This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Kujala, Sari

Effective user involvement in product development by improving the analysis of user needs

Published in:
BEHAVIOUR AND INFORMATION TECHNOLOGY

DOI:
[10.1080/01449290601111051](https://doi.org/10.1080/01449290601111051)

Published: 01/01/2008

Document Version
Peer reviewed version

Please cite the original version:
Kujala, S. (2008). Effective user involvement in product development by improving the analysis of user needs. *BEHAVIOUR AND INFORMATION TECHNOLOGY*, 27(6), 457-473.
<https://doi.org/10.1080/01449290601111051>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

The original place of publication in Behaviour and Information Technology, 27, 6, pp. 457-473:

www.tandfonline.com/doi/full/10.1080/01449290601111051#abstract

Effective user involvement in product development by improving the analysis of user needs

SARI KUJALA

Helsinki University of Technology, Software Business and Engineering Institute,
P.O. Box 9210, FIN-02015 TKK, Finland; tel. +358 50 3862768, fax +358 9 451 4958

Email: sari.kujala@tkk.fi

User involvement has been shown to be beneficial in the development of useful and usable systems. The trend of software development becoming a product-oriented activity creates challenges to user involvement. Field studies appear a promising approach, but the analysis of the gathered user needs has been shown to be demanding. This study presents, on the basis of seven case studies, an early user-involvement process showing how user needs can be analysed and how the input to product development can be identified. In addition, the process is tested in two industrial cases with interviews and a questionnaire. The results show that the process supports effective early user involvement; the resulted requirements were evaluated as being more successful and their quality as better than average. However, the case studies show that user involvement not only provides useful information about users' needs but also increases the understanding of users' values.

Keywords: User involvement, user-centred design, product development, requirements elicitation, user need analysis, field studies

1. Introduction

User involvement is a widely accepted principle in the development of useful and usable software systems (Karat, 1997; Wilson et al., 1997; Bekker and Long, 2000). A lack of user involvement has been repeatedly associated with failed software projects and the benefits of user involvement are shown in several studies (Kujala, 2003, Kujala et al., 2005). However, the mechanism of user involvement, what in fact makes it useful, is not so clearly formulated. There are varied approaches to user involvement and they have different rationales explaining why to involve users. For example, participatory design emphasises democracy and skill enhancement (Ehn, 1993). Other approaches focus mostly on gaining varied information from users. For example, ethnography focuses on the social aspects of human co-operation and user-centred design and contextual design on the context of use.

In addition, software development is increasingly becoming a package- (or product-) oriented activity rather than a custom activity (Carmel and Becker, 1995). Product development contexts and the increasing numbers of users create special challenges to user involvement as Iivari and Iivari (2006) point out. However, participatory design, which is the principal approach to user involvement, focuses mainly on internal or custom development and efforts to adapt the approach to product development have not turned out to be

flawless (Grudin and Pruitt, 2002). Thus, the form and rationale of user involvement need to be reconsidered to fit product development purposes.

The informative role of users is one of the most prevalent rationales for user involvement, as can be seen in Gould and Lewis' (1985) recommendation that the design team should be brought into direct contact with potential users, as opposed to hearing or reading about them through human intermediaries. In addition, Keil and Carmel (1995) suggest that it is less desirable to use indirect links to customers and users because of the information filtering and distortion that can occur when intermediaries or customer surrogates are used. Nowadays, most usability experts consider it impossible to design a usable product without getting feedback from real users; eliciting user requirements is also recommended (ISO 13407, 1999).

On the other hand, there has been discussion as to whether users should be active co-designers and not be considered informants or mere subjects (Muller et al., 1997; Olsson, 2004). Olsson (2004) points out that if users serve as information sources but have no decision-making rights, they still have little or no influence on design. She presents a qualitative analysis of the contributions to design by user representatives compared with interaction designers. One of the contributions seems to be that users had their own values compared to designers and by active participation they could convey these values and their domain knowledge for use by developers. On the basis of earlier experience, Olsson (2004) states that user representatives concentrate on functionality and effectiveness, in comparison with interaction designers, who spend more time on appearance and presentation.

The discussion so far demonstrates that user involvement is not simple information-gathering, but that developers and users have different vocabularies, interests, and values, which makes the communication and interplay complicated. On the other hand, user involvement can be seen as a potential input for designing systems that support human values. By human value, we mean anything that has personal or social worth or meaning. The values may be learned in childhood or adapted as the person takes part in a social group. Value-sensitive design has received growing attention (Friedman, 1996), but the discussion has been on a rather general level, concerning basic human values such as user autonomy, and more could be done to understand how relevant values for a particular project can be discovered (Flanagan et al., 2005).

The focus of this study is on user involvement, particularly in product-development contexts. The goal of this study is to investigate how early user involvement could be effective in product development where the large number of users makes their participation impossible. User involvement is most efficient and influential in the early stages of system development and field studies appear to be a promising approach to early user involvement. However, field studies are often found time consuming, and the analysis of user needs is particularly challenging.

First, the forms of user involvement and the motives as to why users should be involved are discussed. Second, user involvement and its practical realisation in product development context are described. Furthermore, on the basis of the experience gained in seven case studies, specific steps of early user involvement are presented to support the analysis and utilisation of user needs in product development. Finally, the process is evaluated in two industrial cases and the results are discussed.

2. Variety of user involvement

User involvement can be seen as a general term describing direct contact with users and covering many approaches, such as user-centred design, participatory design, ethnography, and contextual design (Kujala, 2003). Furthermore, user involvement is not a one-dimensional concept, but we need to consider the varied forms and motives of involvement in order to apply it to product development.

2.1 Forms of user involvement

The form of user involvement can be broadly characterised as being somewhere on the continuum from informative, through consultative, to participative, as illustrated in Figure 1, adapted from Damodaran (1996). Users may take active roles or they may be involved as providers of information, commentators, or objects for observations.

Figure 1. Forms of user involvement (Damodaran, 1996).

Participatory design is sometimes used interchangeably with user involvement, but it is located at the participatory end of Damodaran's (1996) continuum. Participatory design originated in Scandinavia, where democratic participation and skill enhancement are important features of participatory design (Ehn, 1993). Thus, participatory design originally included the idea of users actively participating in design work. However, Kuhn and Muller (1993) state that outside Scandinavia, the field is more varied, with some theorists and practitioners pursuing a locally adapted form of democratic decision-making and others emphasising effective knowledge acquisition and product quality.

Thus, user involvement may have different forms; the essential distinction between these is how active users' roles are, whether they influence decisions, or whether they even participate in development work. The triangles in Figure 1 illustrate the typical forms of user involvement in product development compared to in-house development. Participatory design is characteristic of in-house development, in which users are known and willing to participate in development work (Rauterberg et al., 1995). It can also be the recommended form of user involvement, as users have possibility of influencing decisions relating to the system. As apparent from Olsson's (2004) work, users and developers may have very different interests, values, and vocabularies. Thus, for user-centred design, user involvement may not have optimal results if the design has been conducted in terms of developers, as developers could ask questions from their point of view and interests and make their own interpretations and decisions. On the other hand, the typical form of user involvement is informative and consultative in product development, as the potential users are not usually interested in putting effort into the development of a commercial product. However, all forms of user involvement support the different motives of user involvement discussed in the next section.

2.2 Motives for user involvement

In addition to different forms, user involvement has varied motives. Muller et al. (1997) identify three convergent motives for participatory design: 1) democracy; 2) efficiency, expertise, and quality, and 3) commitment and buy-in. The first motivation, democracy, was an essential part of the original Scandinavian formulation of participatory design. Developing workplace democracy and the development of workers' competence and power to influence their work and their workplaces were the driving forces of the work (Ehn 1993). Muller et al. (1997) suggest that the second type of motivation has emerged from North American practice and that direct participation by end-users is seen in this context as a means of enhancing the process of gathering (and perhaps interpreting) information for system design. The effectiveness of software design and development is improved by including the users' expertise. Efficiency is improved by getting users to collaborate in design and by involving users early in the design process. The quality of the design and the resulting system is improved through better understanding of the users' work and better combination of the diverse and necessary backgrounds brought by various participants. The third motivation, commitment and buy-in, means that a system is more likely to be accepted by 'downstream' end-users if those users are involved in certain 'upstream' formative activities.

Several others have identified similar kinds of motives or, in other words, benefits of user involvement (Kujala, 2003). For example, Damodaran (1996) lists the benefits that effective user involvement has shown to yield in a variety of studies: 1) improved quality of the system arising from more accurate user requirements; 2) avoidance of costly system features that the user did not want or cannot use; 3) improved levels of acceptance of the system; 4) greater understanding of the system by the user resulting in more effective use, and 5) increased participation in decision-making within the organisation.

In summary, the motives for user involvement can be categorised into three partly overlapping classes:

1. Democratic motives that support:

- workers' participation in decision making and power to influence their work
- development of workers' competence and expertise

2. Organisational motives that support:

- acceptance of the system by end-users
- learning and using of the system

3. Practical development-oriented motives that support:

- understanding the users' work
- defining more accurate user requirements
- improving the quality of the system
- improving development efficiency
- increasing user and customer satisfaction

Figure 2 demonstrates the proposed relations of the practical development-oriented benefits of early user involvement (adapted from Kujala, 2003). Early user involvement seems to have a positive value for users and customers as such (Kujala et al., 2001a, Kujala, 2003), but user and customer satisfaction can be considered as improving through better system quality. Requirements quality is an intermediated factor improving user and customer satisfaction in the long run. Requirement quality is better and development work is performed better as the requirements are based on real information gained from users. In addition, users are available for giving feedback and testing design solutions. Thus, the resulting system is more likely to match the needs of the users.

Figure 2. Effects of early user involvement.

3. User involvement in product development contexts

Sawyer (2000) suggests that the major differences between packaged and custom software development can be characterised as stakeholding and schedule constraints. In custom software development, the customer is the principal stakeholder and often bears a large proportion of the risk. Packaged software development, by contrast, is undertaken directly in support of the developer's strategic business objectives. Sawyer (2000) makes the further point that packaged software development typically exploits a market opportunity without a discrete set of users who can articulate their requirements. Thus, it needs to be carefully considered how user involvement is applied in practice in product development contexts.

3.1. Practical motives of user involvement

As Grudin (1991a) describes, in product development the users often remain unknown until the product is marketed. Keil and Carmel (1995) confirmed by their survey that in the package environment over 90% of the

projects relied on marketing and sales for customer input and requirements. In addition, their results show that more successful projects employed more direct links to users and customers than did less successful projects. In spite of Keil and Carmel's (1995) positive results, it is sometimes argued that user involvement is not an appropriate approach in product development contexts. Furthermore, Grudin (1991b) and Poltrock and Grudin (1994) provide a detailed description of the organisational obstacles to direct contacts 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 contacts when established. The findings of Grudin (1991b) rely on an earlier survey and on interviews with over 200 interface designers from several product development companies.

Furthermore, Iivari (2004) analysed user involvement in two product development organizations. She identifies divergent views of the motivation for user involvement. Democratic goals were not mentioned, but user involvement is seen as helping the projects to 'do a quality job', 'get it right the first time', and as a selling argument and image factor for the company in making profits. Iivari (2004) argues that some of the images of user involvement identified might be interpreted as being in stark contrast with the original aims of the Scandinavian participatory tradition.

However, it can be argued that the characteristics of the product development context make the democratic and organisational motives of user involvement described in Chapter 2.2 somewhat inappropriate. For example, Karlsson et al. (2002) argue that requirements are invented by the developers, since there is not a discrete set of users who can articulate their requirements. A large number of users makes it hard to involve users in a democratic way and thus users cannot be committed through participation either. In addition, users may not be motivated to participate in the development work of a commercial product. Users have a large number of products to select from, and on the other hand, the product may be used in many organisations, which are not known before the product is ready.

Nevertheless, all the practical development-oriented motives of user involvement described in Chapter 2.2 are valid for product development. Even if users may not be motivated to participate in the development work, several cases show that, at least to some extent, they are interested in providing information and feedback (Kujala, 2002). Users want good quality products and better support for their tasks, and so appreciate their views being considered. They are experts in domain knowledge, their tasks and work practices, behaviour and preferences, and general context of use. By gathering this information, it is possible to define user requirements more accurately and, by doing so, create better system quality. Furthermore, better developmental performance is based on gathering correct information sufficiently early.

3.2. Understanding user needs

There are several usability methods that are suitable in the product-development context (e.g. ISO/TR 16982, 2002). However, user involvement is most efficient and influential in the early stages of system development, as the costs involved in making changes increase as the development continues (cf. Ehrlich and Rohn, 1994; Noyes et al., 1996). Thus, ISO 13407 (1999) emphasizes the importance of user involvement in understanding and specifying the context of use, which is the basis of specifying user and organizational requirements. The context of use is defined as the users, tasks, equipment, and the physical and social environments in which a product is used. In a similar vein, Gulliksen et al. (2003) describe in their principles for user-centred system design the importance of understanding the context of use in order to create and maintain a focus on user rather than technical needs. Thus, understanding user needs as they are interwoven with context of use is one of the principle goals of user involvement. Users do not need to conceptualise the context of use, but nevertheless this defines the limits and possibilities the users have in achieving their goals. Thus, we define user needs as problems that hinder users in achieving their goals in a specified context of use. Understanding

the context of use is not valuable as such, but we need to understand user needs or, in other words, we need to understand how the future product can support users in achieving their goals in a specified context of use.

4. Field studies as an approach to gathering user needs

As discussed above, understanding user needs is one of the main goals of user involvement. Field studies are an approach focusing on early user involvement and gathering information from users about their needs. Field studies mean that users and their tasks and environments are studied in their actual context using qualitative methods (cf. Wixon et al., 2002). As field studies are qualitative by nature, the number of participating users is typically not large, but the goal is to select representative users and understand in depth their needs and the typical context of use. Hackos and Redish (1998) describe an extensive range of field methods and provide practical advice on conducting field studies.

4.1 Benefits and challenges of field studies

A literature review of field studies reveals many positive experiences (Kujala, 2003). For example, it was felt that more accurate user requirements were gathered and user needs were better reported. In addition, positive customer and user responses were reported. On the other hand, many challenges to the improvement of field study techniques exist, in particular, how to analyse a large amount of data. The information gathered from users is descriptive and informal by nature and the analysis of it is time demanding. The problems were repeatedly reported in the book 'Field Methods Casebook for Software Design' edited by Wixon and Ramey (1996). Field studies are often seen to be time consuming, providing a vast amount of unstructured data that is difficult to use in development (e.g. Bly, 1997; Hynninen et al., 1999). Moreover, fieldworkers have been found to have problems with communicating results to system developers and with effecting design work (Plowman et al., 1995). Even though field studies are considered beneficial, they are not widely used in practice. For example, the results of a recent survey of user-centred practitioners show that the practitioners themselves consider field studies and user-requirement analysis most important, although not as widely used as other approaches (Wredenburg et al., 2002). The main reason is probably the above-mentioned challenges in analysing user needs. These challenges need to be addressed to make field studies sufficiently practical.

4.2 Analysing user needs and discovering relevant issues

The power of field studies is in deeply understanding context of use. However, it is not a trivial task, and context includes a huge amount of details, but only some of them are relevant for development. For example, Greenberg (2001) provides a summary based on several theories and stating that context is a dynamic construct, changing moment by moment, and people interpret it to perform actions.

In addition to the complexity of the context of use, interpreting and analysing context of use for development purposes is challenging. As Olsson (2004) argues, people are often preoccupied with the current work situation and its inherent routines, but are not so concerned with delivering upfront the appropriate design demands. Thus, because users do not have professional skills to define user requirements, the task of analysing user needs and translating them to user requirements is left to developers. For example, Beyer and Holtzblatt (1998) state that design begins with a creative leap from customer data to the implications for design and from implications to ideas for specific features. In addition, they describe using customer data as a new skill. They recommend recognising the overall work situation and envisioning an integrated solution.

As a solution, Contextual Design provides five work models to represent customer work practice (Beyer and Holtzblatt, 1998). The flow model represents the communication and coordination necessary to do the work. The sequence model shows the detailed work steps necessary to achieve an intention. The artefact model

shows the physical objects created to support the work, along with their structure, usage, and intent, culture, or values. The physical model shows the physical structure of the work environment as it affects the work. These models are then consolidated into models characterising the work structure and basic work strategies across all customers.

The idea of producing and consolidating the contextual work models is to gain an understanding of the context of use or the underlying structure of the work. However, the tasks are complicated and time-demanding in product development contexts. On the other hand, the cleverness of Contextual Design is that designing the user interface and its details are delayed as long as possible and it encourages thorough design change that goes beyond user interface changes (Spinuzzi, 2000). As Holzblatt and Beyer (1993) describe it by themselves: 'We could use prototypes, mockups, or sketches to represent system structure. But we find they focus the team on the user interface. They hide the structure of the system behind user interface details, making it easier to talk about menus, icons, work choice, and screen layout than about whether the structure and organisation are right.'

In Contextual Design, the goal is to redesign the underlying structure of work rather than the artefact. In fact, this is reminiscent of the focus of the task analysis approach. For example, Kirwan and Ainsworth (1993) state that task analysis is the name given to any process that identifies and examines the tasks that must be performed by users when they interact with systems. Jeffries (1997) states that the analyst's goal is to understand the tasks well enough to enumerate their steps and choice points, and to constrain the set of task variations to something that is cognitively manageable but also covers the critical functions that the software will support. Jeffries (1997) does not describe such a conscious goal of redesigning tasks, but the typical result of a task analysis is a list of all the tasks people do in this domain. In addition, he gives a list of ways in which task analysis can be useful in a software project, such as identifying the task to be performed by the system and redundant work that can be minimised.

As Diaper (2001) points out, an outstanding methodological issue for task analysis in general is how to analyse information derived using requirements capture techniques. Furthermore, the task analysis processes described are too difficult to use within tight time scales in product development contexts. For example, Diaper argues that as a method, TADK (Task analysis for knowledge descriptions) is too complicated to use with task analyses of any size without software support.

In summary, it is generally agreed that the goal of user involvement is to understand the context of use and user needs. Better user requirements and user interface can then be defined on the basis of that information. Field studies appear a promising approach to understanding user needs. However, analysing and structuring user needs in product development contexts has been shown to be difficult.

5. Research method

Early user involvement was investigated in real product-development contexts by means of case studies in order to understand the conditions of user involvement in natural product-development settings. This study adopts multiple-case research design strategy (Dubé and Paré, 2003; Yin, 1994). Multiple case studies were performed to understand the influence of variability in context and to gain more general research results than single cases (Yin, 1994). First, the cross-case analysis of the first seven case studies was used to develop a process of early user involvement to support the analysis and utilisation of user needs in product development. Second, two case studies were performed to test the developed early user-involvement process. All the nine case studies were performed as a part of three research projects carried out between 1998 and 2005. The studies were based on real product-development cases in six industrial partners in Finland. The cases represent diverse applications and companies, which enhances the generalizability of the results (Eisenhardt, 1989).

The first seven studies were individual case studies in which within-case analysis was used (cf. Eisenhardt, 1989). Most of the case studies were published and the results are summarised in Kujala (2002). The goal was to evaluate the costs and benefits of the field study approach. As shown in Table 1, several data-gathering methods were used in order to increase the internal validity of the results (Dubé and Paré, 2003). It was found that the field study approach is useful even in a short time frame and incurs relatively low costs. For example, usability was improved and customer satisfaction was increased. The number of involved users varied from three to thirteen. As mentioned earlier, field studies are qualitative in nature and the goal is to understand the needs of the users in depth, rather than to cover large sample of users. The rationale of selecting representative users is described in Kujala and Kauppinen (2004). As Eisenhardt (1989) points out, the within-case analysis gives investigators a rich familiarity with each case, which, in turn, accelerates cross-case comparison. Here, the cross-case analysis was used to identify a process of early user involvement in order to support the analysis and utilization of user needs in product development.

The goal of the last two case studies was to test the analysis process in real product-development contexts. The results were analysed in two stages (Eisenhardt, 1989). Within-case analysis was performed first to provide researchers with a rich understanding of each case. Second, a cross-case analysis was performed to identify the similarities and differences between the cases.

6. Cases: Process of analysing user needs

The first seven case studies are summarised in Table 1. The research questions varied as the research work went on and new problems and solutions identified. Table 1 summarizes the lessons learned concerning the analysis and utilization of user needs. In the first two case studies, reported by Kujala and Mäntylä (2000ab), the goal was to describe the results of field studies rather than analyse them. The work models of Holtzblatt and Beyer (1993) were taken as the starting-point for describing the current situation. However, it was recognised that the work models were too complicated to use cost-effectively and thus only a simple picture of the current context of use was produced. After that, the design work quickly turned to describing the use hierarchies of the future product as task sequence was identified as important in defining the new product. In addition, the problems of users were found to be very useful in identifying new product features.

Case	Research question	Application	Data gathering methods	Lessons learned concerning analysing user needs
I	How can field studies be applied in product development?	Personal digital assistant	Participant-observation, interview	The work models are too complicated to use cost-effectively. Task sequences provided useful information.
II	What are the benefits and costs of field studies to early user involvement compared to usability testing?	Portable communications device	Documentation, experiment (replicated product design), interview	The current problems of the users were a useful information source for understanding their needs.
III	How can field studies be introduced to product development cases?	Elevator	Participant-observation, questionnaire, interview	Analysing and reporting was the most time-consuming and demanding phase in the field studies.
IV	How can user needs be represented and translated into user requirements in industrial product development cases?	Elevators and escalators	Participant-observation	User needs need to be described in a more formal way in order to be useful for technical developers. User-need tables were developed to describe user needs in a form that is easy to transform to use cases. User-needs tables helped the developers translate user needs to user requirements.
V	How can user needs be represented and translated into user requirements in industrial product development cases?	Weather measurement device	Participant-observation	
VI	How can user needs be represented and translated into user requirements in industrial product development cases?	Building modeling software package	Participant observation	
VII	How can user needs analyzed and translated into user requirements?	Street register software	Participant observation	User-needs tables helped the developers to structure the user-need information and to transform the user needs to use cases. However, the process of analysing needs would support the work.

Table 1. Summary of case studies.

In the last five case studies, three of them reported in Kujala et al. (2001b), the field studies were introduced and performed in real product-development contexts and by real developers. It became evident that a clearer and more cost-effective process for analysing the results was needed. In the case study reported by Kujala et al. (2001a), the person hours used in performing the field study was carefully calculated and it was found that analysing and reporting was the most time-consuming phase: 23 % of the time was spent in gathering data and 77 % in analysing and reporting. In addition, the team members found this a particularly demanding phase. In the following studies, user-needs tables were developed to bridge the gap between informal user needs and more formal requirements. In addition, we started to work on the process of analysis and making this explicit enough to enable it to be taught and used effectively.

Based on these case studies, we identified main six steps of early user involvement for identifying user needs and analysing them:

1. Identify stakeholders and user groups

First, the most important stakeholders and user groups need to be identified and described so as to make it possible to reach representative customers and users by means of field studies. We have discussed the identification of users in Kujala and Kauppinen (2004).

2. Visit users and explore their needs

Users are visited in their own environment in order to gather real information about their needs.

3. Describe the current situation

The first step in analysing the results of the field study is to describe the current situation. By the current situation, we mean the context of use and particularly the task sequence. In our case studies, we used a simple figure, a task hierarchy, a scenario, and user-needs tables to describe the current situation.

4. Analyse and prioritise the problems and possibilities

In order to utilise the information gathered in product development, the user needs associated with the future product are identified. In case studies, it appeared clear that understanding the problems of the users was key to developing new products for them and that, when the problems were analysed, it was easy to develop new products to correct those problems and support users in their tasks. Thus, we argue that user needs manifest themselves as either problems that hinder users in achieving their goals, or as opportunities to improve the likelihood of users achieving their goals. Not all the problems and possibilities are clearly stated by users, but the positive and negative aspects of the context of use and the present tasks and tools need to be analysed and identified. The negative aspects may be labour-intensive or complicated task steps or sequences, task details difficult to remember or complete, or task steps performed using traditional methods such as paper and pencil. The positive aspects are those which users value and are not willing to change. Finally, there may be too many user needs for one product to be able to meet and thus the identified user needs need to be categorised and prioritised.

In categorising needs, we have used a similar process to that which Bruseberg and McDonagh-Philp (2001) used for initial data analysis of the aspirations that users stated in focus groups. Bruseberg and McDonagh found that making notes of essential issues and comments made by users can be more effective than producing lengthy verbatim notes. They classified similar repeatedly-mentioned topics into categories and substantiated the weight of the categories by giving them a 'tick' each time a related thought was mentioned. In our case studies, we have similar experiences; developers are not willing to use or capable of utilising verbatim descriptions and we developed user-needs tables to summarise and structure the findings.

Although we identified the problems of the users as essential links to product development, it was not always so easy to utilize the information in development work. There were often an overwhelming number of problems, and technically oriented developers had a problem in processing the written descriptions. We realised that developers need a slightly more formal way of representing user needs so that they could process the information step by step. Thus, we combined the two most essential representations of context of use - a task sequence and user problems and possibilities – in user-needs tables. In other words, user-needs tables represent user needs as users' problems and possibilities, and link them to a task sequence. Several kinds of user information can be summarised in the form of user problems and possibilities. Problems are obstacles that arise from users' characteristics, their physical and social environment, and the overall situation. Possibilities represent users' more implicit needs, and suggest how users' tasks can be supported and improved. However, the information is in a structured form and can be used step by step in redesigning the current situation.

5. Redesign the current situation

The first step in utilising user needs and defining user requirements is redesigning the current situation and deciding the role of the product in the context of use. The values of users need to be identified so that it is possible to decide how to add value for them in the future product. It is essential to identify the focus of the product and the scope of the tasks to be supported. Furthermore, the identified problems should be corrected and users' task sequences need to be streamlined. For example, unnecessary steps are eliminated. However, it is important to keep the existing logic as it supports users in performing their tasks. We have used use cases for describing the redesigned current situation, but use scenarios, storyboards, use workflows, or use flow diagrams could also be used (Hackos and Redish, 1998).

6. Define user requirements

In addition to redesigning the current situation, specific user requirements were also needed.

7. Cases: Testing the analysis process

The goal of the case studies was to test the analysis process in real product development contexts. The cases and the results of the within-case analysis are described first and then the comparison of the cases is presented.

7.1 Case 1

The first study was performed in Tekla, a medium-sized company producing software products for managing infrastructures. The main product, Tekla Structures, is a complex system for experts in structural engineering and it includes an expansive number of functions. The complexity of Tekla Structures is demonstrated by the fact that there are 220 commands on the first menu level, and, in addition, there are dozens of commands on successive menu levels.

The goal of the case study was to gather user needs for the redevelopment of a functionality of Tekla Structures. The product had been developed over the years according to customer feedback. Most of the corrections were rather small fixes. Now, a larger piece of functionality was identified as being difficult for the customers, according to the feedback received by area offices abroad. This functionality was selected for improvement, but the needs of the users were not well enough known and more information was needed from them.

Field studies

The field study was performed by a team of two usability experts, a user interface designer, and a researcher. The researcher's role was that of an expert or a consultant who provided information, brief training, and support for the practitioners. In addition, a requirements engineering team of a requirements engineer, a usability expert, and a researcher participated in planning the field study and analysing the results.

In the beginning, existing information was gathered inside the company, for example from service. This information helped to focus the user interviews and was also of use in the selection of questions. Then, four users representing three customers were interviewed in order to ascertain user needs for the product and to gather feedback on the previous version of the product. One person conducted the interviews and another took notes. In addition, the interviews were audio recorded.

Analysing the results of the field studies

All three customers had developed their own strategy for handling the task and the problems associated with the task. This strategy was rather similar in all of them, even though the product did not introduce or support it. A number of user problems associated with the current task sequence were identified. It was realised that many of these problems will disappear when the functionality and user interface are changed. However, it was important to identify the essential task steps, phases, and problems for basing design work on user needs. Thus, we decided to have a higher-level view of the current task sequences and problems and it was described in a user-needs table (see Table 2).

Step	Problems	Possible solution
1. User creates a seed project (a project containing default environment and company settings).	Currently the seed project is a complicated many-level folder structure containing a complete set of Tekla Structures files, only the model excluded.	File/New opens dialog asking whether user wants to create 1) new seed project, 2) new model with seed project as template....
2. User saves a seed project.		Name/path asked, automatic save

Table 2. Example of the kind of user-needs table developed.

In addition to original use sequence (Step column) and problems and possibilities (Problems column), the requirements engineer added a possible solution column to describe initial solution ideas. This column was useful for getting an idea of how the problems could be solved in the new version of the product.

After that the high-level task sequence described in the user-needs table was redesigned and the new task sequence was described in a use case (see Table 3). The requirement engineer used that table as a starting point and continued to define user requirements. The requirements engineering group also used user-needs tables for gathering feedback both from inside the company and users. The final requirements specification described the user problems and requirements on different levels: the current context of use, problems in the current user interface, requirements gathered inside the house by means of an a requirements management tool, users' comments, use cases, a navigation map, and pictures of the new user interface.

Step	User input	Program action	Note/ question
1.	User indicates that he wants to save the model as a model template.	Tekla Structures asks the user to enter a name and location for the new model template.	Note: File->Save as...
2.	User enters name and location.	Tekla Structures saves the model as a model template to this location.	

Table 3. Example of use case developed.

Evaluating the analysis process

The analysis process was evaluated by interviews and a questionnaire. The requirements engineer and two developers were interviewed and seven persons filled in the questionnaire.

The persons who were working for the project were asked to fill in the questionnaire and three of them were also interviewed. They represented software production, usability group, user documentation, service, product management, and release management. Seven persons filled in the questionnaire, two from software production, two from the usability group, one from user documentation, one from product management, and one requirements engineer who was responsible for writing the specification.

The questionnaire used is in Appendix A. First, the respondents were asked to evaluate the requirements quality in the developed specification and then they were asked to evaluate the usefulness of the descriptions used. Finally, they were asked to evaluate the general requirements quality in Tekla.

The requirements team found the analysis process and the piloted descriptions useful; the descriptions supported analysing in stages and parsing a large amount of information. Different description methods complemented each other in step-by-step analysis and all the steps were important and had their own roles. In addition, documenting the essential issues helped in the structuring and analysis of the large amount of information gathered from users.

Other respondents also found all the other descriptions useful, except the requirements gathered by the requirements management tool. The reason may be that the requirements were preliminary and inconsistent as

they were written by different people. However, the views expressed towards the descriptions seemed to depend on the role the respondent had. Particularly, technical developers were most interested in user interface pictures and a navigation map. They said that they would like to have all the background information available in another separate document. Another solution would be to have several views incorporated in the same document.

Furthermore, the quality of the requirements was evaluated as being better in the pilot project than in Tekla in general, as shown in Figure 3. One person refused to evaluate the general quality as she was a rather new employee of Tekla and thus her evaluation was excluded from Figure 3. In summary, one person out of the five evaluated the quality as being slightly lower in the project than in general and the other four evaluated the quality as being better in the pilot project.

Figure 3. Evaluated requirements quality in the pilot project and the company in general (Mean value of the quality statements, 1=strongly disagree, 4=strongly agree).

Figure 4 shows how the different requirements quality issues were evaluated and it is possible to see how the pilot project differs. The specification of the pilot project was seen as being more successful, complete, and understandable than specifications in general. In addition, the respondents evaluated the specifications as being based on information gained from users and customers.

Figure 4. Respondents' mean assessment of the requirements quality issues.

7.2 Case 2

The second study was performed in F-Secure, a medium-sized company producing software security products. The goal of the study was being pilot field studies in one project, in which a new product was being developed for a new customer group. As the target customer group was new and relatively unknown, the idea was to validate the product idea, understand hidden user needs, and gather user requirements. Unlike the product in Case 1, the application was not used on a daily basis but it can be characterised as working in the background on a user's computer.

Field studies

First, the user study team consisted of a project manager, a chief architect, and a researcher. The researcher's role was that of an expert or a consultant who provided information, brief training, and support for the practitioners. The chief architect interviewed three users with the researcher. The results showed that the user needs depended on the customer type and their risk level. The business owner of the new product was now involved and she decided that more information was needed from users. This time, the user study team consisted of the business owner, a user interface designer, documenting team manager, and a researcher. The team interviewed 14 more users during a period of one month.

Analysing the results of field studies

The time schedule was very tight; there was no time for further analysis of the results of the first three interviews and only short notes were produced. After the rest of the interviews, the results were gathered onto 18 slides. In addition, the researcher produced a user-needs table describing the results. In this case, the description of the two main customer groups and the values of the users were identified as being the most essential results. It was found that the customers consisted of two distinctive groups that had different motives and needs for the product.

Evaluating the analysis process

In this case, the analysis process was much more inaccurate. Short notes were written for each interview and short textual descriptions of the current context of use were pulled together from the individual notes. As the product was intended to work mostly in the background on the computer and it reminded one of a service rather than a software product, redesigning the task sequence was not as essential as in Case 1. Thus, the analysis focused on describing the present context of use for evaluating the new product concept. However, as users' skill level was notably lower than expected and they valued different things than expected, the business owner evaluated field study results as having a significant effect on the scope of the product.

7.3 Comparison of Case 1 and Case 2

The results of the Case 1 and Case 2 are summarised in Table 4. The two cases were representing diverse categories of applications and the results were accordingly different. Case 1 was in line with our earlier case studies, showing the validity of earlier findings and the developed process of early user involvement. It showed that the process helped in structuring and analysing the large amount of information that led from user needs to requirements. As a result, the requirements were evaluated as being more successful and their quality was better than average. In Case 2, the results of the field study was found useful, but the user-need analysis part of the process was not used. The main reason for not using the process was that the application did not engage so many user tasks to be described and redesigned. In this case, the benefit of field studies was derived more from understanding user motivation and values and enabled the correct product concept to be defined from the user-point of view.

Case	Research question	Application	Result
I	How can field study results be analysed and translated into user requirements?	Building modelling software package	The analysis process helped in structuring and analysing the large amount of information. The resulting requirements were evaluated as being more successful and their quality was better than average.
II	How can field study results be analysed and translated into user requirements?	Software security product	The analysis of user tasks was unnecessary as the phase of development was very early and the product is working in the background on a user's computer. Thus, identifying the context of use and user values were evaluated to be more important than analysing task sequence.

Table 4. Summary of the results of Case 1 and Case 2.

These two cases demonstrate how different applications and their development can be. No single user-involvement process fits every case. Case 1 showed the proposed process of early user involvement to be useful. Case 2 revealed that the prerequisite is that the application supports users' tasks that can be identified. Furthermore, Case 2 showed that informed redesign of user tasks is only one aspect of the benefits of the field studies and understanding user motivation is an essential benefit, too.

8. Conclusions

In this article, we have discussed the role of user involvement in product development. Our suggestion is that the most natural and promising form of user involvement in product development is informative and empowered. This means that even if products are developed for a wide audience of users (cf. Tuunanen, 2003) and they may not be motivated to play an active role in product development, it is essential for product developers to be active in gathering information and feedback directly from representative users and understanding their needs and values. Users should not be passive informants as, in spite of the good intentions of the developers, they have different values concerning products and their use.

The most significant user involvement occurs at the beginning of product development, when it is being decided what the product will be like and how it is going to support users. It is essential that product development is grounded on user needs interwoven with context of use. Field studies appear a promising approach to early user involvement, but they are found demanding and not widely used in practice. One of the critical concerns in making field studies effective is how to analyse user needs and discover relevant issues from product-development point of view. This article presents a process of early user involvement that describes how user needs can be analysed and utilised in product development. The process was developed on the basis of the experience gained from seven case studies.

The developed process of early user involvement was also tested in two case studies. The results of Case 1 confirm the insight gained from the earlier case studies, that the process supports early user involvement and analysis of user needs. It was showed that the process helped in structuring and analysing the large amount of information that led from user needs to requirements. Documenting essential information at each step supported analysing in stages and parsing a large amount of information. In addition, the requirements were evaluated as being more successful and their quality was better than average in the company. Thus, effective user involvement was achieved by improving the analysis of user needs.

On the other hand, the process was not used in Case 2. As can be expected, no single user-involvement process fits the development of varied applications in varied contexts. Case 2 demonstrates that the prerequisite of using the proposed process is that the application supports users' tasks that can be identified. Furthermore, it shows that informed redesign of user task is only one aspect of the benefits of field studies; understanding user motivation can be even more essential. It can be argued that, if the product idea does not

involve user values and motivations, it is not acceptable at all and the redesigned task sequence is not useful either. On the other hand, understanding user values and motivations and task sequence are not exclusionary, but, rather, early user involvement supports both issues. For example, according to our experiences, the details of task sequence are often very meaningful to users and problems in task sequence can be annoying to them.

Thus, the process of early user involvement needs to be adapted on the basis of the application type and development situation. Furthermore, the presented analysis process is initial and simplified. For example, including collaboration and multitasks would bring a new dimension to analysis that is not considered here (see Pinelle et al., 2003). However, the goal of the process presented is not to be broad and extensive but to cover the most essential analysis steps for tightly scheduled product development projects. Thus, the process of early user involvement needs to be simple enough to be practical in product development.

The results of the case studies suggest some explanations of the mechanisms of early user involvement in product development. First, users can provide information about the context of use that has an effect on the use of the future product. Second, providing information is not the only mechanism, but placing developers near users increases their understanding of the users' values and attitudes. This interpretation is also supported by the earlier observations of Olsson (2004). In addition, Kujala and Mäntylä (2000a) interviewed developers who happen to be users themselves and found that the developers incorrectly expect ordinary users to have similar patterns of behaviours and values to themselves. However, understanding users' needs and values is only the first step in user involvement, and user-centred development is iterative in nature. As soon as a new product is sketched, it creates new user needs and requirements as the task-artifact cycle predicts (Carroll and Rosson, 1992).

The importance of understanding context of use is already widely recognized and is recommended by ISO 13407 (1999) and Gulliksen et al. (2003). However, in order to make the utilization of the context of use information possible, the links to development work need to be elaborated. The results of the case studies suggest some ways of utilizing the context of use information. In particular, present task sequence, problems, and possibilities are shown to have direct consequences for user requirements and finally for product quality.

In addition, the popularity of different forms of scenarios demonstrates that task sequences and the context of use related to it are widely recognized as useful in system development (see, for example, Carroll and Rosson, 1992; Alexander and Maiden, 2004; Redish and Wixon, 2003). Certainly, different ways of describing task sequences can be useful and valid in analysing user needs. The user-needs table is one of the possible means of presentation; its strength lies in its summary of the context of use in one table in a structured way. Task scenarios that are narrative descriptions of a user performing a task would include the same information, but they are not so straightforward to use in development work. In addition, as Carroll and Rosson (1992) point out, there is an infinity of possible use-scenarios. A user-needs table structures the information so that it can be handled step by step and so the problems and possibilities highlight clearly where the current situation could be improved by a new system.

Overall, this article introduces an initial step towards understanding early user involvement in product development and how to improve its effectiveness. The article provides a process of identifying, analysing, and utilising user needs based on broad experiences from several industrial cases. Furthermore, it analyses a potential mechanism of user involvement.

Appendix A: The evaluation questionnaire

Tekla pilot project questionnaire 2005

This questionnaire is designed to gather experiences from the pilot project for Tekla product development purposes. Specification and requirements refer here to the pilot project start up "Specification of Feature" document. The questionnaire is a part of the CORE research project:

<http://www.soberit.hut.fi/core/english/index.html>

All the information you provide will be kept confidential. The resulting report will present information in aggregate form and such information will be anonymous and unattributable to individual respondents.

The quality of the specification

Please indicate the strength of your agreement/disagreement with the following statements concerning the quality of pilot project specification.

1 = strongly disagree, 2 = disagree, 3 = agree, 4 = strongly agree, 5 = no opinion, don't know

	Disagree		Agree		Don't know
	(1)	(2)	(3)	(4)	
I consider the specification a success.	1	2	3	4	[]
Customer or user requirements are completely defined.	1	2	3	4	[]
All the essential issues from the customer and user point of view are documented.	1	2	3	4	[]
The document describes functions that meet the user needs.	1	2	3	4	[]
The customer and user processes that the system is designed to support are analyzed and documented.	1	2	3	4	[]
The requirements are understandable for those who are not part of the project.	1	2	3	4	[]
The requirements are described using language that a technical person understands.	1	2	3	4	[]
The requirements are described using language that a non-technical person understands.	1	2	3	4	[]
The requirements are based on the information gained from users or customers.	1	2	3	4	[]
In all likelihood, there are moderately few errors in the requirements.	1	2	3	4	[]
The correctness of the requirements is checked with real users.	1	2	3	4	[]
There were many differences in the views of the parties in requirements engineering.	1	2	3	4	[]

The analysis of requirements

The requirements were analyzed and described using several methods, please; indicate how useful you evaluate them?

	Useless		Useful		don't know
	(1)	(2)	(3)	(4)	
Current situation tables	1	2	3	4	[]
Problems in current user interface pictures	1	2	3	4	[]
Requirements in Caliber	1	2	3	4	[]
Comments from users	1	2	3	4	[]
Use cases	1	2	3	4	[]
Navigation map	1	2	3	4	[]
User interface pictures	1	2	3	4	[]

What is good in varied descriptions?

What is problematic in varied descriptions?

What would you recommend? How should requirements be analyzed and described in Tekla?

Other comments:

The average quality of the specifications in Tekla

Please indicate the strength of your agreement/disagreement with the following statements concerning the average quality of the specifications in Tekla.

1 = strongly disagree, 2 = disagree, 3 = agree, 4 = strongly agree, 5 = no opinion, don't know

	Disagree			Agree	don't know
	(1)	(2)	(3)	(4)	
I consider the specification a success.	1	2	3	4	[]
Customer or user requirements are completely defined.	1	2	3	4	[]
All the essential issues from the customer and user point of view are documented.	1	2	3	4	[]
The document describes functions that meet the user needs.	1	2	3	4	[]
The customer and user processes that the system is designed to support are analyzed and documented.	1	2	3	4	[]
The requirements are understandable for those who are not part of the project.	1	2	3	4	[]
The requirements are described using language that a technical person understands.	1	2	3	4	[]
The requirements are described using language that a non-technical person understands.	1	2	3	4	[]
The requirements are based on the information gained from users or customers.	1	2	3	4	[]
In all likelihood, there are moderately few errors in the requirements.	1	2	3	4	[]
The correctness of the requirements is checked with real users.	1	2	3	4	[]
There were many differences in the views of the parties in requirements engineering.	1	2	3	4	[]

Thank you very much!

References

- Alexander, I. And Maiden, N. (eds.) (2004). *Scenarios, Stories, Use Cases: Through the System Development Life-Cycle*. 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. (1998). *Contextual Design: Defining Customer-Centered Systems*. San Francisco: Morgan Kaufmann Publishers.
- Bly, S. (1997). Field work: Is it product work? *Interactions*, 4, 1, 25-30.
- Bruseberg, A. and McDonagh-Philp, D. (2001). New product development by eliciting user experience and aspirations. *International Journal of Human-Computer Studies*, 55, 435-452.
- Carmel, E. and Becker, S. (1995). A process model for packaged software development. *IEEE Transactions on Engineering Management*, 42, 1, 50-61.
- Carroll, J.M. and Rosson, M.B. (1992). Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems*, 10, 2, 181-212.
- Damodaran, L. (1996). User involvement in the systems design process—a practical guide for users. *Behaviour & Information Technology*, 15, 6, pp. 363-377.
- Diaper, D. (2001). Task analysis for knowledge descriptions (TADK): a requiem for a method. *Behaviour & Information Technology*, 20, 3, 199-212.
- Dubé and Paré, G. (2003). Rigor in information systems positivist case research: Current practices, trends, and recommendations. *MIS Quarterly*, 27, 4, 597-636.
- Ehn, P., 1993, Scandinavian design: On participation and skill. In *Participatory Design: Principles and Practices*, edited by D. Schuler and A. Namioka (Hillsdale, NJ: Lawrence Erlbaum), pp. 41-77.
- 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.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, 14, 4, 532-550.
- Flanagan, M., Howe, D.C. and Nissenbaum, H. (2005). Values at play: Design tradeoffs in socially-oriented game design. In *Proceedings of CHI 2005 conference*, ACM, pp. 751-760.
- Friedman, B. (1996). Value-sensitive design. *Interaction*, 3, 6, 17-23.

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

Greenberg, S. (2001). Context as a dynamic construct. *Human-Computer Interaction*, 16, 257-268.

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, 1991, 147-196.

Grudin, J. and Pruitt, J. (2002). "Personas, participatory design and product development: An infrastructure for engagement". In Proceedings of the *Participatory Design Conference*, pp. 144-161.

Gulliksen, J., Göransson, B., Boivie, I., Blonkvist, S., Persson, J., and Cajander, Å. (2003). Key principles for user-centred system design. *Behaviour and Information Technology*, 22, 6, 397-409.

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

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

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.

Iivari, N. (2004). Enculturation of User Involvement in Software Development Organizations – An Interpretive Case Study in the Product Development Context. In Proceedings of the *NordiCHI '04 Conference*, pp. 287-296.

Iivari, J. and Iivari, N. (2006). Varieties of User-Centeredness. In Proceedings of the 39th *Hawaii International Conference on System Sciences*, IEEE, p. 176a.

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

ISO/TR 16982 (2002). *Ergonomics of human-system interaction - Usability methods supporting human centred design*. Geneva, ISO, 44 p.

Jeffries, R. (1997). The role of task analysis in the design of software. In M. Helander, T. K. Landauer and P. Prabhu (eds.) *Handbook of Human-Computer Interaction*, 2nd ed. Amsterdam: Elsevier, pp. 347-359.

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

Karlsson, L., Dahlstedt, Å.G., Natt och Dag, J., Regnell, B. and Persson, A, "Challenges in Market-Driven Requirements Engineering – an Industrial Interview Study", In Proceedings of Eight *International Workshop on Requirements Engineering: Foundation for Software Quality*, Essen, Germany, 2002, pp. 37-49.

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

Kirwan, B., and Ainsworth, L. K. (eds), 1993, *A guide to task analysis* (London: Taylor & Francis).

Kuhn, S., and Muller, M. J., 1993, Participatory design. *Communications of the ACM*, 36, 6, 24-18.

Kujala, S. (2002). User Studies: A Practical Approach to User Involvement for Gathering User Needs and Requirements. *Acta Polytechnica Scandinavica, Mathematics and Computing Series* No. 116. Espoo: the Finnish Academies of Technology.

Kujala, S. (2003). User involvement: a review of the benefits and challenges. *Behavior & Information Technology*, 22, 1, 1-16.

Kujala, S. and Kauppinen, M. (2004). Identifying and Selecting Users for User-Centered Design. In Proceedings of *Nordic Conference on Computer-Human Interaction*, ACM, 297-303.

Kujala, S., Kauppinen, M., and Rekola, S. (2001a). 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.

Kujala, S., Kauppinen, M., and Rekola, S. (2001b). 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.

Kujala, S., Kauppinen, M. Lehtola, L., Kojo, T. (2005). The Role of User Involvement in Requirements Quality and Project Success. In Proceedings of IEEE *International Requirements Engineering Conference* (RE'05), IEEE Computer Society Press, pp. 75-84.

Kujala, S. & Mäntylä, M. (2000a). How Effective Are User Studies? In Proceedings of *HCI'2000 Conference* (Sunderland, UK, 6-8 September), pp. 61-71.

Kujala, S. & Mäntylä, M. (2000b). Studying Users for Developing Usable and Useful Products. Proceedings of 1st *Nordic Conference on Computer-Human Interaction* (Stockholm, Sweden, 23-25 October), pp. 1-11.

Muller, M. J., Halletwell Haslwanter, J., and Dayton, T. (1997). Participatory practices in the software lifecycle. In *Handbook of Human-Computer Interaction*, edited by M. Helander, T.K. Landauer, and P. Prabhu (Amsterdam: Elsevier), Second edition, pp. 255-297.

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.

Olsson, E. (2004). What active users and designers contribute in the design process. *Interacting with Computers*, 16, 377-401.

Pinelle, D., Gutwin, C. and Greenberg, S. (2003). Task analysis for groupware usability evaluation: Modeling shared-workspace tasks with the mechanics of collaboration. *ACM Transactions on Computer-Human Interaction*, 10, 4, 281-311.

Plowman, R., Rogers, Y., Ramage, M. (1995). What are workplace studies for? In Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work. Stockholm, Sweden: Kluwer, pp.309-324.

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.

Rauterberg, M., Strohm, O., Kirsch, C. (1995). Benefits of user-oriented software development based on an iterative cyclic process model for simultaneous engineering. *International Journal of Industrial Ergonomics*, 16, 391-410.

Redish, J. and Wixon, J. Task analysis. In J. A. Jacko and A. Sears (Eds.) *The human-computer interaction handbook: fundamentals, evolving technologies, and emerging applications*. Lawrence Erlbaum Associates, Mahwah, N.J., 2003, pp. 922-940.

Sawyer, P. (2000). Packaged software: Challenges for RE. In Proceedings of the sixth *Int. Workshop on Requirements Engineering: Foundations of Software Quality*, Stockholm, Sweden, pp. 137-142.

Spinuzzi, C. (2000). Investigating the technology-work relationship: a critical comparison of three qualitative field methods. In Proceedings of *IEEE international professional communication conference* and Proceedings of the 18th annual *AMC international conference on computer documentation: technology & teamwork*, Cambridge, Massachusetts, pp. 419-432.

Tuunanen, T. (2003). A New Perspective on Requirements Elicitation Methods. *Journal of Information Technology Theory & Application (JITTA)*, 5, 3, 45-62.

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

Wixon, D., and Ramey, J. (eds.) (1996). *Field Methods Casebook for Software Design*. New York: Wiley.

Wixon, D. R., Ramey, J., Holtzblatt, K, Beyer, H., Hackos, J., Rosenbaum, S. and Page, C, Laakso, S.A. and Laakso, K. (2002). Usability in practice: Field methods evolution and revolution. In Proceedings of the *Extended Abstracts of CHI'2002* (Minneapolis, April 2002), ACM Press, 880-884.

Vredenburg, K. Mao, J.Y. Smith, P.W. and Carey, T. (2002). “A Survey of User-Centered Design Practice”, *Proceedings of the Conference on Human Factors in Computing Systems*, CHI Letters 4(1), pp. 471-478.

Yin, R.K. (1994). *Case Study Research, Design and Methods* (2nd ed.). Beverly Hills, CA: Sage Publications.