Ma 116

ACTA POLYTECHNICA SCANDINAVICA

MATHEMATICS AND COMPUTING SERIES No. 116

User Studies: A Practical Approach to User Involvement for Gathering User Needs and Requirements

SARI KUJALA

Helsinki University of Technology
Department of Computer Science and Engineering
P.O.Box 9600
FIN-02015 HUT
FINLAND

Dissertation for the degree of Doctor of Philosophy to be presented with due permission of the Department of Computer and Engineering for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 7th of June 2002, at 12 o'clock noon.

ESPOO 2002

Kujala, S., **User Studies: A Practical Approach to User Involvement for Gathering User Needs and Requirements**, Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 116, Espoo 2002, 132 pp. Published by the Finnish Academies of Technology. ISBN 951-666-599-3, ISSN 1456-9418.

Keywords: User involvement, usability, user needs, user requirements, requirements elicitation

ABSTRACT

This thesis investigates the role of user involvement in the early phases of product development. It is generally believed that usability and more accurate user requirements are achieved through the involvement of potential users in product development. First, the benefits and challenges of user involvement identified in the literature were reviewed. It was discovered that early user involvement has positive effects on user and customer satisfaction and requirements quality, but it may additionally have negative effects on product development time and cost.

A practical approach to early user involvement referred to as 'user study' was synthesised to find a way to apply cost-effectively early user involvement to real product development contexts. The goal of the user study is cost-effectively to gather data on users and their needs and to translate them to user requirements that support the development of useful and usable products.

The user study approach was then evaluated in four case studies in five different product development companies. The first and second study focused on the usefulness of user studies. The third study investigated introducing the user study approach to a real product development context. The fourth study concerned representing the results of user studies: bridging the gap between user needs and user requirements. The results presented in the thesis reveal that early user involvement is useful even in a short time frame with relatively low costs. The results additionally provide further support for the successful implementation of user involvement in the early phases of the product development.

To Joose

Supervisor

Professor Reijo Sulonen
Department of Computer Science and Engineering
Helsinki University of Technology
Espoo, Finland

Reviewers

Ph.D. Janice (Ginny) Redish
President
Redish & Associates, Inc.
Bethesda, Maryland, USA

Professor Larry E. Wood
Instructional Psychology & Technology
Brigham Young University
Provo, Utah, USA

Opponent

Professor Kari Kuutti
Department of Information Processing Science
University of Oulu
Oulu, Finland

Contents

Acknowledgements

I have been fortunate to be able to work with a number of wonderful people and interesting and fruitful subject matter. I would like to thank my supervisor, Professor Shosta Sulonen, for his valuable advice and encouragement. I am grateful to my former supervisor, Professor Martti Mäntylä, for guiding me to consider the early phases of the product development and his advice and help throughout the work. I also wish to thank the pre-examiners of this thesis, Ph.D. Ginny Redish and Professor Larry Wood, for their valuable comments.

My warm thanks go to the QURE research team: Marjo Kauppinen, Jarmo Hurri, Virve Leino, Tapani Aaltio, and Laura Lehtola. I would particularly like to thank Marjo as she has been my closest co-worker for an extended period. With her incredible personality and charisma, she has encouraged others. She has read and commented on texts, and we have worked on ideas together. She is moreover an excellent project manager and thus, she has made the project work easier for others. The entire SoberIT community has participated in creating a pleasant working atmosphere.

In addition, I would like to thank the "Usability of Smart Products in Information Society" research team: Irmeli Sinkkonen, Marko Nieminen, Sirpa Riihiaho, Simo Säde, and Turkka Keinonen. They and the whole usability group of the time, Anu Kankainen, Jarmo Parkkinen, Hannu Kuoppala, and many others, have been important in my early steps in the usability field.

The research work would not have been possible without the industrial partners and the participating individuals. I thank all the participating companies and people; the spirit has been good and enthusiastic. I would like to thank Kalle Saarinen, Mika Koski, Eija Suikola, and Pirkko Jokela from Teamware, Sanna Rekola, Samu Salmelin, Tomi Sipilä, Pekka Hakola, Ismo Roponen, Arttu Salmenhaara, Juha-Pekka Hienonen, Salla Mannerkoski, and Antti Laine from KONE Corporation, Ville Otra-Aho, Lauri Tuomaala and Jarmo Harju from Vaisala, and Pia Nakari, Jari Sundqvist and Riitta Rautava from Tekla. All participated directly in case studies or they made the studies possible. I am also grateful to the many others from these and other companies who participated in meetings and discussions.

The work has been financed from two sources: the Academy of Finland and Tekes, the National Technology Agency, both of whom are gratefully acknowledged.

Finally, I wish to thank my husband, Arto, for his loving support and comments on a draft of this thesis and for introducing me to software engineering, and my parents, who encouraged me.

Espoo, April 2002
Sari Kujala

List of publications

This thesis is based on the following publications, which are referred to in the text by Roman numerals.

I Kujala, S. (2002). User Involvement: A Review of the Benefits and Challenges. In Soininen, T. (Ed.), Preprints, Software Business and Engineering Institute, Helsinki University of Technology, Report no.: HUT-SoberIT-B1. Espoo, Finland, pp. 1-32.

II Kujala, S. and Mäntylä, M. (2000). Studying Users for Developing Usable and Useful Products. In Gulliksen, J., Oestreicher, L., Severinson Eklundh, K. (Eds.), NordiCHI2000, Design versus Design (Proceedings of 1st Nordic Conference on Computer-Human Interaction), pp. 1-11.

III Kujala, S. and Mäntylä, M. (2000). How Effective Are User Studies? In McDonald, S., Waern, Y., and Cockton, G. (Eds.), People and Computers XIV (Proceedings of Human-Computer Interaction 2000 Conference), Springer-Verlag, pp. 61-71.

IV Kujala, S., Kauppinen, M., and Rekola, S. (2001). Bridging the Gap between User Needs and User Requirements. In Avouris, N. and Fakotakis, N. (Eds.) Advances in Human-Computer Interaction I (Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction PC-HCI 2001), Typorama Publications, pp. 45-50.

V Kujala, S., Kauppinen, M., and Rekola, S. (2001). Introducing User Needs Gathering to Product Development: Increasing Innovation and Customer Satisfaction. In Hirose, M. (Ed.), Human-Computer Interaction INTERACT '01 (Proceedings of Interact 2001 Conference), IOS Press, pp. 856-861.

Terms and definitions

The basic terms used in this thesis are described here. The Finnish translations of the terms are shown in brackets.

Context of use (käyttöyhteys) – Users, tasks, equipment, and physical and social environment in which a product is used (ISO 9241-11: 1998; ISO 13407: 1999).

Customer (asiakas) – The person, or persons, who pay for the product and usually (but not necessarily) decide the requirements (IEEE Std 830: 1998).

Requirement elicitation (vaatimusten hankinta) – The process of discovering the requirements through consultation with stakeholders, from system documents, domain knowledge, and market studies (Sommerville and Sawyer, 1997, p. 11).

Requirements engineering (vaatimustenmäärittely ja -hallinta) – Requirements engineering covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a system (Sommerville and Sawyer, 1997).

Usability (käytettävyys) – The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use (ISO 9241:11; ISO 13407).

Use case (käyttötapaus) – A way of using the system. A use case describes the possible sequences of interactions among the system and one or more actors in response to some initial stimulus by one of the actors (Rumbaugh, 1994).

User (käyttäjä) – An individual interacting with the system (ISO 9241-11: 1998; ISO 13407: 1999).

User-centred design (käyttäjäkeskeinen suunnittelu) – The attitudes and approaches used for developing usable systems (Karat, J., 1997, p. 35).

User involvement (käyttäjien osallistuminen tai mukaantulo) – A general term describing direct contact with users and covering several approaches.

User needs (käyttäjätarve) – User needs refer to problems that hinder users in achieving their goals, or opportunities to improve the likelihood of users' achieving their goals. An important factor affecting on user needs is the context of use.

User requirement (käyttäjävaatimus) – Any function, constraint, or other property that is required in order to satisfy user needs. User requirements are elicited from users and described from the user and customer point of view.

1. Introduction

Understanding customer needs is frequently seen as a success factor in product development. In particular, it is important in the beginning of the development, when requirements are defined. If customer needs are poorly understood in the beginning, it leads to rework later. For example, Blackburn et al. (2000) found that more time and effort invested in the early stages of a software project yields faster cycle times and higher productivity.

The source of information plays an important role in understanding customer needs. Keil and Carmel (1995) found that more successful software development projects employed more direct links to users and customers. Similarly, Chatzoglou and Macaulay (1996) show that users as the main source of information decreased the number of iterations needed. A user is an individual interacting with the product, whereas a customer is a person who pays for the product or who orders it. Even if customers and users have different motivations toward products, the roles are frequently overlapping and both groups are important information sources and stakeholders in product development. Customers are generally considered to be the more important, because they pay for the product. However, users are also important, because they use the ultimate product. In addition, as Coble et al. (1997) point out, the customer's primary goal is generally to buy a system, which supports users in their tasks. In this thesis, the focus is on users.

Usability is a key concept in understanding how to support users in their tasks. It represents the extent to which a product can be used by specified users to achieve their goals with effectiveness, efficiency, and satisfaction in a specified context of use (ISO 9241-11: 1998; ISO 13407: 1999). The benefits of usability can include increased productivity, enhanced quality of work, reductions in support and training costs, and improved user satisfaction (ISO 13407: 1999). This broader definition of usability implies that, in order to develop a successful product, a designer needs to understand what kinds of persons will use the product, what they want to achieve, and in what context they will use it. According to this view, it is apparent that direct contact with users is crucial.

Thus, it is generally agreed that usability is achieved through the involvement of potential users in system design (Karat, J., 1997; Bekker and Long, 2000; Wilson et al., 1997). Moreover, in theory, user involvement is most efficient and influential in the early stages of system development as the cost involved in making changes increases during system development (cf. Ehrlich and Rohn, 1994; Noyes et al., 1996).

However, as John Karat (1997) points out, it is less apparent what the best techniques in user participation are. Real product development operates under heavy time and resource constraints. The results of Heinbokel et al. (1996) suggest that user involvement may even disturb the process of software development.

This thesis concentrates on how to apply early user involvement in real product development contexts. The focus is on products that include software and are developed to a large market of customers. The particular interest is in how to study users in order to gather user needs and represent them in user requirements.

The following sections give a background to the presented papers, describe the essential notions, and state the research problems and goals.

1.1 Research focus

The research presented in this thesis is carried out within the disciplines of human-computer interaction (HCI) and requirements engineering (RE). Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them (ACM SIGCHI, 1996). HCI is an interdisciplinary area, where the focus is on interaction between humans and computational machines.

Requirements engineering (RE) is a sub-discipline of software engineering. Software engineering is concerned with methods, tools, and techniques for developing and managing the process of creating software products (Sommerville, 1996). Requirements engineering focuses on the early phases of software development, where decisions are made on what to implement, and where the foundation is laid for the later phases that determine how to implement it (Regnell, 1999).

Both HCI and RE aim to develop useful and usable computer systems, but currently they form two distinct disciplines with different concepts and methods (Coutaz and Taylor, 1994). RE has a focus on development, and the focus of HCI is on usability.

Gould and Lewis (1985) introduce the principles of user-centred design, but only in recent years has the HCI community generally adopted the title "User-Centred Design" for describing the approaches used for developing usable systems. Usability engineering frequently has a narrower focus on defining measurable usability goals and testing the product against those goals (Wixon and Wilson, 1997). Tyldesley (1988) defines usability engineering as "a process whereby the usability of a product is specified quantitatively, and in advance". Wixon and Wilson (1997) clarify the definition of usability engineering as a process for defining, measuring, and thereby improving, the usability of products.

In addition, usability is one of the essential product attributes for software engineering. As a result, users' needs and perspectives are becoming more salient (e.g. Rumbaugh, 1994; Fairley and Thayer, 1997). However, in software engineering usability occasionally means only that software should have an appropriate user interface and adequate documentation (Sommerville, 1996). In HCI and this thesis, following John Karat's (1997) description, usability is not considered limited to the display and keyboard interface between human and machine, but rather it is considered to encompass how an artefact fits into a complex work or home environment.

Even if usability is considered chiefly in the HCI literature, it has been a major factor in the user's overall perception of quality in interactive systems (Dzida et al., 1978). In addition, software engineers do not want to have usability methods and processes separate from software engineering (Carlshamre and Rantzer, 2001). Thus, RE and HCI can be considered complementary to each other and they should be more closely associated. This thesis focuses on the early phases of product development, utilising both the human view of HCI and the practical development view of RE.

1.2 User-centred design

User-centred design is the general approach used in this thesis. The goal of user-centred design is the development of useful and usable products, although there appears to be no agreed-on definition or process for it (Karat, J. 1997, p. 36-37). However, the principles that Gould and Lewis (1985) present are generally accepted. The principles are:

1) Early focus on users and tasks
2) Empirical measurement
3) Iterative design

The first principle includes the idea of user involvement: Gould and Lewis (1985) recommend bringing the design team into direct contact with potential users, as opposed to hearing or reading about them through human intermediaries, such as marketing, sales, and users' managers. These kinds of indirect links are considered less desirable because they can filter and distort the information gathered about users (Keil and Carmel, 1995). For example, managers may not be aware how the job is really accomplished in practice.

The second principle of user-centred design implies that early in the development process, intended users should use simulations and prototypes to carry out real work, and their performance and reactions should be observed, recorded, and analysed. The third principle suggests that, when problems are found in user testing, as they will frequently be, they must be fixed. Gould and Lewis (1985) suggest a cycle of design, test, and measurement, repeated as an alternative to the waterfall models.

Gould and Lewis (1985) do not specify what 'design' means in the overall software development. User-centred design activities are recommended ideally to take place throughout the lifecycle of the product, with significant activities happening at the early stages before the user interface has been designed (Nielsen, 1993). Since Gould and Lewis's classic article, these ideas have been elaborated, and methodological approaches to usability engineering have been introduced by a number of authors such as Mantei and Teorey (1988), Nielsen (1993), Wixon and Wilson (1997), and Mayhew (1999).

Mantei and Teorey (1988) describe how the traditional and prototyping lifecycles change when human factors activities are added to form one iterative lifecycle. These changes are shown in italic in Table 1. Nielsen (1993) and Wixon and Wilson (1997) take the usability engineering point of view, and they include setting and measuring the usability goals within their lifecycle descriptions. Their descriptions concentrate on user interface design, whereas Mayhew (1999) incorporates usability engineering into general software development lifecycles.

Mantei and Teorey (1988)	**Nielsen (1993)**	**Mayhew (1999)**
Market analysis Feasibility study Requirements definition *Product acceptance analysis* *Task analysis* Global design Prototype construction *User testing and evaluation* System implementation Product testing *User testing* Update and maintenance *Product survey*	Knowing the user Competitive analysis Setting usability goals Parallel design Participatory design Coordinate design of the total interface Applying guidelines and heuristic analysis Prototyping Empirical testing Iterative design Collecting feedback from field use	Requirements analysis - User profile, task analysis, platform capabilities/constraints, general design principles -Usability goals Design/testing/development -Work re-engineering, conceptual model design, CM Mockups, Iterative CM evaluation -Screen design standards, SDS prototyping and iterative evaluation -Detailed user interface design and evaluation Installation -User feedback

Table 1. The usability engineering lifecycles.

In addition, ISO 13407 (1999) identifies four principal activities of user-centred design. These activities are:

a) understanding and specifying the context of use, including the characteristics of the intended users, the task users are to perform, and the environment in which they are to use the system,

b) specifying the user and organisational requirements in relation to the context of use description,

c) producing design solutions iteratively by using user feedback,

d) evaluating designs against requirements at all stages in the system life cycle.

ISO 13407 (1999) provides guidelines for planning the user-centred design process and user-centred design activities, but does not describe specific techniques or methods for the activities.

The benefits of user-centred design are discussed in Study I of this thesis. Mantei and Teorey (1988) introduce the topic of cost-benefit analysis of usability engineering by discussing the cost of incorporating a wide range of usability engineering activities into the development cycle. Bias and Mayhew (1994), Clare-Marie Karat (1997) and Lund (1997) provide a framework for cost-benefit analysis and discuss several excellent examples of cost-benefit analysis, demonstrating that usability activities bring value to corporations. The following are brief examples of the estimated benefits.

Increased sales. Based on "buy decision" data from usability tests and surveys, it is estimated that the new usability-engineered system will have sales that are 25%

higher in the first year compared with "product development as usual" (Karat, 1994).

Increased user productivity. In one case, the reduction in user time to complete the first three tasks from the initial to the final version was 4.67 minutes after three iterations of usability design and testing (Karat, C., 1997). The application had 22876 end users, so the working time saved was 1781 hours. The cost-benefit ratio of task analysis, development of a low-technology prototype, three iterations of usability testing, and redesign was evaluated to be 1:2. In another case, the reduction in time on task from first to final user interface was 9.6 minutes on average after a benchmark test, development of a high-technology prototype, three iterations of usability prototype testing, and redesign (Karat, C., 1997). The cost-benefit ratio of the usability work was evaluated to be 1:100.

Decreased training costs. Dray and Karat (1994) estimate that a well-designed system could decrease training costs by 35%. The project team conducted iterative usability evaluations for prototypes and moved their offices so that they were in constant contact with users and the context in which they performed their work.

Decreased user support. Microsoft announced that the number of support calls dropped dramatically as a result of usability testing and problem identification, leading to a revised design (Reed, 1992). The average time per call fell to less than ten minutes instead of the earlier 45 minutes. Similarly, the Ford Motor Company changed 90 percent of their accounting software for their small car dealerships as a result of usability testing, and they were able to drop the help-line calls to zero (Kitsuse, 1991). Earlier, it took the car dealers three help-line calls merely to get started.

1.3 User involvement

This thesis is focused on the first principle of user-centred design: early focus on users and tasks. It is generally agreed that usability is achieved through the involvement of potential users in system design (Karat, J., 1997; Bekker and Long, 2000; Wilson et al., 1997). ISO 13407 (1999) "Human-centred design processes for interactive systems" similarly recommends the active involvement of users. It is suggested in ISO 13407 that user involvement is essential in order that the relevant user and task requirements can be identified for inclusion in the system specification.

It is apparent that direct contact with users is crucial in order to understand the various contexts of use. Moreover, in theory, user involvement is most efficient and influential in the early stages of system development as the cost involved in making changes increases during system development (cf. Ehrlich and Rohn, 1994; Noyes et al., 1996).

However, a clear definition of user involvement is lacking. The term has been used synonymously with "focus on users" (Wilson et al., 1997), "contacting with system users" (Grudin, 1991a), "consulting end-users" (Noyes et al., 1996), and "participation of users" (Heinbokel et al, 1996). Thus, the role of users has been considered as being somewhere on the continuum from a passive informant to an active participant in system development. User involvement is a general term describing direct contact with users and covering several approaches.

In addition, as John Karat (1997) points out, it is not apparent what the best techniques of user participation are. New approaches, such as contextual inquiry and ethnography, focus on the early phases of development and appear attractive, but they are generally thought to be time consuming.

1.4 Requirements engineering

Requirements engineering constitutes the earliest phase of the software development life cycle. Regnell (1999) states that a major motive for spending time and effort on requirements engineering comes from the objective of developing the software correctly from the beginning, instead of patching it at the end. Davis (1993) also compiled the results of three empirical studies, indicating that it may be up to 200 times more expensive to detect and repair errors in the maintenance stage, compared to detecting and repairing them during the requirements engineering phase. He further argues that the potential impact of errors in requirements is substantial:

- The resulting software may not satisfy users' real needs.
- Multiple interpretations of requirements may cause disagreements between customers and developers, wasting time and money and perhaps resulting in lawsuits.
- It may be impossible to test thoroughly that the software meets its intended requirements.
- Both time and money may be wasted building the wrong system.

To have a successful requirements definition, it is necessary that the entire requirements engineering process is carried through. Kotonya and Sommerville (1998) consider the activities of requirements engineering process to include requirements elicitation, requirements analysis and negotiation, requirements documentation, and requirements validation. In practice, the activities are interleaved and there is a great deal of iteration and feedback from one activity to another. The activities are described below.

Requirements elicitation

Thayer and Thayer (1997) consider requirements elicitation to be the process through which the customers (buyers and/or users) and the developer of a software system discover, review, articulate, and understand the users' needs and the constraints on the software and the development activity. In this definition, the customers have a very active role. However, in product development, there may be a large number of potential users, and it may not be possible to specify in advance the exact customers and users. Thus, customers may not always be available for requirements elicitation and the responsibility of understanding customer needs belongs to designers.

Kotonya and Sommerville (1998) suggest that the requirements elicitation involves understanding the application domain, the specific problem to be solved, the organisational needs and constraints, and the specific facilities needed by system stakeholders. System stakeholders are those people who are affected in some way by the system, for example users, customers, and engineers responsible for system development and maintenance. Thus, several stakeholders are involved in requirements elicitation. Users are some of the most important stakeholders as they use the ultimate product. In addition, users are frequently experts on the application domain, the specific problem to be solved and how the organisation really functions.

In the literature, a wide range of methods are described, for example, introspection, interviews, questionnaires, scenario analysis, and protocol, conversation, interaction, and discourse analyses (Goguen and Linde, 1993; Maiden and Rugg, 1996). Furthermore, there are group methods, such as requirements workshops, where the general benefit is to acquire an agreement between the stakeholders and the development team (Leffingwell and Widrig, 2000). In addition, ethnographic methods have been adapted from sociology (Hughes et al., 1995) and card sorting and laddering techniques from the discipline of knowledge engineering (Maiden and Rugg, 1996). Finally, techniques of storyboarding, prototyping, and use case analysis can be used to uncover missing requirements (Davis, 1993; Rumbaugh, 1994; Leffingwell and Widrig, 2000).

Maiden and Rugg (1996) argue that more than one acquisition method is needed to capture the full range of complex requirements for most complex software-intensive systems. For example, Hughes et al. (1995) suggest that incorporation of ethnography emerges from a previous insufficient attention to the social context of work. Generally, the term "elicitation" is preferred to "capture", to avoid the idea that the requirements are explicit and easily collectable (Karlsson, 1996; Kotonya and Sommerville, 1998).

In HCI, the most common approach to requirements elicitation is to perform field studies, whereby qualitative methods are used to study users and their activities in their own environment (cf. Bly, 1997; Wixon, 1995). Thus, the goal is to elicit user needs instead of ready requirements. The reason for the relevance of field studies is that communicating with users in a laboratory lacks the impact of the context and possible uses of technology (Blomberg et al., 1993). The users' normal environments additionally help them remember details of their behaviour, and it is possible to observe them acting in their usual circumstances. In general, field studies ensure the accuracy of the work models used in early stages of development as they are based on actual work episodes and products (Wood, 1996).

Hackos and Redish (1998) and Wixon and Ramey (1996) describe an extensive range of field study methods. The basic methods are observing and interviewing, but there are several complementary methods, such as role playing, cued recall with videotapes, thinking aloud, story telling, and artefact walk-through. Furthermore, tasks can be represented and validated via scenarios (Carroll, 1995), use cases (e.g., Constantine, 1995), storyboards (Madsen and Aiken, 1993), or prototypes (e.g. Rudd, Stern and Isensee, 1996).

In practice, requirements engineers are familiar with such elicitation methods as observation, interviewing, and using documentation (Goguen and Linde, 1993). On the other hand, based on a large survey of software development organisations, Potts (1993, p. 21) states that, in product development projects, requirements were generally invented, not elicited. These projects had a potential market for their product, but they had difficulties in identifying the customer, and deliberately intermingled requirements and design decisions.

Requirements analysis and negotiation

Kotonya and Sommerville (1998) claim that requirements elicitation and requirements analysis are closely linked and interleaved processes. As requirements are discovered during the elicitation process, some analysis is inevitably carried out. Subsequently, the requirements are analysed in detail and requirements engineers and stakeholders negotiate to agree on the definition of the requirements to be included in the requirements document. Imaz and Benyon (1999) state that analysis is moving from informal descriptions of user data to more formal design representations, and there is no agreement on how this should be confidently and consistently accomplished.

Requirements documentation

For designers, requirements must be expressed and documented in a structured way (Rumbaugh, 1994). Frequently, requirements are divided into "user requirements" and "technical requirements" (e.g. Rombach, 1990; Forsberg and Mooz, 1997; Stevens, 1997). The terms are seldom defined, but the division suggests that there are requirements elicited from users, and requirements that originate from technical constraints and development.

Rumbaugh (1994) suggests that the requirements must be understandable to users, as they must verify them. Thus, user requirements should be written from the user point of view and using their language.

In addition, requirements documentation can be considered from the human problem solving point of view, as in Leveson (2000), who states that specification is aimed at the support of human problem solving. The specifications help the designer, builder, tester, debugger, and maintainer to understand the system well enough to create a physical form or to find problems in or change the physical form. Leveson identifies factors influencing human problem-solving ability from the perspective of cognitive psychology. For example, human problem-solving performance can be improved by providing representations that reduce the problem solver's memory load and that display the critical attributes needed to solve the problem in a perceptually salient way. An incomplete problem representation can actually lead to a worse performance than having no representation at all (Smith, 1989).

Thus, Leveson (2000) argues that a way to cope with complex systems is to structure the situation such that the observer can transfer the problem being solved to a higher level of abstraction. Here, she suggests two ways that humans cope with complexity: top-down reasoning and stratified hierarchies. Thus, models of complex systems can be expressed in terms of a hierarchy of levels of organisation, each more complex than the one below it. Further, each level of software specifications can be thought of as providing *what* information, while the next lower level describes *how*. In addition, Leveson seeks *why* information, as design errors may result when we either speculate incorrectly about higher-level intent or omit it from our decision-making process.

In summary, ideally we need three kinds of hierarchical documentation: *why* and *what* information represented from a user point of view, and *how* information from a technical point of view.

Requirements validation

Kotonya and Sommerville (1998) suggest that requirements validation is intended to detect problems in the requirements document before it is used as a basis for system development. In addition, user needs are always interpreted in user requirements and, as Rumbaugh (1994) points out, users must verify the requirements. This kind of user validation is consistent with the iterative design principle of user-centred design.

1.5 User needs and requirements

In this thesis, the focus is on requirements elicitation activities, where users' needs are discovered. However, the entire requirements engineering cycle needs to be considered so that user needs can be utilised in product development. It is impossible to meet all user needs; there are so many needs and some of them conflict with each other. User needs must be discovered, but also analysed, prioritised, described, and finally documented in user requirements.

User needs are considered to be the first and most informal data to be used in requirements definition. The requirements engineering cycle can be seen as the process by which user needs are converted to user requirements. Thus, user needs form the basis for the rest of the product development; they represent problems that hinder users in achieving their goals, or they represent opportunities to improve the likelihood of users' achieving their goals. An important factor affecting user needs is the context of use, which includes users, tasks, equipment, and the physical and social environments in which a product is used (ISO 13407: 1999). The quality of the product is not separate from the users' world, but depends on who uses it and in which context it is used.

A user requirement can be defined as any function, constraint, or other property that is required in order to satisfy the user's needs (cf. Abbott, 1986). Therefore, user requirements describe how a future product can help users achieve their goals effectively, efficiently, and with satisfaction in their context of use (cf. the definition of usability in ISO 13407: 1999). User requirements include information about those particular user needs that are selected and that are satisfied by the future product. Technical requirements describe how the product will be implemented.

1.6 The challenge of requirements elicitation

A basic question in requirements engineering is how to find out what users really need (Goguen and Linde, 1993). However, the elicitation of requirements information from users is frequently problematic, as discussed by Palmer (1987). The problem can be attributed to poor communication between users and designers. Misinterpretations are evident between them as both sides have different backgrounds, knowledge, vocabulary, and goals. The most frequently mentioned requirements elicitation problems and possible solutions are discussed below and summarised in Table 2.

Users do not know what they want, or they cannot articulate it

Frequently, users do not know what they really need, they cannot articulate the needs, and they are unaware of the technical possibilities and the costs of their requests. As Palmer (1987) states, users may not understand the tasks they face sufficiently well to describe them in a manner understandable to the group attempting to determine requirements.

A difficulty is that part of the users' knowledge has become tacit through automation (Mitchell and Chi, 1985; Wood, 1997). In well-learned tasks, much of the relevant knowledge is no longer consciously available for the person and non-verbal skills and everyday self-evidences are difficult to articulate. In addition, the social nature of work can be vital to successful operations, yet at the same time appears to be so trivial that it is difficult to uncover (Viller and Sommerville, 1999a).

Hudlicka (1996) categorises the elicitation methods into direct and indirect methods. She considers direct methods to be those which rely on the subjects' abilities to articulate their knowledge in response to direct questions. According to her, the indirect elicitation techniques imported from experimental psychology to knowledge engineering assume that relevant knowledge is frequently not easily accessible to conscious thought. Therefore, these methods attempt to overcome this limitation by accessing pieces of the internal knowledge structures indirectly. This is achieved, for example, by asking simple questions about similarities and differences between important domain entities, or by asking the subjects to rate the similarity of two items on a numerical scale. Hudlicka (1999) herself claims that if little knowledge about the domain is available, it is advisable to begin with more open interviews and use direct techniques. Indirect techniques can be used to refine elements of the domain.

The distinction between indirect and direct methods suggests that we need different methods for different purposes and that even implicit knowledge can be elicited using appropriate methods. Goguen and Linde (1993) suggest that the strengths of the elicitation methods seem to be somewhat complementary, so that combinations of the various methods can be usefully applied to particular problems. Karlsson (1996) shows by her case studies that different elicitation methods complement each other, and that generally a more complete picture is provided by field studies.

As suggested earlier, field studies involve the study of users and their activities in their own environment. Field methods such as those used by ethnographers focus on understanding and uncovering the target context which is based on unspoken knowledge (e.g. Ford and Wood, 1996). Thus, users do not need explicitly to articulate their needs. By studying users, their activities and environment, it is possible to understand the underlying problems and possibilities.

There are too many users to study

As Potts (1993) states, one fundamental problem for a product development project is identifying the customer. In addition, it may be felt that practically all people are potential users, and thus there are too many users to study. Furthermore, as Hackos and Redish (1998, p. 111) point out, designers make several assumptions about users that affect their decisions, and such assumptions may be mistaken if the designers have not met the users.

It is not necessary to meet all the users or potential users if the various kinds of users are identified, and representative users are sampled from all essential groups. Hackos and Redish (1998) suggest considering users in terms of characteristics such as job or task type, experience, frequency and level of use of the product, environment, culture, and motivation. A team including, for example, salespersons, marketing professionals, and trainers can then brainstorm a preliminary list of users.

A new product will provide a new way of carrying out the existing tasks

The focus on users' current practices and processes may seem odd when a new product is planned to provide a completely new way of carrying out existing tasks and processes. However, the new product is not used in vacuum; users have needs relating to it depending on the context of use. Thus, the future context of use must be identified. It is accordant with users' needs to let them use their skills, retain the advantages of current processes, and fix problems.

Users require a specific feature or technical solution

Occasionally, it is difficult to focus on users' needs, as the users request a specific feature or technical solution. However, users are not designers and they may believe that a specific feature solves their problems, even if it is not an optimal solution. They may be unaware of other technical possibilities and the costs of their request. Thus, it is a designer's task to discover the underlying user needs and evaluate the alternative solutions.

Problem	Solution
Users do not know what they want, or they cannot articulate it.	Alternative elicitation techniques such as field studies provide a more complete picture without the need for users to articulate their needs. Users are recognised as experts in their tasks; the focus is on their goals, present processes and context of use.
There are too many users to study.	Identify the various kinds of users and sample representative users from all essential groups.
A new product will provide a new way of carrying out the existing tasks.	In order to understand the user needs: • The pros and cons of the present way of achieving the user goals must be identified • The future context of use must be identified • Users should be allowed to use their skills, the advantages of the current processes should be saved and the problems fixed.
Users request a specific feature or technical solution.	The underlying user needs should be discovered. The users may believe this feature solves their problems but it may not be an optimal solution.

Table 2. The problems and solutions of requirements elicitation.

1.7 Research problems and goals

The aim of this work is to study early user involvement in real product development contexts. In the literature, user involvement is considered essential for understanding user needs and achieving usability. In particular, early user involvement is identified as potentially the most efficient.

In practice, it is unclear how early user involvement could be successfully implemented in real product development contexts. It is typical that usability testing is the only usability activity that is used in software companies (Darnell and Halgren, 2001). However, usability tests have limitations, one of which is that evaluation techniques (and usability tests) "react" to an existing design and thus are aimed at "improving" rather than "creating" (Wixon et al., 1994). The success of a usability test depends on how representative the test task and environment are. It is not possible to facilitate users' tasks in their own environment without identifying how and for what reason the users are going to use the product. In addition, usability tests are often conducted so late in the development cycle that major changes must wait until later revisions before they can be implemented (See Page, 1996).

Early user involvement plays a role in understanding user needs, including context of use, in the early stages of product development. Field methods can be recognised as most promising in understanding user needs as users are studied in their own environment. However, field studies are generally considered time consuming and effort intensive (Bly, 1997) and thus they add to product development costs.

The fundamental question of this work is whether user involvement is useful in requirements elicitation even before a prototype of the system exists. As suggested earlier, it is unclear how cost-effective it is to involve users before a prototype exists, and how users should be involved in practice. The work concerns field methods, and gathering user needs and requirements in the early stages of product development. Subsequent development phases and other user-centred activities such as setting usability goals and usability testing have been excluded.

The research questions are:

1. What are the benefits and challenges of user involvement?
Here the focus is particularly on the early phases of product development.

2. How can early user involvement be applied in product development projects?

3. How can early user involvement be introduced to product development organisations?

The hypotheses were formed after the first research question was answered through a literature review. The principal hypothesis is that early user involvement has benefits for product development on condition that it is practical and cost-efficient enough. This hypothesis is tested by developing a practical approach to user involvement and evaluating it in real product development contexts.

Thus more detailed goals of this work are as follows:

- Clarify the nature of user involvement and describe its challenges and benefits based on the literature (I).

- Synthesise a practical approach to early user involvement which can be utilised in real product development contexts. The aim is to face the identified challenges and to achieve the expected benefits (II, IV).

- Provide empirical evidence of the benefits and specify costs of the approach in the real product development contexts (II, III, IV, V).

- Investigate ways of introducing the approach in product development organisations (IV, V).

Figure 1 illustrates the primary pieces of this work.

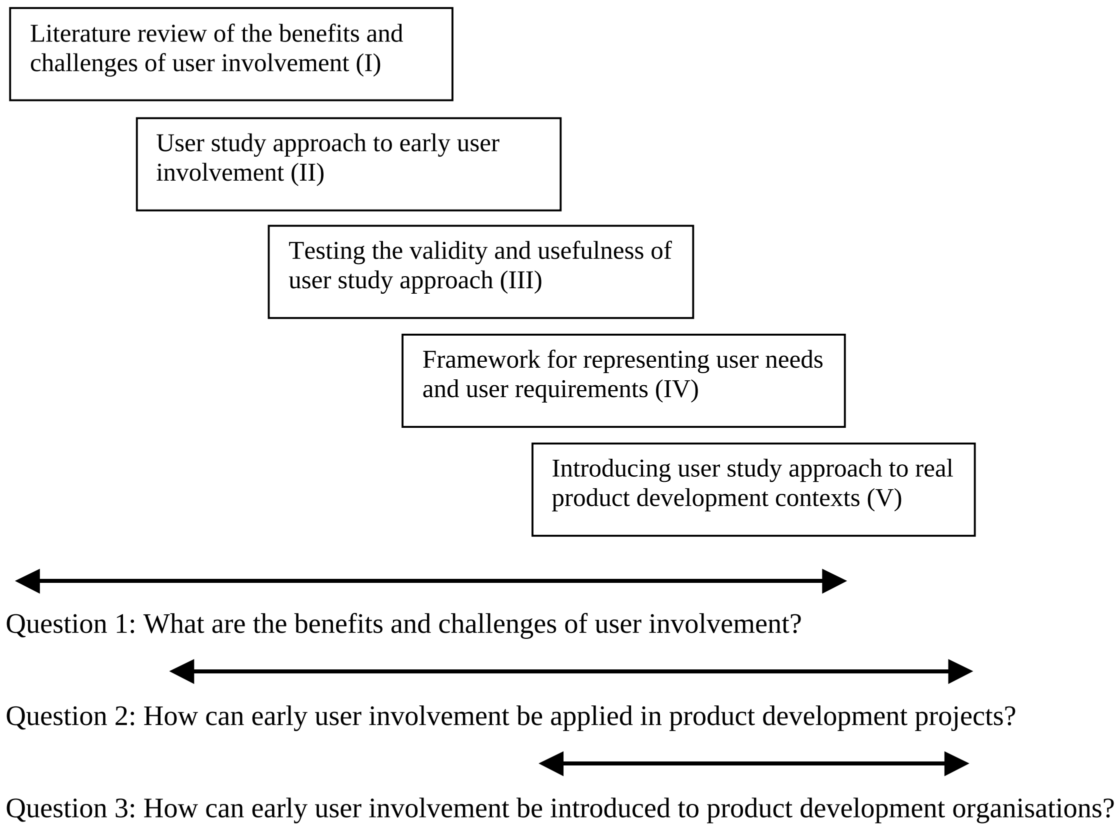

Figure 1. The relation of the research work pieces and the research questions. The numerals refer to the publications in which each piece is dealt with.

2. Research setting

This work has various goals, but the general aim is to explain user involvement and synthesise an approach for utilising user involvement in real product development contexts. The product development is characterised by tight schedules and limited resources, and it is thus challenging to introduce new practices to product development companies. In addition, the proposed approach must fit into the limits of real circumstances. Therefore, this work was conducted within real product development companies.

This chapter describes the research projects in which the research work was completed and the research approach and methods used. It additionally gives an overview of the case studies completed, concerns relating to evaluating the user study approach, and the product development contexts of the case companies.

2.1 Research projects

All the studies were based on real product development cases in companies in Finland. Studies II and III were completed in Usability of Smart Products in Information Society-research project during years 1998-1999. The final two studies (IV and V) were based on work with the three industrial partners as a part of the QURE project during 1999-2001.

The aim of the Usability of Smart products in Information Society project was to consider usability issues of small interactive devices. It was a collaborative effort between the Usability Group at the Department of Computer Science at HUT and the Department of Product and Strategic Design at University of Art and Design Helsinki. All six researchers were focused on their own topics. The project was supported by the Academy of Finland, and it had no industrial financiers or partners. The first two product development cases were obtained for studies II and III from outside companies.

The QURE project (Quality through Requirements) is a three-year research project ongoing at the Helsinki University of Technology. The aim of the project is both to help the industrial partners in Finland to improve their requirements engineering practices and to do research work on human-computer interaction and requirements engineering. Studies IV and V were performed as part of normal project work in the QURE. This work was multidisciplinary and collaborative between the project members and the industrial participants.

2.2 Research approach and methods

The case-study research strategy accords with research work in real product development contexts. Yin (1994) defines a case study to be an empirical inquiry within its real-life context, particularly when the boundaries between phenomena and context are not clearly evident. This work consists of four case studies, in which multiple sources of evidence were used, as recommended by Yin (1994). Evidence was gathered by using documentation, participant-observation, interviews, and questionnaires (Table 3). In addition, Study III was complemented with gathering empirical data from an experiment.

Study	**Problem**	**Data gathering methods**
I User Involvement: A Review of the Benefits and Challenges	What are the benefits and challenges of user involvement in product development?	Literature review
II Studying Users for Developing Usable and Useful Products	How can early user involvement be applied in product development?	Participant-observation, interview
III How Effective Are User Studies?	What are the benefits and costs of the proposed approach to early user involvement compared to usability testing?	Documentation, experiment (replicated product design), interview
IV Bridging the Gap between User Needs and User Requirements	How can user needs be represented and translated into user requirements in industrial product development cases?	Participant-observation
V Introducing User Needs Gathering to Product Development: Increasing Innovation and Customer Satisfaction	How can the proposed approach be introduced to product development contexts?	Participant-observation, questionnaire, interview

Table 3. The research problems and data in the articles.

The research approach of the case studies relates to action research as defined by Avison et al. (1999), who suggest that action research is an iterative process involving researchers and practitioners acting together on a particular cycle of activities, including problem diagnosis, action intervention, and reflective learning. In this work, problem diagnosis was conducted through reviewing the literature and interviewing product developers. Developers were interviewed in studies II and III, and in the early part of studies IV and V, developers were interviewed and the initial state of user and customer practices of the case studies were assessed. The results of the latter are described at the end of Section 2.5.

This work also represents the engineering research method. Glass (1994) describes an engineering research method based on the Dagstuhl Workshop (Tichy, 1993) so that it is observing existing solutions, proposing better solutions, building and developing, measuring and analysing, iterating until no further improvement is possible. In this work, improvement proposals on existing techniques are implemented, evaluated, and redesigned. First, a user study approach is proposed to overcome existing challenges (II), it is evaluated (II, III, IV, V), and developed further to better fit with software engineering practices (IV).

2.3 Overview of the case studies

Table 3 summarises the research problems described in the publications. In the beginning of the work, a literature review was conducted to clarify the nature of user involvement and to distinguish different approaches of the user involvement (Study I). Then a practical user study approach was formed unifying the views and methods of different user involvement approaches (Study II). This framework was evaluated in four case studies (Studies II, III, IV, and V). As a result of these experiences, a way of presenting user needs and requirements was elaborated further and evaluated in one case study (Study IV).

In the first two case studies, the researcher performed the user studies. The practitioners offered the real product development problem, and they were subsequently interviewed to evaluate the process and the results. In the final two case studies, the goal was to understand how the framework can be introduced to real product development contexts and how effective it is in those contexts. The researcher had a role in guiding the user studies and developers actually performed them. The researcher's role was that of an expert or a consultant who provided information, instructions, training, and support for the practitioners.

2.4 Evaluating the effects of early user involvement

The main theme of the work was evaluating the effects of early user involvement in terms of benefits and costs. As earlier described, this was achieved by forming a practical user study approach and evaluating it. The criteria for evaluating the user study approach were formulated after the literature review, in which influential factors of early user involvement were identified.

In Study I, three intermediate factors were identified influencing how early user involvement can contribute to product quality: product development performance, better requirements, and usability. The factors can be divided according to the attributes shown in Table 4. Early user involvement is shown to have direct or indirect effects on many of the product development attributes; examples of the literature are referred in Table 4. It is more difficult to find evidence of the effects on requirements quality or the usability of the final product. Their effects have generally been shown indirectly by describing the quality of information gained from users and the value of insights gained from field studies.

The factors and their attributes set requirements for a user study approach, and they can be used in evaluating the effects of the approach in a real product development context. Many of the attributes are difficult and time consuming to measure in real product development contexts. However, we can evaluate what kind of effects the user study approach has on the attributes of product development. For example, user involvement can have both positive and negative effects on product development time, but both kinds of effects are difficult to evaluate in case studies. However, we know that the time required to involve users is frequently a crucial factor in projects (Wilson et al., 1997), and we can measure the time spent on learning and using the user study approach.

Influential factors	**Attributes**	**Explanation of the attributes**
Product development performance	Product development time	Time needed for the project (Chatzoglou and Macaulay, 1996).
	Costs of product development	Cost of the project (Chatzoglou and Macaulay, 1996).
	Number of iterations	Repetitions of the same normalised activities to create preliminary versions of the final product (Chatzoglou and Macaulay, 1996).
	Quality of team interaction	Democracy, openness to criticism, competition and dominance in a team (Heinbokel et al., 1996).
	The team's domain knowledge	The spread of application domain knowledge among team members (Hofmann and Lehner, 2001).
	Relations among customers, users, and developers	Mutual understanding and work relationships among all the stakeholders (Muller and Carr, 1996), customers' perception of product development team (Rowley, 1996), the likelihood that developers' decisions match the needs of users (Rowley, 1996).
Quality of requirements	Completeness	Everything that the software is supposed to do is included in the requirements specification (Davis et al., 1997).
	Correctness	Every requirement in the requirements specification contributes to the satisfaction of a need (Davis et al, 1997).
	Understandability	All classes of requirement specification readers can easily comprehend the meaning of the requirements with minimum explanation (Davis et al, 1997).
	Verifiability	There exist finite, cost effective techniques that can be used to verify that every requirement stated therein is satisfied by the system as built (Davis et al, 1997).
Usability	Ease of use	The system is easy to learn, efficient to use, memorable, and it has a low error rate (Nielsen, 1993).
	Fit with user needs	The depth and breadth of information obtained about customers' task and needs (Juhl, 1996), understanding of the problems that users face (Blomberg et al., 1996).
	User satisfaction	Users are subjectively satisfied when using the system (Nielsen, 1993).

Table 4. The factors influencing early user involvement.

The attributes raise partly overlapping concerns for evaluating the user study approach (Table 5). These concerns were observed in case studies using the data gathering methods described in Section 2.2.

Effecting factors	Attributes	Concerns in the case studies
Product development performance	Product development time	• How much time is spent in learning a user study approach? • How much time is spent in a user study approach?
	Costs of product development	• Is it possible to use the approach in real product development contexts? Is the approach acceptably cost-efficient?
	Number of iterations	• Does the approach offer useful information for usability specialists and developers? Is the information available early enough?
	Quality of team interaction	• Is the elicited information communicated effectively to developers? • Do the user study results reduce the number of disagreements among members of the development team?
	The team's domain knowledge	• Does the approach offer useful information for usability specialists and developers?
	Relations between customers, users and developers	• How do developers and salespersons react to user studies and the results of the user studies? • How do users and customers react to user studies and the results of the user studies?
Quality of requirements	Completeness	• Does the approach offer useful information for usability specialists and developers? • Does the approach provide a comprehensive view of users and their needs?
	Correctness	• Is the elicited information communicated accurately to developers?
	Understandability	• Is the elicited information understandable to developers?
	Verifiability	• Does the user study approach offer information for improving usability testing?
Usability	Ease of use	• Does the user study offer information that helps developers to produce more easy-to-use products?
	Fit with user needs	• Does the user study offer information that helps developers to produce products that meet user needs?
	User satisfaction	• Does the user study offer information that helps developers to produce products that are more satisfying to users?

Table 5. Concerns relating to evaluating the user study approach in case studies.

2.5 Product development context

In addition to knowing how the case studies were completed, one must consider the development contexts and companies in which they were conducted. Grudin (1991a) describes three principal contexts in which interactive software is developed and various conditions that affect the development process. The contexts are contract development, product development for a large market of users, and in-house development of a system for a specific set of users. The different circumstances of each context greatly affect the approach and tools that are likely to be used. This is particularly true in the case of user involvement. For example, in product development, contrary to contract development, the development team is known early, but the users are not effectively known until the product sells. With in-house development, both groups are identifiable from the outset.

All the case companies in this thesis were developing products and the scope of the work was in product development. Grudin (1991a) states that the focus of product development is on the user interface. He claims that usability expectations increase in more mature software markets, and that product development companies are entering the phase in which users' needs are more important factors than software constraints in development. In addition, Grudin (1991b) provides a detailed description of the organisational obstacles to direct contact between developers and users in large product development organisations. Such obstacles include challenges in motivating the developers, identifying appropriate users, obtaining access to users, motivating the users, and in deriving benefits from user contact when established. These findings rely on an earlier survey and on interviews with over 200 interface designers from several product development companies.

The product development projects are characterised by scarce personnel, tight schedules, and uncertainty. Products may have a large number of potential users, who cannot be specified in advance. Under such less-than-perfect circumstances, designers frequently find involving users too time consuming to be practical.

In addition, there are other factors influencing software development, for example the size of the development company or organisation, organisational structures, project size, and competitive situation. The size of the company may affect factors such as how many extra resources exist for supporting product development work. Large companies may have more resources and more elaborated divisions of labour compared to start-up or small product development companies (Grudin, 1991a). Furthermore, large companies are more likely to have usability groups or usability experts. Organisational structures and project sizes affect the type and quality of communication between project members and others in the organisation. A tight, competitive situation generally creates pressures for shortening development time.

In addition, in product development, the product type may change the development conditions. For example, Ulrich and Eppinger (2000) suggest that there are several variants to the development process corresponding to product and development types. The product development types are market pull, technology push, platform products, process intensive, and customised. Market pull product development begins with a market opportunity whereas technology push product development assumes a given technology. Platform products include an assumption of a technology platform while process intensive products are highly constrained by the production process. Customised products are slight variations of existing configurations. In addition, one can distinguish software development process and products, which may be embedded, interactive, or consumer products.

In this thesis, the first two case studies were in consumer product projects in one medium-size and one large organisation. A major factor in determining competitiveness of consumer products is frequently how quickly a product is delivered, and the frequent releases of consumer products create the time pressure (Blackburn et al., 2000). On the other hand, the competition in the marketplace provides motivation to improve usability.

The last two case studies were conducted in product development units of medium-size or large companies. The participating companies were the KONE Corporation, Tekla, and Vaisala. Their products are embedded and/or interactive systems, and they are either new versions of older products or totally new products (Table 6). The sizes of the case companies in terms of number of employees and the products types involved in case studies are shown in Table 7.

	Study II Teamware	**Study III X**	**Study IV**			**Study V KONE**
			KONE	**Tekla**	**Vaisala**	
Consumer product	X	X				
Embedded product	X	X	X		X	X
New version		X	X			X
New	X			X	X	

Table 6. Case companies and product types.

	Number of employees	**Products**
Teamware	220	PDA-device
X	23 000	Portable communications device
Kone	23 000	Elevators and escalators
Tekla	500	Information management system for building
Vaisala	1100	Weather measurement instrument

Table 7. The size of the case companies and the products involved in case studies.

In the early phases of the QURE project, the current requirements engineering practices of the industrial partners were assessed. In addition, the initial state of user and customer practices were examined in more detail by interviewing in three of the industrial partner companies.

Three designers representing different projects were selected from each of the companies. We found that the projects were technology driven. The companies had considerable experience of developing products, and the practitioners seemed to rely on this earlier experience, using indirect links such as sales persons to users and customers. Five out of nine projects had no direct contact with users, or contact was not established until the late prototype phase. Consequently, in certain projects the user and customer requirements were not elicited or documented. Even if a project had direct contacts with users, gathering user information was occasionally passive, methods were informal, and the gathered information was not documented in two of the companies. Six interviewees reported that they would need more information about customers, and three of them similarly mentioned users. Some of them felt that more information and more instruction were needed about how to gather, describe, and classify user and customer requirements.

Two of the companies did not have usability experts, groups, or special usability resources. The companies did not have defined practices for usability in place, even if several kinds of usability engineering practices were used in some projects. It can be seen that user-centred practices had not generally been adopted. There were no plans to assess how the human-centred activities fit into the overall product development processes and generally usability goals were not set. Neither the current processes followed by users nor their use contexts were generally considered in projects. Formal usability testing was not used. Rather, prototypes were shown to customers, or feedback was obtained when a customer had tried to use a pilot product. Finally, the customer feedback from final products was gathered efficiently in one of three projects. In addition, one of the companies had set up a help desk to gather customer feedback.

3. Results

This chapter summarises the results of the presented work. The results are grouped in relation to the research goals presented in Chapter 1.

3.1 Study I: The benefits and challenges of user involvement

The first part of this work was a literature review on the benefits and challenges of user involvement. It was found that user involvement is loosely used in the literature to mean "direct contact with users" and covers several approaches.

	User-centred Design	**Participatory Design**	**Ethnography**	**Contextual Design**
Emphasis	Usability	Democratic participation	Social aspects of work	Context of work
Typical methods	Task analysis, prototyping, usability evaluations	Workshops, prototyping	Observation, video-analysis	Contextual inquiry[1], prototyping

Table 8. User involvement approaches.

The four most common approaches were identified: user-centred design, participatory design, ethnography, and contextual design. In addition, task analysis can involve users (see Hackos and Redish, 1998; Johnson, 1989; Diaper, 1989). The roots and methods of the approaches are closely linked, but they all have different emphasis and a different rationale for involving users (Table 8).

The relevant research literature was reviewed in order to understand the benefits and challenges of user involvement. Three streams of research, field studies, qualitative research, and quantitative research, were reviewed. The field studies stream included case studies, which involve the use of qualitative methods to study users and their activities in their own environment (cf. Bly, 1997; Wixon, 1995). The user involvement approaches covered by field studies were ethnography, participatory design, and contextual design. The qualitative research stream included research work in which the goal was to directly assess the relationship between the costs and benefits of user involvement. This research approach concerns chiefly the participatory approach to user involvement. The quantitative research stream was focused on the effects of chiefly participatory and user-centred design on quantitative aspects of system success.

---

[1] Contextual inquiry is a field interviewing method, which combines observing and interviewing (Beyer and Holzblatt, 1998).

The three streams of research reviewed seem to have similar results. User involvement is clearly useful and it has positive effects on both system success and user satisfaction. The streams of research reveal some evidence relating to Damodaran's (1996) expected benefits of user involvement. Table 9 shows the expected benefits, which were supported by the different research streams.

Expected benefits	**Research streams**		
	Field studies	Qualitative research	Quantitative research
More accurate user requirements	X	X	X
Avoiding costly system features that the user did not want or cannot use	X	X	X
Improved levels of acceptance of the system	X	X	X
Greater understanding of the system by the user			
Increased participation in decision-making in the organisation		X	

Table 9. Evidence offered by the three streams of research supporting the expected benefits of user involvement.

In particular, the benefits of usability engineering, including prototyping and iterative usability evaluation, are clearly demonstrated. However, the focus of usability engineering is on usability testing and reacting to an existing design in Wixon's et al. (1994) words. As the success of usability testing depends on how representative the test tasks and environment are, user needs may be ignored. Therefore, the benefits of early user involvement before a prototype exists are not well established in the usability engineering literature.

A number of factors influences the success of early user involvement. Two principal categories are the development context, where user involvement happens, and the user involvement approach, or how it happens. The development context includes several variables such as system type and the number of users and customers involved (Section 2.5). In addition, McKeen and Guimares (1997) found that users need to be far more involved in cases of high task and/or system complexity.

In addition, the effects of early user involvement are complicated (Figure 2). The link between early user involvement and user and customer satisfaction is evident in the literature. Many positive responses from both customers and users are reported. Foster and Franz (1999) found a positive correlation between user participation and system acceptance and Baroudi et al. (1986) found a correlation between user involvement and system usage. The involvement seems to produce a positive outcome for users and customers, as described in Figure 2. However, most of the effects may come through intermediate factors such as requirements quality.

The results of field studies and qualitative research suggest that designers feel that they get more accurate user requirements, and they better understand user needs and usability requirements by involving users. Good's (1992) case study provides initial evidence that the understanding of the users' world can lead to more innovations. However, the links between early user involvement and better requirements, or those between usability and better system quality are not demonstrated empirically. It is difficult to determine the exact means and the extent to which user involvement influences system development as there are many intervening variables and the time frames are relatively long.

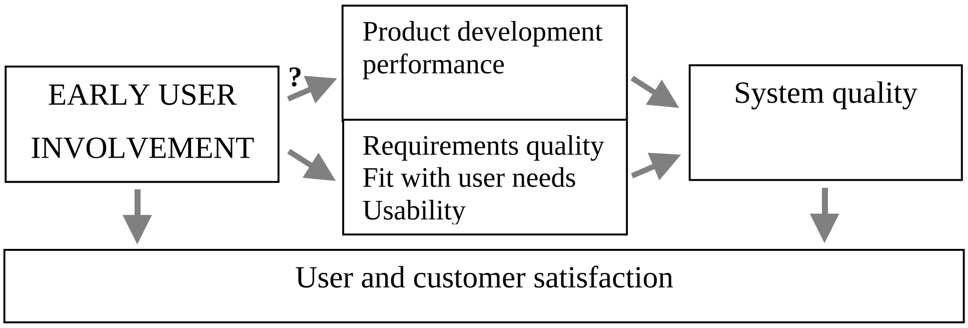

Figure 2. The effects of early user involvement.

Early user involvement additionally affects the performance of the product development team. Quantitative research indirectly shows the benefits of early user involvement. For example, Chatzoglou and Macaulay (1996) show that users, as the main source of information, decreased the number of iterations needed and thus, in most cases, lowered the elapsed time and cost of project development. Poltrock and Grudin (1994) also found that designers viewed marketing as ineffective in obtaining the information needed in order to define their product requirements, and that they were frequently frustrated by the difficulty of deciding what to do without the relevant information from users.

However, if there are communication problems between users and developers, user involvement may actually increase product development time and cost. Heinbokel et al. (1996) and Wilson et al. (1996) report that when users participate in the design project, problems arise when they demand changes in a late stage of development and designers must resolve conflicts between user groups. In addition, several challenges in field study techniques are reported, for example how to spend less time in using them, how to analyse a large amounts of data, and how to compare subjective data across users.

Therefore, even if early user involvement seems to have positive effects on requirements quality and user and customer satisfaction, it also has costs on product development. As the product development speed and productivity have become critical factors in competitiveness (Blackburn et al., 2000), the significance of these costs is noteworthy. The effects of user involvement on product development performance are unclear. It appears that the function of early user involvement is valuable for providing user need information and user and customer satisfaction. Thus, the costs can represent worthwhile investments that pay back in overall development. On the other hand, early user involvement can be a time consuming process, and field methods are not generally used in product development. In order to introduce early user involvement in real product development context, current field methods must be developed further to be cost-effective and easy to learn.

Comparing the approaches of user involvement

The order of superiority of user involvement approaches is difficult to assess. Most of the negative effects are studied and reported when a participatory approach is used. For example, Wilson et al. (1996, 1997) report difficulties in communicating between users and developers and conclude that, ideally, all stakeholders should be motivated, and that users should be educated about the entire design process. Heinbokel et al. (1996) even argue that user participation disturbs the process of software development. In their study, the projects with user participation had to deal with several problems related to developer-user relations that were not present in projects without user participation. For example, users proposed new ideas and demanded changes in a later stage of development. In Heinbokel's et al. (1996) study, projects with high user participation showed lower overall success, fewer innovations, and a lower degree of flexibility as evaluated by the team leaders and users' representatives. Characteristic of these projects was that the participation was informal as no specific methodology was mentioned.

User participation, alone, does not seem sufficient for success; developers need techniques for understanding users and their needs. Designers should take an active role in user involvement. Users are experts in their own field, but they do not need to be experts in design. Users may not be able to communicate their precise requirements, but they are able to explain their goals and how they approach their tasks. Equipped with this kind of information, a designer can work on behalf of the users to produce the solution they need.

Techniques exist, but all of the varied approaches attract both proponents and critics, and few objective comparisons of methods or approaches are available. Contextual inquiry and ethnographic methods are appreciated because they provide detailed information about the context of use and implicit needs of users. Special emphasis in ethnography is laid on analysing a socially organised work setting (Hughes et al., 1995).

The problem that these approaches have in common appears to be the need for a closer connection to development work. Maiden and Rugg (1996) claim that requirements engineers need considerable training in the use of ethnographic methods, and that the methods may take a considerable time to master. Even supporters admit to these challenges and they identify the principal obstacle as the presentation of the results of ethnographic techniques in a form that is readily usable by designers (Hughes et al. 1995). In addition, contextual design may lead to a vast amount of raw data and may be too time- and labour-intensive (Juhl, 1996).

As Millen (2000) points out, the ever-increasing pace of new product development requires more efficient methods. Fortunately, the approaches continually improve. Several approaches have considered the cost-effectiveness of ethnographic or other field methods (e.g. Hughes et al., 1995; Millen, 2000; Sperschneider and Bagger, 2000; Viller and Sommerville, 1999a; Wixon and

Ramey, 1996; Wood, 1997). Hackos and Redish (1998) also provide a practical guide to field methods. Currently, a research challenge exists to evaluate and develop these new approaches and their effectiveness in real development contexts.

The approaches are becoming more similar, and deliberate attempts have been made to integrate, for example, ethnographically informed study of work practice and participatory design (Karasti, 2001). In the final analysis, the question may not be which approach and methods to select, but what can be learned from these methods and approaches. Thus, the methods we should use may depend on the situation. The participatory design approach is the basis for the user involvement philosophy and users' rights to influence their own tools. It introduced the idea of bringing end users into direct contact with designers. However, the roles of users and designers should be carefully considered. Ethnography offers information on how to study social aspects of work. Contextual design continues this approach and proposes, for example, the good principles of visits to users and the development of effective user-developer relationships. In addition, task analysis demonstrates the importance of goals, tasks, and task sequences.

3.2 Studies II and IV: A practical approach to early user involvement

In Study II, a practical approach to early user involvement was designed for use in real product development contexts. The approach is referred to as 'user study' to reflect an interest in everything about users and their needs: their characteristics, context of use, and tasks. In Study IV, the approach was elaborated further and a way of presenting user needs was developed.

The aim was to gain the benefits and face the challenges identified in the literature. The field study approach was selected because it seems to solve, both theoretically and practically, identified requirements elicitation problems. In 1997, when this work started, there was already evidence that studying users in their own environment would help in eliciting and providing a more complete picture of user needs (e.g. Wixon et al., 1990; Holtzblatt and Beyer, 1993; Karlsson, 1996). Field study techniques go beyond gathering only verbal data by incorporating observations made in the user's environment (Wixon et al., 1990). Being in the user's environment helps him to remember details as familiar objects are present. Field studies are focused on real users' needs: their behaviour and their context of use.

The main challenge identified was that user involvement might actually have negative effects on product development time. In addition, practitioners of field study techniques have reported such problems as how to spend less time with users, how to analyse large amounts of data, and how to compare subjective data across various users (Wixon and Ramey, 1996). Most of the product developers did not use field studies at all. In the beginning of the second research project, we explored the state of user and customer practices in three Finnish companies by interviewing product developers, and found that developers in five out of nine projects had no direct contact with users, or that contact with them was not established until the late prototype phase.

Thus, the challenge was how to conduct cost-effective field studies and how to introduce these new techniques to companies. A large number of methods are already available, but companies are unaware of them or reluctant to invest in them. Managers and developers want to have evidence of the usefulness and effectiveness of the new methods before implementing them. There was no need to develop new user involvement methods, but rather to synthesise existing approaches and methods in a practical and usable approach that would be attractive to companies.

The contribution of Study II was to provide a framework for discovering what kind of information is needed from users, what combination of elicitation methods to select, and how to use these methods effectively in practice. All this information appeared to be necessary in order that field studies could be introduced to companies. For example, developers find it difficult to know what to ask when they are interviewing users.

Special attention was paid to the cost-effectiveness and usability of the approach and to utilising the views from cognitive psychology, ethnography, contextual design, and task analysis. The principles of contextual inquiry described by Beyer and Holtzblatt (1998) were adapted and followed as general principles of field studies. These principles are context, partnership, interpretation, and focus:

- The principle of context means that users are visited in their own environment. In a real context, it is possible to observe the richness of user behaviour and environment, and refresh the users' memory of their processes and practices. The conversation is kept on a concrete level focusing on what the user is doing.

- The goal of the partnership is to make the researcher and the user collaborators in understanding the user's tasks. The users are the experts in their tasks and the researcher assumes the role of an apprentice. This improves communication between the users and the researcher, as apprentices want to know how to do the tasks, but they are not assumed to bring useful skills to the relationship.

- Interpretation is the assignment of meaning to the observation - what it implies about task structure and about possible supporting systems. Not even a small detail of the user behaviour is meaningless; the underlying reason must be understood.

- Focus means that the goal of the user visit is planned in advance. Focus gives the researcher a way to keep the conversation on relevant aspects of tasks without taking control entirely from the user.

In describing what kind of information is needed from users, various aspects are considered essential in different approaches. For example, ethnography comes from social sciences, and therefore it considers the social nature of work the most important kind of user information.

What kind of information about users is considered important depends on our theory of human nature. The present work was influenced by general psychological perspectives on human beings. In psychology, there are various theoretical points of views of human nature, as Myers (1993) states. However, in textbooks, the various perspectives are integrated and human beings are seen as biological, social, and psychological creatures that are active and goal-oriented. Thus, human behaviour is affected by both internal factors, for example cognitive issues, and by environmental issues such as culture.

One of the basic ideas of this work is that new products should adhere to intrinsic limits of human information processing capabilities. An essential part of this is to know what kind of knowledge and skills the users already posses; for example, Ericsson and Kintsch (1995) show that people are able to stretch their limited information processing capabilities if they can use their relevant knowledge from previous experience.

Johnson et al. (1988) suggest that the most important aspect of the user's knowledge is task-related knowledge. Its basic structural components are goals, operations, methods, and selection rules. It is intuitively clear that designers should know the goals of the users in order to support them, and it is also vital for them to know how the users currently achieve these goals. By identifying user goals, it is additionally possible for designers to discover the problems and needs that users have. Users do something and they achieve some goals, but they may actually want something else.

The ability of users to use a system depends on how well the system matches their goals and intended actions, and how well the users can realise their expectations and utilise their knowledge of tasks and procedures. Figure 3 illustrates the layers of user characteristics concerning the use of products. The core of the characteristics is basic human information processing capabilities, which are rather constant. General knowledge and skills widen the basic information processing capacity. Then, the domain-specific knowledge and skills widen the capacity even more.

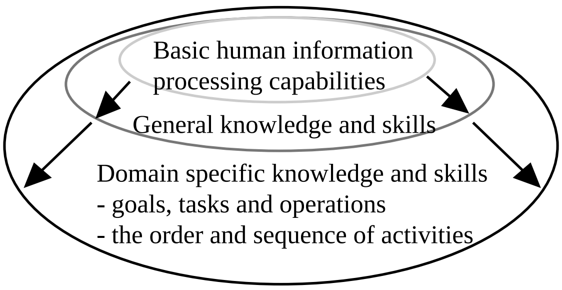

Figure 3. A model of the user characteristics concerning the use of products.

Thus, the information topics to be gathered from users are identified as:

1. Background information
The goal of gathering background information is to help the analyst to interpret the results and classify the users. Typical background information relates to age, profession, technical orientation, previous computer experience, work experience, and educational level.

2. Goals and preferences
The aim of these topics is to understand what users want to achieve and how an intended application can support their tasks and allow better ways to achieve the goals.

3. User's knowledge, skills, and experiences
The aim of these topics is to understand what users can do, how they employ objects in accomplishing their goals, and what kind of work-related processes they have.

4. Context of using an intended application
A design team should understand the physical and social context in which their application will be used in order to support the user tasks in an optimal way.

5. Pros and cons of the current tools and actions
The current tools and actions may have advantages which users are unwilling to relinquish. On the other hand, tools and actions may be suboptimal. An intended system should include most of the desirable features of the current tools and actions, while eliminating the difficulties users have with them.

The combination of three techniques was selected to elicit user information: semi-structured interviewing, thinking-aloud technique, and interactive feature conceptualisation (Figure 4). The main technique is semi-structured interviewing, which is influenced by earlier work of Bauersfeld and Halgren (1996), Beyer and Holzblatt (1996), and Wood (1997). The interviews are carried out in the natural settings of potential users, using their own task-related language. The interviewer prepares questions from each of the information topic. However, the questions are not followed strictly but are used as a checklist. The idea is to gain deeper understanding and help the user to remember details by seeing and perhaps trying the tools and artefacts being discussed. The users are encouraged to show artefacts and give demonstrations.

Two other techniques were included to augment interviewing. The interactive feature conceptualisation technique was selected to aid conversation and documenting. The aim here is to form an overall picture of the user and his context and to obtain a classification of tools, persons, places, properties, and concepts. The technique is adopted from Bauersfeld and Halgren (1996) and developed further. During the interview, tools, processes, places, persons, etc., that the user mentions are recorded on sticky notes. At the end of the interview, all sticky notes are placed on a large sheet of paper. Users are asked to rearrange the items into categories that make sense to them in their context. It was found that it is occasionally easier for the user if the interviewer places sticky notes containing places (e.g. home, car) on a sheet of paper, and the user places other notes according to them. As the notes are grouped, the interviewer tries to understand the overall picture of the user's context of use.

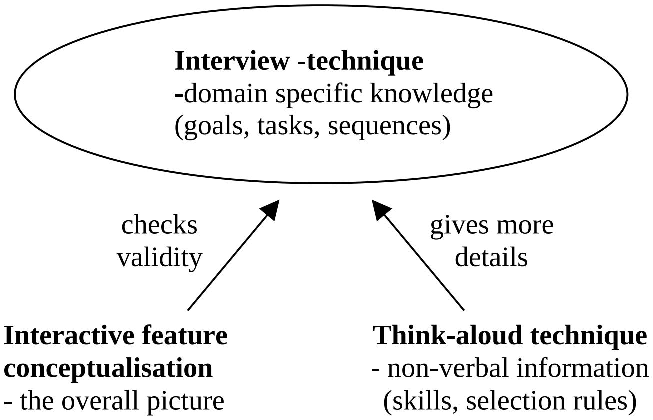

Figure 4. Information eliciting techniques.

A modified think-aloud technique was selected to uncover skills and information that cannot be gathered just by asking. Users are asked to tell how they use a tool by thinking aloud during the imagined use. Users have the tool in hand, imagine the typical use situations, and describe how they would use the tool in the situation.

The results are described in written reports organised according to information topics. In addition, two diagrams are used. The first diagram describes the overall results based on the interactive feature conceptualisation-picture. We developed further the Bauersfeld and Halgren's (1996) method so that we could use the sticky notes and a flip chart to draw a model of the user's context of use. Users placed sticky notes according to physical places, so we drew a physical world according to places. The relevant individuals, groups, tools, and artefacts were categorised according to the places. We added arrows with texts to represent communication flows and ways. The expressed user needs and problems were also included in the diagram by arrows and boxes with texts.

The second diagram is a task hierarchy diagram. The diagram shows how the user task could be redesigned, and what kind of task hierarchy is created. The diagram is similar to the task model of Johnson, Johnson, and Wilson (1995) containing the goals of the user, but from the point of view of using the future system. The diagram describes use situations, places of use, procedures, and user roles.

In Study IV, we found that it was not easy for a technically oriented designer to use user needs in product development. In one company, the designer was able to write user study reports as he was advised, but he could not see how to use the documents in user requirements definition. Therefore, a slightly more formal way of representing user needs was needed, so that designers could use the information in analysing and rationally selecting a good combination of user needs for inclusion in the future system, and to then transfer the descriptions to use cases.

A further link from user needs to user requirements was developed using user needs tables and use cases.

The user requirements activities were identifying stakeholders, gathering user needs, describing user needs by user need tables, documenting use cases, and gathering user feedback. The new activities are shown in Table 10 as they were piloted in three product development companies.

	Vaisala • new version	**KONE** • new version	**Tekla** • new product
Identifying stakeholders	X	X	X
Gathering user needs	X	X	X
Describing user needs for use cases	X	X	
Documenting use cases		X	X
Gathering user feedback			X

Table 10. Activities completed in the three case companies.

The user need tables were developed to represent user needs as users' problems and possibilities, and to link them to task sequences (Table 11). The user need table was converted to user requirements in a form of high-level use cases (Table 12).

User need tables are not sufficient to represent all user needs; other representations such as user profiles and photographs can be used in parallel with them. However, user need tables can act the means whereby several kinds of user information can be summarised in a form of user problems and possibilities. Problems are obstacles that arise from users' characteristics, their physical and social environment, overall situation. Possibilities represent users' more implicit needs, and suggest how users' tasks can be supported and improved. Later, we added a high priority column to the table, so it is possible to attach priority information to the user need table.

We hypothesised that it is easier to use user needs data in design when the findings are summarised and connected to the task sequence that forms the basis for use case descriptions. The use case driven approach is a popular solution, which software engineering provides to help with the problem of gathering and representing user requirements (Jacobson, 1995; Rumbaugh, 1994). Use cases are widely accepted among designers, providing an opportunity to transmit the user point of view to requirements engineering. In addition, use cases resemble 'scenarios of use', a popular technique in human-computer interaction (Carroll, 1995).

Task sequence:	Problems and possibilities:
Step 1: When trapped in an elevator, passenger makes an emergency alarm.	• Problem: Passengers want to get out of the elevator as soon as possible • Problem: All kinds of passengers must be able to make an alarm call (blind, foreigners etc.) • Problem: Sometimes passengers may make false alarms unintentionally. • Problem: Passengers may be in panic. • Problem: Passengers need instant confirmation that they have created a connection to the service centre operator and that they are going to get help.
Step 2: Unoccupied service centre operator receives the emergency alarm call and asks for information.	• Problem: Different versions and types of remote monitoring systems. • Problem: Passenger is the only information source. • Problem: Service centre operator does not notice the emergency alarm call.
Step 3: Service centre operator completes transmission of information to the system and sends it to the area serviceman.	• Problem: Laborious phase for the service centre operator. • Problem: Simultaneous calls must be differentiated. • Problem: Serviceman cannot see all information. • Problem: Inadequate information from a site system. • Possibility: Instructions as to how to operate the system. • Possibility: Possibility to open phone line from Call Centre to the elevator.
Step 4: Service centre operator calls the serviceman and reads the description of the failure.	• Problem: Extra work for the service centre operator.

Table 11. An example of a user need table.

Use Case:	Making An Emergency Alarm Call
Summary:	An entrapped passenger pushes the emergency alarm button in order to get help. A service centre operator receives the emergency alarm call and informs the passenger that a serviceman will come and let the passenger out of the elevator.
Actors:	Passenger and service centre operator
Preconditions:	An elevator has stopped between floors and there is a passenger in the elevator. The goal of the passenger is to get out of the elevator safely and as quickly as possible.
Basic sequence:	Step 1: The passenger presses the emergency alarm button. Step 2: The service centre operator gets a visible notification of the emergency alarm call on the screen with an optional audio signal. Step 3: The service centre operator accepts the emergency alarm call. Step 4: The system opens a voice connection between the service centre operator and the passenger. Step 5: The system indicates to both the passenger and the service centre operator that the voice connection is open. Step 6: The system guides the service centre operator as to what information to ask of the passenger. Step 7: The service centre operator informs the system that the emergency alarm call is correct.
Exceptions:	Step 1: If an entrapped passenger does not push the alarm button long enough (less than 3 seconds), the system alerts the passenger with a voice announcement. Step 7: If the passenger has pressed the emergency alarm button by accident, the service centre operator informs the system that the emergency alarm call is false. The system resets the emergency alarm call.
Post conditions:	The entrapped passenger knows that the service centre operator will contact a serviceman who will help the passenger out of the elevator safely as soon as possible.

Table 12. An example of a use case.

3.3 Studies II, III, IV and V: Evaluating the user study approach

The user study approach to early user involvement was evaluated in four case studies (II, III, IV and V).

Study II

In Study II, the user study approach was evaluated in a realistic setting of designing a personal digital assistant (PDA) application. The PDA application in question was designed to be a personal organiser with a calendar and notebook. The designers had already created their first interface prototype of the PDA software, and they were interested in gathering more information about target users to support their design work. The researcher conducted user studies with three users. The usefulness of the results was evaluated by interviewing an interaction designer and a usability specialist to whom they were given.

It was found that the user study approach is capable of providing a reasonable depth of knowledge in a short time frame with relatively low costs of 27 person hours. Both the designer and usability specialist found the overall results useful. In particular, identified user roles and situations, users' goal structure and context of use were new findings. In addition, the results helped to resolve hidden (and erroneous) assumptions of the designer. For instance, the designer seemed to have a different model of using a calendar from that of users. The designer considered deadlines important aspects of the diary. In contrast, interviewees spent considerable effort organising information and classifying to-do items while giving deadlines far less attention. It also appeared that only one out of three users was as the designer expected. The user study results helped to classify the users according to their requirements and the type of application they would need. While three users did not constitute a basis for making the necessary decisions for the next stages of the design, the results helped by making such intractable issues visible to the designer.

Study III

In Study III, the user study approach was evaluated by redesigning the functionality of an existing product based on a user study, and comparing the process and results with the baseline design process, in which the functionality was first developed.

The user study was conducted for a large Finnish company. Six users were interviewed in their work places. The study was focused on a set of functions of a portable communications device aimed at supporting mobile users. The set of functions and their interface had already been designed in the baseline design process, and now the functions were redesigned independently and without knowledge of the earlier design. A hypothesis of the user study was that the existing product could be improved by matching the functions and their labels more closely with the needs and the expectations of users.

The user study process was compared with a baseline design process that included usability tests of 33 users. The results show that the user study was useful despite the modest investment of 46 person hours. The baseline product was designed within the large company, and the design team could not offer an estimation of the time spent on design. Thus, the benefits of the user study were estimated by comparing them with other characteristics of the baseline design process. A rough estimation of the resources allocated to the usability tests can be derived from the fact that 33 users participated in them.

First, three designers who had participated in the baseline design process were interviewed to find out what kind of knowledge they had applied when they created the original functions during the baseline design process. The interview showed that the designers had incorrectly expected the users to have a similar pattern of use to the one they had themselves. The designers found the results of the user study useful for understanding the priorities of the users, their use contexts, and their specific ways of use.

Second, the analysis of the usability test results of the baseline process showed clearly that the user study results predicted most of the problems that had surfaced in the usability tests and the overall reactions of the users. In particular, all conceptual problems could be predicted by user studies.

Finally, the usability of the concrete design propositions created in the user study process was compared with the usability of the baseline functions. The comparative usability test supported the usefulness of the user studies. The users also thought that the new names matched their use better than did the baseline ones.

Table 13 summarises the results of the comparative usability test. Altogether, the four experienced users of the baseline product spent slightly more time performing the tasks than did the four novices using the new names. Two experienced users had already learned to use the functions, but two other experienced users had as many problems (nine) as the novices (eight) did in total. The novices had more difficulties with certain specific user interface features than did the experienced users (five versus two problems), but they had fewer problems in completing the task and understanding it conceptually.

Product	User group	Number of users	Problems	Mean time spent (min)
Existing	Experienced	4	9	9.51
Changed	Novices	4	8	8.18

Table 13. The results of the comparative usability test.

In summary, the results of Study III show that the user study was useful despite the modest investment of 46 person hours. All three comparisons made with the baseline design process and its results suggest that the user study outweigh the costs. The user study provided information for the designers as they had a slightly different model of user behaviour from the model that emerged from user studies. In addition, user study results seem to predict future usability problems.

The case further gave preliminary evidence that user study results lead to more usable products. The results of the user study helped the inexperienced psychologist to design a better product than that developed in the design team of the large company with the help of usability tests.

Study IV

The results of Studies II and III show that an early user study can transform design into a more informed activity. However, a psychologist performed the user studies, and there seems to be a gap between the user study results and the design ideas: the results must be interpreted and different solutions synthesised and selected. Studies II and III do not show that the user study approach could be effectively used in companies, or that developers could use the results of user studies.

In Studies IV and V, the user study approach was introduced and used in real product development contexts and by real developers. First, in Study IV the approach was piloted in three product development companies. A new approach was developed to represent the user study results in a form that are easy to use and convert to user requirements in product development. The approach is described in more details in Section 3.2.

It was discovered that the proposed approach decreases the gap between user needs and user requirements. User need tables are a useful way to represent user needs to make them understandable and useful for developers; they help the developers translate user needs to user requirements and write use cases from the user point of view. Thus, the results of the user study were easier to utilise in requirements definition. In addition, when the user need tables were missing, a developer had difficulties in writing use cases from the user point of view. The developer did not know all the details of the user tasks to be completed or the natural order of the tasks, and describing the use cases by users' language was difficult.

The case study also shows that a use case approach is not effective for gathering user needs if used in isolation; gathering user feedback with use cases is not sufficient. Users are still interpreting use cases on the basis of their current way of performing the tasks, and these implicit assumptions jeopardise the mutual understanding between users and developers. Thus, we need real data on users and their needs before use cases are documented.

Study V

In Study V, the user study results were evaluated after piloting the approach in KONE Corporation. A multidisciplinary group planned the user study, and two product developers carried out it. An outside usability researcher supported the process.

Although the investment was a modest 111,5 person hours in total, the user study results were useful for product development and the benefits to innovation and customer satisfaction were visible. The usefulness of the user study results was evaluated after presenting the user study results in a meeting. The participants were asked to evaluate the usefulness of the user study and the results and complete a questionnaire.

The respondents evaluated the results as very useful. The project manager found the results very useful for his work and the user study group found them rather useful. The salespersons expressed the view that the results did not provide them with extremely new information and that the results were not extremely important for their work, but they evaluated the results as very useful for KONE and their product development. One salesperson reported that the results were important for selling, and that a concrete proposition was needed for the customer.

This type of presentation meeting with sales persons appeared to be successful. The salespersons could relate their own experiences to the identified customer's problems, and evaluate whether the results could be generalised to other customers. They found the identified problems common among their other customers. Thus, the project manager got new information that he was not able to obtain by interviewing the salespersons earlier.

The project manager was also interviewed to discover whether he got new information and ideas from the user study, and how he planned to use the user study results. The project was deferred, so it was not possible to evaluate how the user study affected the requirements documents, but the project manager attempted to evaluate the usefulness of the results. The project manager found that certain results were already recognised and merely reinforced earlier ideas. However, he later stated that the results were full of new product ideas and he presented a long list of new information that he planned to use in requirements documenting. He planned to use almost all the findings.

The salespersons were initially slightly suspicious toward user studies and allowing development people to visit their customers. After the first study, they found the results useful for KONE, and indicated that they could present a correction plan for the customer. After the second user study, the feedback was very positive from the sales department. A salesperson sent thanks via e-mail to the user study group because KONE had made a service contract with a customer that included all their elevators and escalators. The customer evaluated KONE as superior after the user study and the salesperson concluded that the user study was significant in negotiations as the customer got "visible product development" in their own building. The costs of user studies were reasonable in the light of such a benefit.

3.4 Study V: Introducing user study approach to product development

Study V was designed to investigate ways of introducing the user study approach in product development organisations. As Bannon (1995) argues, in addition to describing techniques, the relevancy of approaches for professional system developers must be considered. Hynninen et al. (1999), from Nokia Mobile Phones, discovered challenges that they experienced in gathering user needs in an industrial setting. They found that in practice, certain skills are required of the design team. In addition, it was difficult to motivate various people in the organisation to gather and analyse data as the methods require considerable resources and the results could be seen only after weeks of hard work.

We started to improve the practices with a multi-disciplinary group and small-scale user studies were piloted to gather user needs in KONE Corporation. The results were encouraging. Persons without knowledge of behavioural science could easily learn to conduct user studies. On starting to improve the practices with a multi-disciplinary group, small-scale user study pilots motivated the group members. Although the investment was modest, the user study results were useful for product development, and the benefits to innovation and customer satisfaction were visible.

The new methods were learnt by doing, the group members could see the results quickly, and they became convinced of the importance of the gathering of user needs and of the effectiveness of the methods. The results of the first user study attracted more people to join the group.

Based on the experiences of Study V, KONE decided to invest in the gathering of data on user needs, and have recruited a usability specialist whose responsibility is to organise regularly new user studies. She continued the user study work in KONE. During the following two years after the user study pilots, KONE performed several single customer visits and four user studies, in which approximately 80 users and customers have been visited. In addition, new persons from KONE have been involved in performing user studies so that they could learn the methodology. One of the studies was conducted for a product development project and user need tables were used. The other studies concerned general context of use, but the user need tables were not suitable for them as detailed task steps were not elicited.

In Study IV, user need tables and use cases were piloted in Kone, Tekla, and Vaisala. The user study approach was found useful, as described in chapter 3.2. However, introducing user studies to Tekla and Vaisala has not proceeded as far as in Kone. Tekla joined the QURE-project later than other companies and, in the beginning, the focus has been on developing general requirements engineering processes. Only one person participated in visiting users when the user study approach was piloted. However, usability is now seen as one of the most important focus areas in Tekla, and a usability engineering group has been formed in order to improve usability engineering processes and organise training. Tekla has also decided to form a development group for improving requirements elicitation activities.

In Vaisala, the focus has been on general process improvement and no official decision has been made to pilot the user study approach. The user study approach was used in a single project in which only one person participated, and thus the results were not so comprehensive and visible in this company. The improvement of requirements engineering processes performed in the QURE-project is described in Kauppinen and Kujala (2001a, 2001b) and Kauppinen et al. (2002).

4. Conclusions

This thesis addresses issues related to early user involvement in real product development contexts. First, the benefits and challenges of user involvement identified in the literature were reviewed. It was revealed that early user involvement appears to be promising on the condition that user involvement methods such as field studies are developed to be more cost-effective, and the roles of the users and designers are carefully considered. Designers should take an active role in eliciting user information and understanding user needs through studying users' goals and behaviour in the users' own environment.

A practical approach to early user involvement referred to as 'user study' was synthesised. The goal was to find a way to apply cost-efficiently early user involvement to real product development contexts, and to evaluate the usefulness of this kind of early user involvement in real product development. The main results are summarised in Table 14.

Study	Problem	Results
I	What are the benefits and challenges of user involvement in product development?	User involvement has clearly positive effects on system success and user satisfaction. The communication between users and developers poses challenges to product development work. Field study methods should be more cost-effective to use.
II	How can early user involvement be applied in product development?	A practical approach to user involvement 'user study' was developed. The approach was tested in one case, and the results were evaluated to be useful although the resources invested were modest.
III	What are the benefits and costs of the proposed approach to early user involvement compared to usability testing?	The user study approach was evaluated to provide useful information for product development. Preliminary evidence suggested that user studies are a more effective way of improving usability of the product than iterative usability testing.
IV	How can user needs be represented and translated into user requirements in industrial product development cases?	User need tables were developed to represent user needs. It was discovered that the user needs tables help designers to bridge the gap between the user needs and user requirements when the use case approach is used.
V	How can the proposed approach be introduced to product development contexts?	In introducing user studies to product development small-scale pilot studies motivated the developers. Developers and salesmen found user studies useful. Innovation and customer satisfaction were increased.

Table 14. The research problems and results in the articles.

4.1 Implications

The principal implications of the four case studies are as follows:

1) Early user involvement is useful even in a short time frame with relatively low costs.
The results provide preliminary evidence that user studies in the early phase of product development represent a more effective way of improving usability of the product than iterative usability testing (Study II). Developers, a usability expert, and salesmen found the results of early user involvement useful (Studies II, III, V). Innovation and customer satisfaction were increased (Study V).

2) The proposed user study approach is a practical way of involving users and gathering their needs.
The real product development case showed that persons without knowledge of behavioural science can easily learn to do user studies if they receive support (Studies IV, V). The approach was cost-efficient and the results were useful despite the modest costs (Studies II, III, IV, V). User need tables were introduced as a new representation for displaying the results. It was found that user need tables help designers to bridge the gap between user needs and user requirements when the popular use case approach is used (Study IV).

3) In introducing user studies to product development, small-scale pilot studies and multidisciplinary group work represent an effective strategy.
Group members committed to improvement actions as they planned them. The group learned by doing and new people could participate when others already had experience with conducting user studies. Starting with a small-scale user study was found to be motivating. Group members could see the results quickly and became convinced of the importance of gathering data on user needs and the effectiveness of the methods (Study V). The user study approach was simple enough to be introduced to companies, in which no usability group existed. It is a considerable challenge to train and disseminate advanced user-centred methods even where usability is recognised as one of the strategy focus areas in a large company (Carlshamre and Rantzer, 2001).

The practical significance of the results can be evaluated by comparing them to earlier research findings.

First, in the literature, field methods are frequently considered complicated in practice (e.g. Hynninen et al., 1999). The present study showed that simple methods are valuable and easier to introduce to product development. Simple methods also reduce the risk that usability expertise itself might constitute a bottleneck in disseminating new practices company wide, as was the case in Ericsson (Carlshamre and Rantzer, 2001).

It is typical that product development companies generally practice usability testing (Darnell and Halgren, 2001). Now, we have initial evidence that it can be even more effective to allocate usability efforts to earlier development phases, and to study and understand users even before a prototype exists. Thus, we can also influence the effectiveness and satisfaction components of product quality by ensuring that users can achieve their goals by using the product in their own context in a way that satisfies their needs. However, usability testing is also needed to test the user interface solution and the interpretations made from user studies.

Furthermore, it is well known that several developers do not want anything that represents addendum to the official and established development process rather than an integral part of it (Carlshamre and Rantzer, 2001). In the present study, early user involvement was linked to ordinary product development through utilising a common use case oriented approach to documenting user requirements. Technical developers were aided in using the user need information by linking it through user need tables to use cases.

4.2 Characterising the user study approach compared to other approaches

Varied field methods have their own strengths and weaknesses. We can characterise the user study approach and compare it with other approaches. The user study approach is simple and cost-efficient, and it provides a comprehensive view of users, including their goals, cognitive aspects, and social and physical environment. However, the information it provides may not be so profound as more time consuming approaches could uncover. However, as Wixon (1995) suggests, the methods should be judged according to their purpose. The user study approach is useful in the requirements elicitation phase, when we already have an idea of the product, and we are gathering and documenting user needs and requirements. Its strength lies in product development, where time frames are short and where the alternative is to do nothing for gathering user needs.

Wixon (1995) suggests a framework for considering qualitative research methods in design and development. The framework consists of a set of dimensions, which can be used to categorise qualitative methods and the kind of understanding that method produces.

One of the dimensions behind both data analysis and the overall consideration of data gathering methods is whether one is gathering data to discover aspects of user work or whether one is aiming to make decisions or to prioritise development work for an engineering team (Wixon, 1995). Wixon (1995) suggests that field methods are at the discovery end of the continuum and thus avoid the prior specification of user tasks, metrics or performance, and measurement in general. Instead, the emphasis is on discovering what users do, how they do it, in what context they do it, and why they do it. Discovery-oriented methods are more appropriate in the early parts of the development process and in some domains could be viewed as a prerequisite to the more decision- and measurement-oriented parts of the process.

User study approach can be characterised in a similar way. However, it is slightly nearer to the decision end of the continuum than are other field methods. In the data gathering phase, the categories of relevant information are already fixed, and the information is gathered and analysed according to these categories. This renders the approach more effective and easier to use, but some information may be missed. In particular, when the user is not observed for extended periods, certain non-verbal information is missed.

Moreover, we can characterise user study approach as a top down approach, as Wood (1996) characterises ethnographic interviewing. He refers to contextual inquiry as a bottom-up approach because of its usual emphasis on first observing and collecting the large amounts of data, and then inductively abstracting work flows and other more general descriptions of the tasks being analysed. In both Wood's approach and the user study approach interviewing results are used as a general framework within which to interpret specific observations and samples of real work. In the top down approach, certain details of understanding may be lost as it does not start from scratch; however it is more cost-efficient for product development purposes.

Similarly, user need tables in the user study approach are one way to transforming user-centred analysis to concrete design among many others described in Wood (1998). For example, Graefe (1998) describes how scenarios can be converted to use cases. However, in user need tables, a special effort is made to link user need descriptions and use cases clearly, and make the transformation easy. As Wood (1998) states in his introduction to the book, there are several effective ways to build the bridge, each suited to particular contexts and constraints. User need tables accord with situations where cost-effectiveness and representing formal user requirements for design are needed.

However, the elicitation methods of the user study approach cannot be characterised as unique, but based on methods such as ethnographic interviewing (Wood, 1996, 1997). In addition, several other approaches have recently been introduced in order to render ethnography or other field methods more cost-efficient and practical. As Viller and Sommerville (1999b) suggest, work has not been unsuccessful but frequently falling short of fitting in with the day-to-day practice of systems designers. In addition, there is the challenge of introducing new approaches to practical product development. For example, Hughes et al. (1995) emphasise the importance of determining focus in "quick and dirty" ethnography studies so that field studies might not be such a prolonged activity. They also recommend debriefing meetings between ethnographers and designers. However, these field methods depend on the special skills of ethnographers.

Viller and Sommerville's (1999a) approach is a promising step towards utilising ethnography in design, but it is not broadly used. They present a set of social viewpoints and concern questions to support the analysis work. In their case, designers performed the analysis themselves, supported by the ethnographically-informed guidance.

4.3 Limitations

The limitations of this research work are evaluated here using Kitschenham's et al. (1995) case study guidelines. This work was based on a literature review and four case studies. The general weakness of the case study methodology is that the results may not be generalisable to other development conditions. Thus, the results of this work may be valid only as far as there are similar development conditions such as product type and company size. However, the work was carried out in multiple cases with five different product development companies, so the results cover more than one case and company.

Kitchenham et al. (1995) recommend that the selected pilot projects are representative of the type of projects the company generally undertakes. They claim, however, that it may be difficult in practice to control the choice of case-study projects and so it was in this research. We had to select starting projects of the moment. However, the projects were general projects in these companies, and the case studies were conducted in several different projects in several companies.

In addition, the effect of confounding factors should be minimised in selecting pilot projects. However, we had to select projects in which the members were enthusiastic enough to take part in case studies. Thus, more research work is needed in other product development companies and situations, in which the developers are more sceptical towards the user-centred approach.

We evaluated the user study approach in cases where the participants used these kinds of user-centred methods for the first time, and were learning to use them. Kitchenham et al. (1995) warn against confounding learning how to use a method or tool with trying to assess its benefits. The effects of learning to use the method or tool might interfere with the benefits of using it. However, the introduction and learning phases were considered essential parts of the usefulness of the user study approach, and the goal was to develop an easy-to-learn and cost-efficient approach. Therefore, it was in fact desirable that the effects of learning interfered with the benefits of using it.

4.4 Future challenges

Implementing early user involvement company wide is the greatest challenge in the future. By way of example, Carlshamre and Rantzer (2001) cite the difficulties experienced by Ericsson, where in fact usability has been recognised as one of the strategy focus areas. They found that maturity of usability and introducing methods is not only about depth, how well methods are introduced; it is also about breadth, how widely the methods are accepted throughout the company. In their case, it was a considerable challenge to train and disseminate advanced user-centred methods, and usability expertise itself constituted a bottleneck. The dissemination of good practices from single projects throughout an organisation requires special effort.

Similarly, KONE has hired a usability specialist whose responsibility is to organise regularly new user studies. The next step in this company is to plan systematically the implementation of good practices. The group in this study included usability oriented people, but now they must convince the most technically oriented practitioners of the importance of gathering data on user needs. In our experience, they additionally need support in using the user study methods.

The last two years after piloting the user study approach have already shown that KONE has continued to use the user study approach; four user studies and several single customer visits have been performed and approximately 80 users and customers met. In addition, new persons have been involved in performing user studies in order to learn the methodology. Their sales organisation has suggested more user study targets, as they consider user studies important for their customers and their own organisation. User study document templates have been developed, but now it is considered that a more formal user study process needs to be defined so that other persons can perform user studies.

In addition, user study methods need to be refined to be yet more practical in the industrial setting. Analysing and reporting results were found to be demanding in the case studies, so this part of the work requires some improvement. The impression of working with "a huge pile of unstructured data" was not, however, experienced, as the entire study was performed in a structured manner. Nevertheless, more straightforward tools and more routine ways of handling digital photographs and videotapes are needed in order to present the results as clearly as possible to product development engineers.

References

Abbott, R. J. (1986). An Integrated Approach to Software Development. New York: John Wiley & Sons.

ACM SIGCHI (1996). Curricula for Human-Computer Interaction. A report of ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group. New York: ACM. http://www.acm.org/sigchi/cdg/

Avison, D., Lau, F., Myers, M., and Nielsen, P. A. (1999). Action research. Communications of the ACM, 42, 1, 94-97.

Bannon, L. J. (1995). The politics of design: Representing work. Communications of the ACM, 38, 9, 66-68.

Baroudi, J. J., Olson, M. H., and Ives, B. (1986). An empirical study of the impact of user involvement on system usage and information satisfaction. Communications of the ACM, 29, 3, 232-238.

Bauersfeld, K. and Halgren, S. (1996). "You've got three days!" Case studies in field techniques for the time-challenged. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: Wiley.

Bekker, M. and Long, J. (2000). User involvement in the design of human-computer interactions: Some similarities and differences between design approaches. In McDonald, S., Waern, Y., and Cockton, G. (eds.), People and Computers XV (Proceedings of HCI'2000), Springer-Verlag, pp. 135-147.

Beyer, H. and Holtzblatt, K. (1996). Contextual techniques starterkit. Interactions, 3, 6, 44-50.

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

Bias, R. G., Mayhew, D. J. (Eds.) (1994). Cost-Justifying Usability. San Diego, CA: Academic Press.

Blackburn, J., Scudder, G., and Van Wassenhove, L. N. (2000). Concurrent software development. Communications of the ACM, 43, 11, 200-214.

Blomberg, J, Giacomi, J., Mosher, A., and Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In Participatory Design: Principles and Practices. Hillsdale, NJ: Lawrence Erlbaum, pp. 123-155.

Blomberg, J., Suchman, L., Trigg, R. H. (1996). Reflections on a work-oriented design project. Human-Computer Interaction, 11, 237-265.

Bly, S. (1997). Field work: Is it product work? Interactions, 4, 1, 25-30.

Carlshamre, P. and Rantzer, M. (2001). Dissemination of usability: Failure of a success story. Interactions, 8, 1, 31-41.

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

Chatzoglou, P. C. and Macaulay, L. A. (1996). Requirements capture and analysis: A survey of current practice. Requirements Engineering, 1, 2, 75-87.

Coble, J. M., Karat, J., and Kahn, M. G. (1997). Maintaining a focus on user requirements throughout the development of clinical workstation software. In Proceedings of the CHI'97, Conference of Human Factors in Computing Systems, ACM, pp. 170-177.

Constantine, L. L. (1995). Essential modeling: Use cases for user interfaces. Interactions, 2, 2, 34-46.

Coutaz, J. and Taylor, R. N. (1994). Introduction to the workshop on software engineering and human-computer interaction: Joint research issues. In "Software Engineering and Human-Computer", Lecture Notes In Computer Science 896. Berlin: Springer-Verlag, pp. 1-3.

Damodaran, L. (1996). User involvement in the systems design process–a practical guide for users. Behaviour & Information Technology, 15, 6, 363-377.

Darnell, E. and Halgren, S. (2001). Usability throughout the product development cycle. In Branaghan, R. J. (Ed.) Design by People for People: Essays on Usability. Chicago: Usability Professionals' Association.

Davis, A. M. (1993). Software Requirements - Objects, Functions, and States. New Jersey: Prentice Hall.

Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A. and Theofanos, M. (1997). Identifying and Measuring Quality in a Software Requirements Specification. In Thayer, R. H. and Dorfman, M. (Eds.), Software Requirements Engineering. Washington: IEEE Computer Society Press, pp. 164-175.

Diaper, D. (1989). Task observation for Human-Computer Interaction. In Diaper, D. (Ed.) Task Analysis for Human-Computer Interaction. New York: Wiley, pp. 210-251.

Dray, S. M. and Karat, C. (1994). Human factors cost justification for an internal development project. In Bias, R. G. and Mayhew, D. J. (Eds.) Cost-justifying Usability. San Diego, CA: Academic Press.

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

Ehrlich, K. and Rohn, J. A. (1994). Cost justification of usability engineering: A vendor's perspective. In Bias, R. G. and Mayhew, D. J. (Eds.) Cost-justifying Usability. San Diego, CA: Academic Press, pp. 73-110.

Ericsson, K. A. and Kintsch, W. (1995). Long-term working memory. Psychological Review, 102, 211-245.

Fairley, R. E. and Thayer, R. H. (1997). The concept of operations: The bridge from operational requirements to technical specifications. In Thayer, R. H. and Dorfman, M. (Eds.), Software Requirements Engineering (2nd Edition). Los Alamitos, California: IEEE Computer Society Press.

Ford, J. M. and Wood, L. E. (1996). An overview of ethnography and system design. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: John Wiley & Sons, pp. 269-282.

Forsberg, K. and Mooz, H. (1997). System engineering overview. In Thayer, R. H. and Dorfman, M. (Eds.), Software Requirements Engineering (2nd Edition). Los Alamitos, California: IEEE Computer Society Press.

Foster, S. T. and Franz, C. R. (1999). User involvement during information systems development: A comparison of analyst and user perceptions of system acceptance. Journal of Engineering and Technology Management, 16, 329-348.

Glass, R. L. (1994). The software research crisis. IEEE Software, 11, 6, 42-47.

Goguen, J. A. and Linde, C. (1993). Techniques for requirements elicitation. In Thayer, R. H. and Dorfman M. (Eds.), Software Requirements Engineering (2nd Edition). Los Alamitos, California: IEEE Computer Society Press.

Good, M. (1992). Participatory design of a portable torque-feedback device. In Proceedings of CHI, Conference of Human Factors in Computing Systems, ACM, pp. 439-446.

Gould, J. D. and Lewis, C. (1985). Designing for usability: Key principles and what designers think. Communications of the ACM, 28, 3, 300-311.

Graefe, T. M. (1998). Transforming representations in user-centered design. In Wood, L. E. (Ed.), User Interface Design: Bridging the Gap From User Requirements to Design. Boca Raton, FL: CRC Press, pp. 57-80.

Grudin, J. (1991a). Interactive systems: Bridging the gaps between developers and users. IEEE Computer, 24, 4, 59-69.

Grudin, J. (1991b). Systematic sources of suboptimal interface design in large product development organization. Human-Computer Interaction, 6, 2, 147-196.

Hackos, J. T. and Redish, J. C. (1998). User and Task Analysis for Interface Design. New York: Wiley.

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, 226-236.

Hofmann, H. F. and Lehner, F. (2001). Requirements engineering as a success factor in software projects. IEEE Software, 18, 4, 58-66.

Holzblatt, K. and Beyer, H. (1993). Making customer-centered design work for teams. Communications of the ACM, 36, 10, 93-103.

Hudlicka, E. (1996). Requirements elicitation with indirect knowledge elicitation techniques: Comparison of three methods. In Proceedings of the Second International Conference on Requirements Engineering ICRE'96, IEEE, pp. 4-11.

Hudlicka, E. (1999). Knowledge elicitation in complex military environments. In Proceedings of IEEE Conference and Workshop on Engineering of Computer-Based Systems, pp. 328-224.

Hughes, J., King, V., Rodden, T., and Andersen, H. (1995). The role of ethnography in interactive systems design. Interactions, 2, 2, 56-65.

Hynninen, T., Liukkonen-Olmiala, T., and Kinnunen, T. (1999). No pain, no gain, applying user-centered design in product concept development. In

Brewster, S., Cawsey, A., and Cockton, G. (Eds.), Human-Computer Interaction – INTERACT ’99 (Volume II). Proceedings of the Seventh IFIP Conference on Human-Computer Interaction, IOS Press, pp. 201-205.

IEEE Std 830 (1998). IEEE Recommended Practice for Software Requirements Specifications. IEEE Standard.

Imaz, M. and Benyon, D. (1999). How stories capture interactions. In A. Sasse and C. Johnson (Eds.), Human-Computer Interaction—INTERACT '99: Proceedings of the Seventh IFIP Conference on Human-Computer Interaction, Vol. 1, IOS Press, pp. 321-328.

ISO 13407 (1999). Human-centred design processes for interactive systems. ISO/TC159/SC4. International Standard.

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

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

Johnson, P. (1989). Supporting system design by analyzing current task knowledge. In Diaper, D. (Ed.) Task Analysis for Human-Computer Interaction. New York: John Wiley & Sons, pp. 161-185.

Johnson, P., Johnson, H., Waddington, R., and Shouls, A. (1988). Task-related knowledge structures: Analysis, modelling and application. In D. M. Jones and R. Winder (Eds.), People and Computers IV. Cambridge: Cambridge University Press, pp. 35-62.

Johnson, P., Johnson, H., & Wilson, S. (1995). Rapid prototyping of user interfaces driven by task models. In J. M. Carroll (Ed.), Scenario-Based Design: Envisioning Work and Technology in System Development. New York: John Wiley & Sons.

Juhl, D. (1996). Using field-oriented design techniques to develop consumer software products. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: John Wiley & Sons.

Karasti, H. (2001). Increasing sensitivity towards everyday work practice in system design. Acta Universitatis Ouluensis, A Scientiae Rerum Naturalium 362. Oulu: Oulu University Press.

Karat, C. (1994). A comparison of user interface evaluation methods. In Nielsen, J. and Mack, R., (Eds.), Usability Inspection Methods. New York: John Wiley & Sons.

Karat, C. (1997). Cost-justifying usability engineering in the software life cycle. In Helander, M., Landauer, T. K., and Prabhu, P. (Eds.) Handbook of Human-Computer Interaction, Second edition. Amsterdam: Elsevier Science, pp. 653-688.

Karat, J. (1997). Evolving the scope of user-centered design. Communications of the ACM, 40, 7, 33-38.

Karlsson, M. (1996). User Requirements Elicitation, A Framework for the Study of the Relation between User and Artefact. Thesis for the degree of Doctor of Philosophy. Göteborg: Chalmers University of Technology.

Kauppinen, M. and Kujala, S. (2001a). Assessing requirements engineering processes with the REAIMS model: Lessons learned. In Proceedings of Eleventh Annual International Symposium of the International Council on Systems Engineering (INCOSE2001), (Melbourne, Australia, 1-5 July).

Kauppinen, M. and Kujala, S. (2001b). Starting improvement of requirements engineering processes: An experience report. In Proceedings of 3rd International Conference on Product Focused Software Process Improvement (Profes2001). Berlin Heidelberg: Springer-Verlag, pp. 196-209.

Kauppinen, M., Kujala, S., Aaltio, T., and Lehtola, L. (2002). Introducing requirements engineering: How to make a cultural change happen in practice. In Proceedings of IEEE Joint International Requirements Engineering Conference (RE' 02), (Essen, Germany, 9-13 September).

Keil, M. and Carmel, E. (1995). Customer-developer links in software development. Communications of the ACM, 38, 5, 33-44.

Kitchenham, B., Pickard, L., Pfleeger, S. L. (1995). Case studies for method and tool evaluation. IEEE Software, 12, 4, 52-62.

Kitsuse, A. (1991). Why aren't computers… Across the Board, (October), 28, 44-48.

Kotonya, G. and Sommerville, I. (1998). Requirements Engineering, Processes and Techniques. Chichester: John Wiley and Sons.

Leffingwell, D. and Widrig, D. (2000). Managing Software Requirements: A Unified Approach. Boston: Addison-Wesley Publishing.

Leveson, N. G. (2000). Intent specifications: An approach to building human-centered specifications. IEEE Transactions on Software Engineering, 26, 1, 15-35.

Lund, A. M. (1997). Another approach to justifying the cost of usability. Interactions, 4, 3, 48-56.

Madsen, K. and Aiken, P. H. (1993). Experiences using cooperative interactive storyboard prototyping. Communications of the ACM, 36, 4, 57-63.

Maiden, N. A. M., Rugg, G. (1996). ACRE: selecting methods for requirements acquisition. Software Engineering Journal, 11, 3, 183-192.

Mantei, M. M., Teorey, T. T. J. (1988). Cost/benefit analysis for incorporating human factors in the software lifecycle. Communications of the ACM, 31, 4, 428-439.

Mayhew, D. J. (1999). The Usability Engineering Lifecycle. San Francisco, Morgan Kaufmann Publishers.

McKeen, J. D., Guimaraes, T. (1997). Successful strategies for user participation in systems development. Journal of Management Information Systems, 14, 2, 133-150.

Millen, D. R. (2000). Rapid ethnography: Time deepening strategies for HCI field research. In Proceedings of DIS'00 Conference (Designing Interactive

Systems: Processes, Practices, Methods, and Techniques). New York: ACM Press, pp. 280-286.

Mitchell, A. A. and Chi, M. T. H. (1984). Measuring knowledge within a domain. In Nagy P. (Ed.) The Representation of Cognitive Structures. Toronto: Ontario Institute for Studies in Education.

Muller, M. J. and Carr, R. (1996). Using the CARD and PICTIVE participatory design methods for collaborative analysis. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: John Wiley & Sons.

Myers, D. G. (1993). Exploring Psychology, (2nd ed.). New York: Worth Publishers.

Nielsen, D. (1993). Usability Engineering. London: Academic Press.

Noyes, J. M., Starr, A. F. and Frankish, C. R. (1996). User involvement in the early stages of the development of an aircraft warning system. Behaviour & Information Technology, 15, 2, 67-75.

Page, S. R. (1996). User-centered design in a commercial software company. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: John Wiley & Sons, pp. 197-213.

Palmer, J. D. (1987). Uncertainty in software requirements. Large Scale Systems, 12, 257-270.

Poltrock, S. E., Grudin, J. (1994). Organizational obstacles to interface design and development: Two participant — observer studies. ACM Transactions on Computer-Human Interaction, 1, 1, 52-80.

Potts, C. (1993). Software-engineering research revisited. IEEE Software, 10, 5, 19-28.

Reed, S. (1992). Who defines usability? You do! PC/Computing, (Dec), 220-232.

Regnell, B. (1999). Requirements Engineering with Use Cases - a Basis for Software Development. Reports on Communication Systems 132. Lund University: Lund.

Rombach, H. D. (1990). Software Specification: A Framework. SEI Curriculum Module SEI-CM-11-1.2, Software Engineering Institute, Carnegie Mellon University, January, 1990.

Rowley, D. E. (1996). Organizational considerations in field-oriented product development: Experiences of a cross-functional team. In Wixon, D. and Ramey, J. (Eds) Field Methods Casebook for Software Design. New York: John Wiley & Sons.

Rudd, J., Stern, K., and Isensee, S. (1996). Low vs. high-fidelity prototyping debate. Interactions, 3, 1, 76-85.

Rumbaugh, J. (1994). Getting started - Using use cases to capture requirements. Journal of Object-Oriented Programming, September 1994, 8-23.

Smith, G. F. (1989). Representational effects on the solving of an unstructured decision problem. IEEE Transactions on Systems, Man, and Cybernetics, 19, 5, 1083-1090.

Sommerville, I. (1996). Software Engineering (5th ed.). Wokingham, England: Addison-Wesley.

Sommerville, I. and Sawyer, P. (1997). Requirements Engineering: A Good Practice Guide. Chichester: John Wiley & Sons.

Sperschneider, W. and Bagger, K. (2000). Ethnography fieldwork under industrial constraints: Towards Design-in-Context. In Gulliksen, J., Oestreicher, L., Severinson Eklundh, K. (Eds.), NordiCHI2000, Design versus Design (Proceedings of 1st Nordic Conference on Computer-Human Interaction), pp. 1-7.

Stevens, R. (1997). Last words on the (project) Flight Recorder: A plea for user requirements. Requirements Engineering, 2, 1, 61-62.

Thayer, R. H. and Thayer, M. C. (1997). Software requirements engineering glossary. In Thayer, R. H. and Dorfman, M. (Eds.), Software Requirements Engineering (2nd Edition). Los Alamitos, California: IEEE Computer Society Press.

Tichy, W. F., Haberman, N., and Prechelt, L. (1993). Summary of the Dagstuhl workshop on future directions in software engineering. ACM SIGSOFT Software Engineering Notes, 18, 1, 35-48.

Tyldesley, D. A. (1988). Employing usability engineering in the development of office products. The Computer Journal, 31, 5, 431-436.

Ulrich, K. T. and Eppinger, S. D. (2000). Product Design and Development (2nd Edition). Boston: McGraw-Hill.

Viller, S. and Sommerville, I. (1999a). Social analysis in the requirements engineering process: From ethnography to method. In Proceedings of the IEEE International Symposium on Requirements Engineering (Limerick, 1999) IEEE Press, pp. 6-13.

Viller, S. and Sommerville, I. (1999b). Coherence: An approach to representing ethnographic analyses in systems design. Human-Computer Interaction, 14, 1&2, 9-41.

Wilson, A., Bekker, M., Johnson, H., Johnson, P. (1996). Costs and benefits of user involvement in design: Practitioners' views. In People and Computers XI, Proceedings of HCI'96, pp. 221-240.

Wilson, A., Bekker, M., Johnson, P., and Johnson, H. (1997). Helping and hindering user involvement – A tale of everyday design. Proceedings of CHI’97, Conference of Human Factors in Computing Systems, ACM, 178-185.

Wixon, D. (1995). Qualitative research methods in design and development. Interactions, 2, 4, 19-26.

Wixon, D., Holtzblatt, K., and Knox, S. (1990). Contextual Design: An emergent view of system design. Proceedings of CHI'90, Conference of Human Factors in Computing Systems, ACM, pp. 329-336.

Wixon, D., Jones, S., Tse, L., Casady, G. (1994). Inspections and design reviews: Framework, history, and reflection. In Nielsen, J. and Mack, R., (Eds.), Usability Inspection Methods. New York: John Wiley & Sons.

Wixon, D. and Ramey, J. (Eds.) (1996). Field Methods Casebook for Software Design. New York: John Wiley & Sons.

Wixon, D. and Wilson, C. (1997). The usability engineering framework for product design and evaluation. In Helander, M., Landauer, T. K., and Prabhu, P.

Englewood (Eds.) Handbook of Human-Computer Interaction (2ed Ed.), Cliffs, N.J.: Elsevier Science.

Wood, L. E. (1996). The ethnographic interview in user-centered work/task analysis. In Wixon, D. and Ramey, J. (Eds.) Field Methods Casebook for Software Design. New York: Wiley, pp. 35-56.

Wood, L. E. (1997). Semi-structured interviewing for user-centered design. Interactions, 4, 2, 48-61.

Wood, L. E. (Ed.) (1998). User Interface Design: Bridging the Gap From User Requirements to Design. Boca Raton, FL: CRC Press.

Yin, R. K. (1994). Case Study Research: Design and Methods, Second Edition. Thousand Oaks, CA: SAGE Publications.