nielsen 1993):

- Simple and natural dialogue
- Speak the users language
- Minimize the users memory load
- Consistency
- Feedback
- Clearly market exits
- Shortcuts
- Good errors messages
- Prevent errors
- Help and documentation

Heuristic evaluation has been identified as a very easy to learn, quick and cost beneficial method. In addition, heuristic evaluation has succeeded in identifying the most of minor, local problems, whereas usability testing usually succeeds in identifying the most major, global problems. (Dumas et al. 1993; Karat 1994; Nielsen 1993).

Cognitive Walkthrough

A walkthrough is a process of going step by step through a system design getting reactions from the users. Users, system developers and usability experts together examine and discuss quality of use issues associated with the dialogue steps in a task scenario. Normally, one or two members of the design team will guide the walkthrough, while one or more users will comment as the walkthrough proceeds. The walkthrough method obtains reactions to design in an informal manner. Being flexible it obtains reactions, allowing the users' discussion to range over issues not originally considered. Walkthrough requires some form of a prototype to show to the users and for them to react to.

Cognitive walkthrough is also a well-known usability inspection method that evaluates the ease of learning to use the system. The method has a theoretical basis related to learning by exploring, and the analysis focuses on users’ mental processes instead of the user-interface. Cognitive evaluation can be carried out in an early phase; the user-interface can be very roughly specified. (Wharton et al. 1994; Riihiaho 2000)

Either an individual or a group can carry out cognitive evaluation. The reviewers evaluate a user-interface related to one or more specific user tasks. The analysts focus on detailed actions in a certain task. There is a set of predefined questions the analysts ask related to each action of the task. The analysts try to discover design errors that would interfere with learning by exploring the system. (Wharton et al. 1994; Riihiaho 2000)

However, cognitive walkthrough has a reputation of being a labour intensive, tedious method - detailed, focused analysis of each action of only one task takes quite much time. (Wharton et al. 1994; Riihiaho 2000)

Case Nomadic Media: lessons learnt

Laboratory usability testing : Laboratory usability testing of multi-device user-interfaces generated through UI description and modelling concept (enabling rapid prototyping; see Step 4) developed in the Nomadic Media project has been carried out to evaluate the usability and adaptability of a service based on a Nomadic Media scenario. The service can be used with a variety of devices that differ in their size, bandwidth, screen size and keyboard.

Methods used were laboratory usability testing, SUS (System Usability Scale) questionnaire and interviews (both before and after the tests). Our experiences indicate that in this type of tests (evaluating multi-device user interfaces) the evaluation results may be affected by the usability of the device in question as well as by the test order during the evaluation (i.e., whether the user starts with mobile phone, laptop or iPAQ).

Laboratory usability testing was applied also to evaluate multiplayer mobile gaming with and without a public display. The experiences show that it may be difficult to create and achieve a real communal atmosphere in a laboratory as the application is designed to be used in a public context with many people around. (Strömberg et al., 2005)

Heuristic evaluation: Heuristic evaluation has been carried out by using the Nielsen’s heuristics on a context sensitive, adaptive and personalised Nomadic Media service.

The experience on the method showed that any kind of usability evaluation necessitates definition of basic functionality and some sort of user-interface design (e.g. main screens and navigation paths). Even though heuristic evaluation is suggested for early phases of design, it is not meaningful unless user-interface design has been properly carried out. This evaluation highlighted the difficulties in communication between usability professionals and developers. These two groups might have totally different understanding of usability and divergent motives for usability evaluation. In a situation in which the service to be evaluated is in a very early phase and developers are involved in architecture development, not in designing user-interfaces or even defining the basic functionality of the prospective service, any kind of usability feedback is difficult to provide.

Recommendations:

- Empirical user testing should always be carried out, since it is capable of finding out problems that the inspection methods miss. Usability testing is the best way to find out serious usability problems that are related to the user tasks and experiences and it has been successful in identifying recurring usability problems and altogether the largest number of usability problems.

- The costs associated with usability testing are not necessarily high; usability testing using a paper prototype or cardboard mock-ups can be carried out with moderate resources. (Dumas et al. 1993; Karat 1997; Desuvire 1994; Karat 1994)

- If the development organization is reluctant for taking usability into account, it is suggested that methods that allow the developers to participate in the evaluation sessions should be selected. (Karat 1997; Riihiaho 2000; Karat 1994)

- However, also usability inspection methods should be used as a supplement to empirical user testing. Generally usability inspection methods are more quick and easy to apply. Furthermore, they are able of covering a wider range of the user-interface than the selected tasks users carry out in the test sessions. (Dumas et al. 1993; Karat 1997; Desuvire 1994; Karat 1994; Riihiaho 2000)

- Shifting focus from traditional desktop computing to ubiquitous computing applications (that is available everywhere and all the time) means that a very heterogeneous group of users' everyday practises needs to be understood and supported (Abowd et al. 2002). This new way of using technology brings also challenges to usability evaluation. It is nowadays widely recognised that intelligent compound systems where many users and devices communicate simultaneously clearly require a new approach to system design and evaluation (Scholtz et al. 2002).

- Functionally complete prototypes are needed in the user evaluations to enable the end users to concentrate on functionalities and features offered instead of struggling possible technical problems. The end users' experience as well as concentration is negatively affected if the evaluation sessions are constantly interrupted due to technical problems. The interruptions disturb also the atmosphere of the user evaluation.

Suggested resources for additional reading:

Abowd, G., Mynatt, E, and Rodden, T. (2002). The Human Experience, Pervasive Computing, Vol. 1(1), 48-57.

Beyer, H. and Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann.

Desurvire, H. W. (1994). Faster, Cheaper! Are Usability Inspection Methods as Effective as Empirical Testing? Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Dumas, J. S. and J. C. Redish (1993). A Practical Guide to Usability Testing. Norwood, Ablex Publishing Corporation.

ISO/IEC (1999). 13407 Human-Centred Design Processes for Interactive Systems, ISO/IEC 13407: 1999 (E), 1999.

Jefferies, R., Miller, J.R., Warton, C. & Uyeda, K.M. 1991, User interface evaluation in the real world: a comparison of four techniques. Reaching through technology - CHI'91. Proceedings of the conference on human factors in computing systems, 27 Apr - 2 May 1991, New Orleans, Louisiana pp 119 - 124.

Karat, C.-M. (1994). A Comparison of User Interface Evaluation Methods. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Karat, J. (1997). User-Centered Software Evaluation Methodologies. Handbook of Human-Computer Interaction. M. G. Helander, T. K. Landauer and P. V. Prabhu. Amsterdam, Elsevier Science.

Lindgaard, G. (1994). Usability testing and system evaluation, A guide for designing useful computer systems. Chapman & Hall, UK. Paperback ISBN 0-412-46100-5.

Mack, R. L. and J. Nielsen (1994). Executive summary. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Mayhew, D. J. (1999). The Usability Engineering Lifecycle. San Fancisco, Morgan Kaufman.

Nielsen, J., and Molich, R. (1990). Heuristic evaluation of user interfaces, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1-5 April), 249-256.

Nielsen, J. (1993). Usability Engineering. San Diego, Academic Press, Inc.

Nielsen, J. (1994). Heuristic Evaluation. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Riihiaho, S. (2000). Experiences with Usability Evaluation Methods. Computer Science and Engineering, Helsinki University of Technology: 113

Rubin, J. (1994). Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. New York: Wiley & Sons.

Scholtz, J., Arnstein, L., Kim, M., Kindberg, T. & Consolvo, S. "User-centered Evaluations of Ubicomp Applications," Intel Research, IRS-TR-02-006, May. 1, 2002.

Strömberg, H., Leikas, J., Suomela, R., Ikonen, V. and Heinilä, J. (2005): Multiplayer gaming with mobile phones - enhancing user experience with a public display. Intetain (Intelligent technologies for interactive entertainment); Madonna di Cambiglio 30.11.- 2.12.2005, Italy. (http://www.springerlink.com/(qicg0vjkkft1rx45fhpj0j45)/app/home/contribution.asp)

Wharton, C., J. Rieman, et al. (1994). The Cognitive Walkthrough Method: A Practitioners Guide. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Virzi, R. A. (1997). Usability Inspection Methods. Handbook of Human-Computer Interaction. M. G. Helander, T. K. Landauer and P. V. Prabhu. Amsterdam, Elsevier Science.

Wixon, D. and C. Wilson (1997). The Usability Engineering Framework for Product Design and Evaluation. Handbook of Human-Computer Interaction. M. Helander, T. Landauer and P. Prabhu. Amsterdam, Elsevier Science B.V: 653-688.

4 Summary of the methods

Method	Lifecycle step	Users needed	Main advantage	Main disadvantage	References
Step 1: Define user requirements and context of use					

Interviews	Task analysis,	3-10
		(most often used as part of user-based testing)
			Flexible, in-depth attitude and experience probing	Time consuming, hard to analyze and compare	Nielsen 1993;
					Maguire 1998;
					Preece 1994

Observation	Task analysis, follow up studies	3 or more	Ecological validity, reveals users real tasks, suggests functions and features	Appointments hard to set up, no experimenter control	Drury 1995;
					Maguire 1998;
					Nielsen 1993;
					Rubin 1994;

Literature study	Concept definition	None	Possibilities to utilise earlier research in concept definition	Could be time consuming	
Brainstorming	Concept definition, early design	4-8	Quick	Group ideation does not suit everyone.	Jones 1980; Maguire 1998
Evaluation of the usage of corresponding products	Early design	3 or more	Helps to understand the earlier designs in actual usage		
Card sort	Early design	3 - 10	Helps structure interface content.	May not correlate with real usage situations.	Brassard M. (Ed.) 1988
Task Analysis	Early design	At least 1	Provides knowledge of the tasks that the user wishes to perform.	Time consuming. Produces much data requiring considerable effort to analyse.	Kirwan et al. 1992
Contextual Inquiry	Concept definition, early design	At least 1	Direct observation of work setting. Allows analyst to make accurate interpretations of data.	More time-intensive than traditional interviewing techniques.	Beyer et al. 1998
Scenarios	Early design, iterative design, final testing	3 or more (depends of the design stage)	Easy and cheap to build. helps users to adapt to the usage situation.		Carroll (Ed.) 1995; Rosson et.al. 2002

QOC	Early design, concept definition	Representative of each stakeholder	Analyse scenarios systematically	Time consuming	MacLean et al. 1991
Step 2: Visualise design ideas					

Visualisation of scenarios	Early design		Allows the designers to communicate effectively with the users.		Aikio et al. 2005;
					Androile 1991;
					Carroll (Ed.) 1995;
					Dumas et al. 1993;
					Maguire 1998;

Use cases	Iterative design		Simplify the traceability between different object models.		Jacobson 1995
Storyboarding	Early design		Simple and cheap. Users can evaluate the direction that the interface is heading.	Can only roughly display the system interaction.	Androile, S. 1991
Multimedia based visualisation	Early design		Allows the designers to communicate effectively with the users.	Laborious, requires expertise of video and audio technologies	
Step 3: Carry out user evaluations of designs					

Focus Groups	Early design, iterative design, formative evaluation. Task analysis and user requirements.	4-8	Spontaneous reactions and group dynamics. Allows find out opinions or factors to be incorporated in other methods (e.g., surveys).	Hard to analyze. Low validity.	Caplan 1990;
					Ikonen et al. 2005;
					Maguire 1998;

Group Discussions/ Future Workshops	Concept definition, early design	3-6 participants in group	Help to summarise the ideas and information held by individual members.	Group ideation does not suit everyone.	Poulson et al. 1996
Step 4: Create concepts and prototypes					
Paper Prototyping	Early design		Cost-effective, early detection of problems; fast changes can be made	Doesn't demonstrate technical capability; the "response time" is artificial	Dumas et al. 1993
Wizard of Oz Prototyping	Early design, iterative design		Save time on programming. E	People need to be trained to act	Maudsley et al. 1993

			xtra understanding of users can be achieved.	as computers. Less realistic.	
Video Prototyping	Early design		Presents design ideas realistically.	Video prototypes do not support the evaluation of fine design detail.	Vertelney 1989
Rapid Prototyping	Early design, iterative design		Allows users to visualise future systems		Maguire 1996
Proof-of-concept demonstrators	Early design		Allows users to visualise future systems		
Step 5: Evaluate concepts and prototypes					

Laboratory usability testing	Early design, iterative design	At least 3	Pinpoint user misconceptions, cheap test	Unnatural for users, hard for expert users to verbalise	Dumas et al. 1993;
					Karat 1997;
					Lindgaard 1994;
					Mayhew 1999;
					Nielsen 1993;
					Rubin 1994;
					Wixon et al. 1997

Field trial	Final testing, follow up studies	At least 3	Evaluation in real usage environment may reveal problems that can not be found in laboratory tests.	Laborious, expertise needed.	Beyer et al. 1998;
					Karat 1997;
					Nielsen 1993;

Heuristic evaluation	Early design	None
		(3-5 usability experts)
			Finds individual usability problems, can address expert user issues, quick and cost beneficial method	Doesn't involve real users, doesn't find surprises related to their needs	Jefferies et al. 1991;
					Nielsen et al. 1990;
					Nielsen 1993;
					Nielsen 1994;

Cognitive Walkthrough	Early design, competitive analysis	None (usability experts)	Specific focus on ease of learning	Doesn't involve real users, labor-intensive, doesn't find surprises related to their needs	Riihiaho 2000;
					Wharton et al. 1994;

5 UCD in organisational context

Even though the importance of UCD and usability of the products and systems has been widely accepted in industry, initiating and facilitating UCD in development has still proven out to be problematic in practice. Many different kinds of problems have been identified that people initiating and facilitating UCD in their organization should be aware of, those being e.g.

- Even identifying and getting in contact with the users might be difficult in the development of commercial SW products or services for external use (e.g. Grudin 1991; Grudin 1993), i.e. in the development of Nomadic Media type of products and services. The development might also be totally organizationally isolated from the users, and the marketing might be the only group in contact with the users (or only with customers) (e.g. Beyer et al. 1998; Cooper 1999; Grudin 1991; Grudin 1993, Mayhew 1999b). These might be perceived as initial problems related to UCD, but starting to apply UCD methods in practice should solve these problems.

- Many times the development cycle is also very short, due to which UCD and iteration of design solutions might be resisted. Related to this, there is also a tendency to apply UCD too late, therefore having no effects on design (e.g. only usability testing is carried in the end of a project). (Aucella 1997; Bodker et al. 2002; Cooper 1999; Gronbak et al. 1993; Grudin 1993; Jokela 2005; Kyng 1994; Poltrock et al. 1994; Rosson et al. 2002.).

- Furthermore, the development organizations may have no prior knowledge of UCD, users and the context of use, due to which the huge amount of new knowledge needs to be acquired in the development (Gronbak et al. 1993; Jokela 2005; Kyng 1994; Rosenbaum et al. 2000; Tudor 1998).

- Altogether, the difficulty of getting UCD accepted in organizations has been highlighted. The position of the usability specialists might be weak, their credibility questioned and their work undervalued, and UCD might be perceived as expensive and delay. (Aucella 1997, Bloomer et al 1997; Jokela 2005; Kyng 1994; Mayhew 1999b; Rosenbaum et al. 2000, Tudor 1998.).

- Finally, multidisciplinary teamwork and cooperation between different organizational units and occupational communities might be difficult due to the differences between developers and users, between different members of multidisciplinary design teams and between different organizational units (Gronbak et al. 1993; Grudin 1993; Heinbokel et al. 1996; Mayhew 1999a; Poltrock rt al. 1994).

Due to the challenges, UCD literature has provided guidelines for how to successfully introduce and facilitate UCD in development organizations. These guidelines are summarized as follows:

- First of all, it is important that UCD contributes to the business success of the development organization (Beyer et al. 1998; Bloomer et al. 1997; Cooper 1999; Fellenz 1997; Mayhew 1999a; Mayhew 1999b; Rosenbaum et al. 2000). Generally, usability is postulated as an important competitive edge in maturing SW markets (Cooper 1999; Grudin 1991; Nielsen 1993; Rosson et al. 2002) – related to the NM type of products and services developed for consumer markets usability and positive user experience surely are important factors. One should, nevertheless, take care that

UCD makes sense from the business perspective and is related to achieving key business goals (Beyer et al. 1998; Bloomer et al. 1997; Cooper 1999; Fellenz 1997). Related to this, one should assess the value of UCD in relation to other driving forces in the development organization. Decisions made in this phase (what, how much, where and why) will 'set the scene' for the prospective implementation of UCD.

- Developers are typically the most important target group, who should perceive usability specialists as team members and allies (not as preachers and criticizers of their work). The developers should take part in UCD in order to ensure that UCD really has effects on the design. (Aucella 1997; Bloomer et al. 1997; Cooper 1999, Fellenz 1997; Jokela 2005; Mayhew 1999a; Mayhew 1999b, Rosenbaum et al. 2000, Rosson et al. 2002; Tudor 1998). Therefore, it is very important to utilize UCD methods and techniques in which the developers can participate in the process and get in contact with the users.

- However, also management commitment and support is an important criterion for success. A high level champion is needed to allow UCD to have authority, autonomy and access to development. (Beyer et al. 1998; Cooper 1999; Fellenz 1997; Nielsen 1993). Therefore, when a decision to implement UCD into the development is made, one should take care that a high level champion at the managerial level is also available.

- Also a strong, experienced, professional group of usability specialists is needed (Aucella 1997; Fellenz 1997; Jokela 2005; Mayhew 1999a; Vredenburg et al. 2002). Carrying out UCD activities always necessitates specific education, skills and competencies, due to which it is important to make sure that in the UCD team there are professional UCD experts - not only inexperienced people getting only to know UCD.

- In addition, creation of documentation of best practices and UCD methods and techniques, and a formal development process with UCD included are important. Also the results of usability activities should be documented and made available. (Aucella 1997, Boivie et al. 2003; Fellenz 1997, Mayhew 1999a; Mayhew 1999b; Nielsen 1993; Vredenburg 1999.) One should note, however, that an appropriate set of best practices, methods and tools depend on the development context (type of product, type of project, size of the company, etc.)

- Furthermore, cooperation with marketing, sales, training, documentation etc. should be initiated to ensure multidisciplinary and cross-functional teamwork. Related to this, it is argued that usability specialists should act as change agents addressing many different target groups in their organizations. They should be able to tailor their message and present their results in languages each target audience understands. (Beyer et al. 1998; Bloomer et al. 1997; Cooper 1999; Hutchings et al. 1995; Grudin 1991; Mayhew 1999a; Mayhew 1999b; Poltrock et al. 1994, Rosenbaum et al. 2000.)

- Altogether, it is argued that 'usability' should be sold into organisation, related to which important is to show the benefits achieved. Also cost-benefit tradeoffs may play a major role in the adoption of UCD - low cost methods are preferred. Finally, the resources for UCD should be well planned and budgeted to make sure that UCD does not increase the development costs and time. (Aucella 1997; Bloomer et al. 1997;

Cooper 1999; Mayhew 1999a; Mayhew 1999b; Nielsen 1993; Rosenbaum et al. 2000; Rosson et al 2002; Vredenburg et al. 2002.).

- Some studies also argue that one should understand the context in which UCD is to be facilitated thoroughly in order to select the most suitable strategy. It is highlighted that if one is to change the development practice one should understand the current practice and context in depth to be able to define how to define and implement the new practice including UCD. The social, political, organizational and cultural context should be taken into account. (Bekker et al. 1996; Beyer et al. 1998; Bodker et al. 2002; Boivie et al. 2003; Cooper 1999; Grudin 1993; Gärtner et al. 1996; Heinbokel et al. 1996; Hutchings et al. 1995; Jokela 2005; Symon 1998; Rosson et al. 2002; Tudor 1998.).

- Related to taking into account the cultural context of the development organizations, one should understand the particular usability myths, beliefs, values and attitudes in the organization, because they define the usability culture of the organization, and they may act as cultural obstacles in the facilitation of UCD (Bloomer et al. 1997; Mayhew 1999a). Bloomer and Croft (1997) suggest that one should understand the particular usability myths and values in the organization, present these myths and values to the personnel of the organization, and afterwards address them by tailoring the message and by focusing on the 'hot buttons' of each target group. Some studies maintain that UCD should be made part of organizations' culture (Catarci et al. 2002; Rosenbaum et al. 2000). Other studies highlight that the UCD facilitation strategy should be aligned with the context, because different strategies work in different projects and political environments (Gärtner et al 1996; Symon 1998; Tudor 1998). It has been argued that the facilitation of UCD likely succeeds if it is customized to the existing culture, because there is no 'one size fits all' (Aucella 1997; Symon 1998).

- One specific approach to start the improvement a company's performance in UCD practices is to carry out a *usability maturity assessment* (Taylor et al. 1998; ISO/IEC 2000; Kurosu et al 2000; Bevan et al. 2001, Dzida et al 2001; Jokela 2004). In such an assessment, the UCD practices of a company are contrasted against an 'ideal' model of UCD. The results of an assessment point out the strengths and weaknesses of the company in UCD, typically both quantitatively and qualitatively. The hypothesis is that knowing the status of UCD helps the management and usability practitioners where to focusing the improvement actions of the UCD practices in a company.

Based on the assumption that 'there is not one size fits all', culturally compatible strategies for UCD have been proposed (Iivari, in press) in relation to a competing values model (discussed extensively in Woodman et al. 1991) outlining four different culture types organizations can reflect: group, adhocracy, rational and hierarchical (see figure 5.1). The group culture emphasizes flexibility and has an internal focus emphasizing the development of people in the organization. Cohesion and morale are needed for human resource development. Additional values are the sense of belonging, trust, participation, openness, teamwork and a sense of family. The adhocracy culture, on the other hand, also emphasizes flexibility, but the focus is on external environment. Flexibility and readiness to change are necessary to achieve growth and resource acquisition. Additional values are innovation, adaptation, creativity, entrepreneurship and dynamism. Furthermore, the hierarchical culture emphasizes control and has an internal focus. Information management and coordination are necessary to achieve stability and equilibrium. Factors such as measurement, documentation, order, rules, regulation, uniformity and smooth operation are valued. Finally, the rational culture type puts emphasis on control and external orientation. Planning and goal setting are ways of achieving productivity and efficiency. Other valued factors are accomplishment, productivity, decisiveness, competitiveness and market superiority. (Cameron et al. 1991; Denison et al. 1991.)

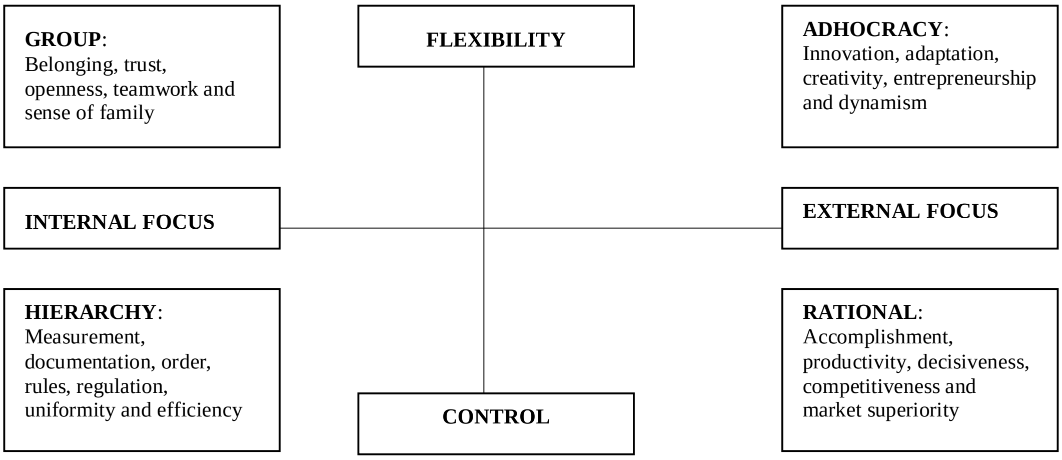

GROUP: Belonging, trust, openness, teamwork and sense of family
FLEXIBILITY
ADHOCRACY: Innovation, adaptation, creativity, entrepreneurship and dynamism
INTERNAL FOCUS
EXTERNAL FOCUS
HIERARCHY: Measurement, documentation, order, rules, regulation, uniformity and efficiency
CONTROL
RATIONAL: Accomplishment, productivity, decisiveness, competitiveness and market superiority

Figure 5.1. Competing values model (based on Cameron - Freeman, 1991, Denison - Spreitzer 1991)

The recommendations for practice presented in the studies that utilize the competing values model in the analysis of organisational culture in relation to different kinds of change efforts have been reviewed (Iivari, in press). They are considered as preliminary guidelines for the prospective facilitators of UCD, since in the context of UCD, studies of this kind have not been carried out yet. The literature suggests the following strategies as suitable for different culture types, i.e. one should emphasize the following aspects depending on the culture type the organization reflects the most:

- In the group culture type, one should select methods and tools (for UCD) that emphasize informal information sharing, training, teamwork and employee ownership. Participation in the decision-making, morale and trust should be encouraged;

- In the adhocracy culture type, one should select methods and tools (for UCD) that emphasize innovation, experimentation, the creation and sharing of new ideas, and freedom to take risks. UCD methods, which rely on teamwork, brainstorming and iteration, are suitable. This culture type is an ‘early adopter’. This strength should be utilized and new, innovative solutions searched for and experimented with;

- In the hierarchy culture type, one should select methods and tools (for UCD) that emphasize rules, standard procedures, documentation and control (of the product and the process). For example implementation of a process model with UCD integrated could be suitable in this culture type. In addition, this culture type is suitable for any kind of large-scale improvement programs, since procedures and rules are already in place;

- Finally, in the rational culture type, one should select methods and tools (for UCD) that emphasize measurement and cost benefit analyses that show the business benefits of the change effort – including UCD. In addition, in this context the change efforts should always be closely linked to the business strategy and objectives.

Suggested resources for additional reading:

Aucella. A. (1997): Ensuring Success with Usability Engineering. Interactions May + June. Pp.19-22.

Bekker, M. M. and Vermeeren, A. P. O. S. (1996): An analysis of user interface design projects: information sources and constraints in design. Interacting with Computers 8(1). Pp. 112-116.

Bevan, N. and J. Earthy (2001). Usability process improvement and maturity assessment. IHM-HCI 2001, Lille, France, Cépaduès-Editions, Toulouse

Beyer, H. and Holtzblatt, K. (1998): Contextual Design: Defining Customer-Centered Systems. San Francisco: Morgan Kaufmann Publishers.

Bloomer, S. and Croft, R. (1997): Pitching Usability to Your Organization. Interactions November + December. Pp. 18-26.

Bødker, S. and Buur, J. (2002): The Design Collaboratorium – a Place for Usability Design. ACM Transactions on Computer-Human Interaction 9(2). Pp. 152-169.

Boivie, I., Åborg, C., Persson, J. and Löfberg, M. (2003): Why usability gets lost or usability in in-house software development. Interacting with Computers 15. Pp. 623-639.

Cameron, K. S. and Freeman, S. J. (1991): Cultural Congruence, Strength, and Type: Relationships to Effectiveness. In Richard W. Woodman and William A. Pasmore (Ed.). Research in Organizational Change and Development. An Annual Series Featuring Advances in Theory, Methodology, and Research 5. Greenwich: JAI Press. Pp. 23-58.

Catarci, T., Matarazzo, G. & Raiss, G. (2002): Driving usability into the public administration: the Italian experience. International Journal of Human-Computer Studies 57. Pp. 121-138.

Cooper, A. (1999): The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity. Indianapolis: Sams.

Denison, D. R. and Spreitzer, G. M. (1991): Organizational Culture and Organizational Development: A Competitive Values Approach. In Richard W. Woodman and William A. Pasmore (Eds.). Research in Organizational Change and Development. An Annual Series Featuring Advances in Theory, Methodology, and Research 5. Greenwich: JAI Press Inc. Pp. 1-21.

Dzida, W., T. Geis and R. Freitag (2001). The DATech Usability-Engineering and Testing Process based on ISO 13407. Proceedings of the 6th Congress on Software Quality Management, Cologne, Germany

Fellenz, C. B. (1997): Introducing Usability into Smaller Organizations. Interactions September/October. Pp. 29-33.

Gronbak, K., Grudin, J., Bødker, S. and Bannon, L. (1993): Achieving Cooperative System Design: Shifting From a Product to a Process Focus. In D. Schuler & A. Namioka (Eds.): Participatory Design: Principles and Practices. New Jersey: Lawrence Erlbaum Associates. Pp. 79-98.

Grudin, J. (1991): Interactive Systems: Bridging the Gaps Between Developers and Users. IEEE Computer 24(4). Pp. 59-69.

Grudin, J. (1993): Obstacles to Participatory Design in Large Product Development Organizations. In D. Schuler & A. Namioka (Eds.): Participatory Design: Principles and Practices. New Jersey: Lawrence Erlbaum Associates. Pp. 99-122.

Gärtner, J. and Wagner, I. (1996): Mapping Actors and Agendas: Political Frameworks of Systems Design and Participation. Human-Computer Interaction 11. Pp. 187-214.

Heinbokel, T., Sonnentag, S., Frese, M., Stolte, W. and Brodbeck, F. C. (1996): Don't underestimate the problems of user centredness in software development projects – there are many! Behaviour & Information Technology 15(4). Pp. 226-236.

Hutchings, A. F. and Knox, S. T. (1995): Creating Products - Customer Demand. Communications of the ACM 38(5). Pp. 72-80.

Iivari, N. (in press): 'Representing the User' in Software Development – A Cultural Analysis of Usability Work in the Product Development Context. Accepted for publication in Interacting with Computers.

ISO (2000). 18529 Human-centred Lifecycle Process Descriptions. ISO/IEC TR 18529: 2000

Jokela, T. (2004). "Evaluating the user-centredness of development organisations: conclusions and implications from empirical usability capability maturity assessments." Interacting with Computers. Interacting with Computers 16(6): 1095-1132.

Jokela, T. (2005): Performance rather than capability problems. Insights from assessments of usability engineering processes. In the proceedings of Profes 2005 – 6th International Conference on Product Focused Software Process Improvement. Oulu, Finland, June 13th – 15th 2005.

Kurosu, M., M. Ito, Y. Horibe and N. Hirasawa (2000). Diagnosis of Human-Centeredness of the Design Process by the SDOS. Proceedings of UPA 2000, Usability Professionals Association, Asheville, North Carolina

Kyng, M. (1994): Scandinavian Design: Users in Product Development. Proc. CHI 1994. New York: ACM Press. Pp. 3-9.

Mayhew, D. J. (1999a): Business: Strategic Development of Usability Engineering Function. Interactions September + October. Pp. 27-34.

Mayhew, D. J. (1999b): The usability engineering lifecycle: a practitioner's handbook for user interface design. San Francisco: Morgan Kaufmann Publishers, Inc.

Nielsen, J. (1993): Usability engineering. Boston: Academic Press.

Poltrock, S. and Grudin, J. (1994): Organizational Obstacles to Interface Design and Development: Two Participant – Observer Studies. ACM Transactions on Computer-Human Interaction 1(1). Pp. 52-80.

Rosenbaum, S., Rohn, J. A. and Humburg, J. (2000): A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys. CHI Letters 2 (1). Pp. 337-344.

Rosson, M. B. and Carroll, J. (2002): Usability Engineering: Scenario-based Development of Human-Computer Interaction. San Francisco: Morgan-Kaufman.

Symon, G. (1998): The Work of IT System Developers in Context: An Organizational Case Study. Human-Computer Interaction 13(1). Pp. 37-71.

Tudor, L. (1998): Human Factors: Does Your Management Hear You? Interaction January + February. Pp. 16-24.

Woodman, R. and Pasmore, W. (Eds.) (1991): Research in Organizational Change and Development. An Annual Series Featuring Advances in Theory, Methodology, and Research 5. Greenwich: JAI Press Inc.

Vredenburg, K (1999): Increasing Ease of Use - Emphasizing organizational transformations, process integration, and method optimisation. Communications of the ACM 42(5). Pp. 67-71.

Vredenburg, K., Mao, J, Smith, P. W. & Casey, T. (2002): A survey of user-centered design practice. In D. Wixon (ed.) Proceedings of Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves, New York: ACM Press. Pp. 471-478.

6 References

Abowd, G., Mynatt, E, and Rodden, T. (2002): The Human Experience, Pervasive Computing, Vol. 1(1), 48-57.

Aikio, K.-P. – Jounila, I. – Jokela, T. (2005): Developing Techniques for Visualising Scenarios. HCII 2005 – 11th International Conference on Human-Computer Interaction. Las Vegas, USA, July 22nd -27th 2005.

Aikio, K.-P. and I. Jounila (2005): Guide for visualizing scenarios. University of Oulu, Department of Computer Sciences (Master's graduate thesis: in Finnish).

Androile, S. (1991): Storyboard Prototyping. Wellesley, MA: QED Information Sciences.

Aucella. A. (1997): Ensuring Success with Usability Engineering. Interactions May + June. Pp.19-22.

Baecker, R. M., Grudin, J. Buxton, W. A. S. and Greenberg, S. (Eds.) (1995): Readings in Human-Computer Interaction: Toward the Year 2000 (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers.

Battarbee, K. (2004): Co-Experience – Understanding user experiences in social interaction. Academic Dissertation. Publication series of the University of Art and Design. Helsinki A 51.

Bekker, M. M. and Vermeeren, A. P. O. S. (1996): An analysis of user interface design projects: information sources and constraints in design. Interacting with Computers 8(1). Pp. 112-116.

Bevan, N. and J. Earthy (2001): Usability process improvement and maturity assessment. IHM-HCI 2001, Lille, France, Cépaduès-Editions, Toulouse

Beyer, H. and K. Holtzblatt (1998). Contextual Design: Defining Customer-Centered Systems. San Francisco, Morgan Kaufmann Publishers.

Bloomer, S. and Croft, R. (1997): Pitching Usability to Your Organization. Interactions November + December. Pp. 18-26.

Bødker, S. and Buur, J. (2002): The Design Collaboratorium – a Place for Usability Design. ACM Transactions on Computer-Human Interaction 9(2). Pp. 152-169.

Boivie, I., Åborg, C., Persson, J. and Löfberg, M. (2003): Why usability gets lost or usability in in-house software development. Interacting with Computers 15. Pp. 623-639.

Brassard, M. (1989). The Memory Jogger Plus: Featuring The Seven Management And Planning Tools Methuen MA., GOAL/QPC.

Buchenau, M. and Fulton-Suri, J. (2000): Experience Prototyping. In Proceedings of DIS´00. ACM, Brooklyn, New York.

Cameron, K. S. and Freeman, S. J. (1991): Cultural Congruence, Strength, and Type: Relationships to Effectiveness. In Richard W. Woodman and William A. Pasmore (Ed.). Research in Organizational Change and Development. An Annual Series Featuring Advances in Theory, Methodology, and Research 5. Greenwich: JAI Press. Pp. 23-58.

Caplan, S (1990): Using Focus Group methodology for ergonomic design. Ergonomics, 33.5, 527-537.

Carroll, J. M. (Ed.) (1995): Scenario-Based Design: Envisioning Work and Technology in System Development. New York: Wiley & Sons.

Carroll, J.M., (Ed.) (2003): HCI Models, Theories, and Frameworks. Toward a Multidisciplinary Science. San Fransico. Morgan Kaumann.

Catarci, T., Matarazzo, G. & Raiss, G. (2002): Driving usability into the public administration: the Italian experience. International Journal of Human-Computer Studies 57. Pp. 121-138.

Cooper, A. (1999): The inmates are running the asylum: Why high-tech products drive us crazy and how to restore the sanity. Indianapolis: Sams.

Courage C. & Baxter K. (2005): A practical guide to user requirements. Methods, Tools & Techniques. Morgan Kaufman Publishers, San Francisco).

Cushman, W.H., & Rosenberg, D.J. (1991): Human factors in product design. Amsterdam: Elsevier.

Denison, D. R. and Spreitzer, G. M. (1991): Organizational Culture and Organizational Development: A Competitive Values Approach. In Richard W. Woodman and William A. Pasmore (Eds.). Research in Organizational Change and Development. An Annual Series Featuring Advances in Theory, Methodology, and Research 5. Greenwich: JAI Press Inc. Pp. 1-21.

Desurvire, H. W. (1994): Faster, Cheaper! Are Usability Inspection Methods as Effective as Empirical Testing? Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Drury, C.G. (1995): Methods for direct observation of performance. In: J. Wilson & E.N. Corlett, Evaluation of human work: A practical ergonomics methodology, Taylor & Frances.

Dumas, J.S., Redish, J.C. (1993): A Practical Guide to Usability Testing. Norwood, NJ: Ablex Publishing.

Dzida, W., T. Geis, T. and Freitag, R. (2001): The DATech Usability-Engineering and Testing Process based on ISO 13407. Proceedings of the 6th Congress on Software Quality Management, Cologne, Germany.

Hutchings, A. F. and Knox, S. T. (1995): Creating Products - Customer Demand. Communications of the ACM 38(5). Pp. 72-80.

Fellenz, C. B. (1997): Introducing Usability into Smaller Organizations. Interactions September/October. Pp. 29-33.

Gronbak, K., Grudin, J., Bødker, S. and Bannon, L. (1993): Achieving Cooperative System Design: Shifting From a Product to a Process Focus. In D. Schuler & A. Namioka (Eds.): Participatory Design: Principles and Practices. New Jersey: Lawrence Erlbaum Associates. Pp. 79-98.

Grudin, J. (1991): Interactive Systems: Bridging the Gaps Between Developers and Users. IEEE Computer 24(4). Pp. 59-69.

Grudin, J. (1993): Obstacles to Participatory Design in Large Product Development Organizations. In D. Schuler & A. Namioka (Eds.): Participatory Design: Principles and Practices. New Jersey: Lawrence Erlbaum Associates. Pp. 99-122.

Gärtner, J. and Wagner, I. (1996): Mapping Actors and Agendas: Political Frameworks of Systems Design and Participation. Human-Computer Interaction 11. Pp. 187-214.

Heinbokel, T., Sonnentag, S., Frese, M., Stolte, W. and Brodbeck, F. C. (1996): Don’t underestimate the problems of user centredness in software development projects – there are many! Behaviour & Information Technology 15(4). Pp. 226-236.

Helander, M., Landauer, T.K., Prabhu, P., eds. (1997). Handbook of human-computer interaction. Amsterdam: Elsevier.

Ikonen, V., Leikas, J., Strömberg, H. (2005): Nomadic Elderly: design requirements for technologically enhanced traveling experience. 5th International Conference of the International Society for Gerontechnology, Nagoya, Japan, May 24-27, 2005.

Iivari, N. (in press): ‘Representing the User’ in Software Development – A Cultural Analysis of Usability Work in the Product Development Context. Accepted for publication in Interacting with Computers.

ISO (2000): 18529 Human-centred Lifecycle Process Descriptions. ISO/IEC TR 18529: 2000

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

ISO/IEC (1999): 13407 Human-Centred Design Processes for Interactive Systems, ISO/IEC 13407: 1999 (E), 1999.

Jacobson, I. (1995): The Use-Case Construct in Object-Oriented Software Engineering. In: J. Carroll (Ed.) Scenario-Based Design. Wiley & Sons.

Jefferies, R., Miller, J.R., Warton, C. & Uyeda, K.M. 1991: User interface evaluation in the real world: a comparison of four techniques. Reaching through technology - CHI'91. Proceedings of the conference on human factors in computing systems, 27 Apr - 2 May 1991, New Orleans, Louisiana pp 119 - 124.

Jokela, T. (2004): When Good Things Happen to Bad Products: Where arethe Benefits of Usability in the Consumer Appliance Market? ACM Interactions XI.6: pp 28-35.

Jokela, T. (2004): “Evaluating the user-centredness of development organisations: conclusions and implications from empirical usability capability maturity assessments.” Interacting with Computers. Interacting with Computers 16(6): 1095-1132.

Jokela, T. (2005): Performance rather than capability problems. Insights from assessments of usability engineering processes. In the proceedings of Profes 2005 – 6th International Conference on Product Focused Software Process Improvement. Oulu, Finland, June 13th – 15th 2005.

Jones, J.C.: Design Methods: Seeds of Human Futures, 1980 edition, John Wiley & Sons, (1980).

Jordan, P.W., Thomas, B., Weerdmeester, B.A., & McClelland, I.L. (1996): Usability evaluation in industry. London: Taylor & Francis.

Karat, C.-M. (1994): A Comparison of User Interface Evaluation Methods. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Karat, J. (1997): User-Centered Software Evaluation Methodologies. Handbook of Human-Computer Interaction. M. G. Helander, T. K. Landauer and P. V. Prabhu. Amsterdam, Elsevier Science.

Kirwan, B and Ainsworth, L.K. (Ed.) (1992): A Guide to Task Analysis. London: Taylor and Francis.

Kuniavsky, M. (2003): Observing the User Experience: A Practioner's Guide to User Research. Morgan Kaufmann.

Kurosu, M., M. Ito, Y. Horibe and N. Hirasawa (2000). Diagnosis of Human-Centeredness of the Design Process by the SDOS. Proceedings of UPA 2000, Usability Professionals Association, Asheville, North Carolina.

Kyng, M. (1994): Scandinavian Design: Users in Product Development. Proc. CHI 1994. New York: ACM Press. Pp. 3-9.

Leikas, J., Väätänen, A., Virtanen, T. & Suihkonen, R. (2005): Design rationale - Bridging the Gap between user requirements and implementation. HCI International 2005 Conference, HCII 2005. Las Vegas, 22 -27 July 2005. HCII. USA (2005).

Leikas, J. and Strömberg, H. (2005): Novel User Interfaces for Promoting Social Interaction in Multiplayer Games. ENACTIVE/05, November 17-18, Genoa, Italy.

Leikas, J., Strömberg, H., Ikonen, V., Suomela, R. and Heinilä, J. (2006): Multi-user Mobile Applications and a Public Display: Novel Ways for Social Interaction. (Accepted). Pervasive Computing and Communications, Pisa, Italy, 13-17 March 2006.

Lindgaard, G. (1994): Usability testing and system evaluation, A guide for designing useful computer systems. Chapman & Hall, UK. Paperback ISBN 0-412-46100-5.

Mack, R. L. and J. Nielsen (1994): Executive summary. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

MacLean, A., Young, R.M., Bellotti, V.M.E., and Moran T.P. (1991): Questions, Options, and Criteria: Elements of Design Space Analysis. Human-Computer Interaction 6(3&4), 201-250

Maguire M. (1996): Prototyping and evaluation guide. HUSAT Research Institute, The Elms, Elms Grove, Loughborough, Leicestershire, LE11 1RG, UK. October.

Maguire, M. (Ed.) (1998): RESPECT User-Centred Requirements Handbook, version 3.2 June 1998.

Maudsley, Greenberg and Mander (1993): Prototyping an intelligent agent through Wizard of Oz. Interchi '93 Conference Proceedings, pp 277-284.

Mayhew, D. J. (1999a): Business: Strategic Development of Usability Engineering Function. Interactions September + October. Pp. 27-34.

Mayhew, D. J. (1999b): The usability engineering lifecycle: a practitioner's handbook for user interface design. San Francisco: Morgan Kaufmann Publishers, Inc.

McClelland, I. and Fulton Suri, J. (2005): Involving people in design. (Chapter in 'Evaluating Human Work 3rd edition'). Editors: J Wilson and N Corlett, Taylor and Francis, London Mayhew, D. J. (1999). The Usability Engineering Lifecycle. San Fancisco, Morgan Kaufman.

Nielsen, J. (1994): Heuristic Evaluation. Usability Inspection Methods. J. Nielsen and R. L. Mack. New York, John Wiley & Sons, Inc.

Nielsen, J. and Sano, D. (1994): SunWeb: User Interface Design for Sun Microsystem's Internal Web.

Nielsen, J., and Molich, R. (1990): Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1-5 April), 249-256.

Nielsen, J. (1993): Usability Engineering. San Diego, Academic Press, Inc.

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

Nomadic Media (2003): FPP-Final Project Proposal.

Poltrock, S. and Grudin, J. (1994): Organizational Obstacles to Interface Design and Development: Two Participant - Observer Studies. ACM Transactions on Computer-Human Interaction 1(1). Pp. 52-80.

Poulson, D., Ashby, M. and Richardson, S. (Eds) (1996: Userfit: A practical handbook on user-centred design for Assistive Technology. Brussels-Luxembourg: ECSC-EC-EAEC.

Practices. New Jersey: Lawrence Erlbaum Associates. Pp. 99-122.

Preece, J. et al. (1994). Human-Computer Interaction. Wokingham. Addison-Wesley.

Riihiaho, S. (2000): Experiences with Usability Evaluation Methods. Computer Science and Engineering, Helsinki University of Technology: 113.

Rosenbaum, S., Rohn, J. A. and Humburg, J. (2000): A Toolkit for Strategic Usability: Results from Workshops, Panels, and Surveys. CHI Letters 2 (1). Pp. 337-344.

Rosson, M.B. & Carroll, J.M., (2002): Usability engineering. Scenario-based development of human-computer interaction. San Francisco, CA: Morgan Kaufmann Publishers.

Rubin, J. (1994): Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests. New York: Wiley & Sons.

Rutkowski. C. (1982): An introduction to the Human Applications Standard Computer Interface, Part I: Theory and principles. Byte, 7 (10), October 1982, 291-310.

Scholtz, J., Arnstein, L., Kim, M., Kindberg, T. & Consolvo, S.: "User-centered Evaluations of Ubicomp Applications," Intel Research, IRS-TR-02-006, May. 1, 2002.

Shneiderman, B. (1998). Designing the User Interface: Strategies for Human-Computer Interaction (3d edition). Reading, MA: Addison-Wesley.

Stanton, N., ed. (1998). Human factors in consumer products. London: Taylor & Francis.

Strömberg, H., Leikas, J., Suomela, R., Ikonen, V. and Heinilä, J. (2005): Multiplayer gaming with mobile phones - enhancing user experience with a public display. Intetain (Intelligent technologies for interactive entertainment); Madonna di Cambiglio 30.11.- 2.12.2005, Italy. http://www.springerlink.com/(qicg0vjkkft1rx45fhpj0j45)/app/home/contribution.asp

Symon, G. (1998): The Work of IT System Developers in Context: An Organizational Case Study. Human-Computer Interaction 13(1). Pp. 37-71.

Tudor, L. (1998): Human Factors: Does Your Management Hear You? Interaction January + February. Pp. 16-24.

Vertelney, L. (1989): Using video to prototype user interfaces. SIGCHI Bulletin, 21, 2, 57-61.

Wharton, C., Rieman, J., Lewis, C., & Polson, P. (1994): The cognitive walkthrough method: A practitioner's guide. In Nielsen, J., & Mack, R. L. (Eds.), Usability inspection methods, 105-140. New York, NY: John Wiley & Sons.

Virzi, R. A. (1997): Usability Inspection Methods. Handbook of Human-Computer Interaction. M. G. Helander, T. K. Landauer and P. V. Prabhu. Amsterdam, Elsevier Science.

Vredenburg, K., Isensee, S., and Righi, C. (2002): User-Centered Design: An Integrated Approach. New Jersey. Prentice Hall.

Vredenburg, K (1999): Increasing Ease of Use – Emphasizing organizational transformations, process integration, and method optimisation. Communications of the ACM 42(5). Pp. 67-71.

Vredenburg, K., Mao, J, Smith, P. W. & Casey, T. (2002): A survey of user-centered design practice. In D. Wixon (Ed.) Proceedings of Conference on Human Factors in Computing Systems: Changing our World, Changing Ourselves, New York: ACM Press. Pp. 471-478.

Wilson, J.R., & Corlett, E.N. (1995): Evaluation of human work. A practical ergonomics methodology. London: Taylor & Francis.

Standards:

ANSI (2001). Common Industry Format for Usability Test Reports. NCITS 354-2001.

ISO/IEC (1998). 9241-11 Ergonomic requirements for office work with visual display terminals (VDT)s - Part 11 Guidance on usability. ISO/IEC 9241-11: 1998 (E).

ISO/IEC (1999). 13407 Human-Centred Design Processes for Interactive Systems. ISO/IEC 13407: 1999 (E).

ISO/IEC (2000). 18529 Human-centred Lifecycle Process Descriptions. ISO/IEC TR 18529: 2000 (E).

ISO/IEC (2002). ISO/TR I6982 Usability methods supporting human-centred design. ISO/IEC 16982: 2002 (E)

Journals:

ACM publications (info on ACM website) including
Transactions on Computer-Human Interaction,

Interactions,

Communications of the ACM and

Transactions on Information Systems

Applied Ergonomics

Behaviour and Information Technology

Computers and Human Behavior

Human-Computer Interaction

Interacting with Computers

International Journal of Human-Computer Interaction

International Journal of Human-Computer Studies

Journal of Applied Psychology

Journal of Experimental Psychology

Personal and ubiquitous computing

User modeling and user-adapted interaction

Websites

www.usabilitynet.org (overview)

http://www.hcibib.org/ (listing of numerous resources regarding HCI)

http://degraaff.org/hci/ (listing of numerous resources regarding HCI)

http://www.best.com/~jthom/usability (compilation of usability test techniques & methods)

http://www.usabilityhome.com/ (similar to the previously listed site)

http://www.useit.com/papers/ (contains several papers by Jakob Nielsen on usability and usability testing)

http://usableweb.com/ (compilation of HCI resources and usability test techniques)

http://www.uie.com/ (compilation of HCI resources and usability test techniques)

http://www.stcsig.org/usability/resources/ (various resources and tools for usability testing)

http://www.ejeisa.com/nectar/inuse (inuse handbook of user-centered design)

http://hcibib.org/tcuid/ (introduction to UCD)

http://www.usability.serco.com/trump/methods (a selection of UCD methods)

http://www.usabilityprofessionals.org/usability_resources/guidelines_and_methods (UCD methods and guidelines overview)