7 LINKING USER NEEDS AND USE CASE-DRIVEN REQUIREMENTS ENGINEERING

Sari Kujala

Software Business and Engineering Institute, Helsinki University of Technology, Finland

Abstract

Requirements engineering is the first and the most critical step in software development. One of the basic questions in requirements engineering is how to find out what customers and users really need. In addition, user needs must be expressed by structured, formal user requirements. Use cases are often seen as supporting the process of capturing requirements from the user's point of view. In addition, there is increasing evidence that involving users as the main source of information in requirements engineering is a vital prerequisite in successful projects. The human-computer interaction community has developed a variety of methods for understanding the context of use and eliciting user needs directly from the users themselves. The challenge has been to bridge the gap between informal user need descriptions and formal user requirements. This chapter presents an approach that shows how user-centered requirements analysis can be effectively integrated to use case-driven requirements engineering. Firstly, user needs are gathered directly from users using semi-structured, small-scale field studies. Secondly, the results are summarized in user need tables to ease their utilization and their linking to use case descriptions. Thirdly, the user need tables are transformed into use case descriptions. The approach has been validated by several industrial cases in real development contexts.

A. Seffah (eds.), Human-Centered Software Engineering – Integrating Usability in the Development Process, 113–125.
© 2005 *Springer. Printed in the Netherlands.*

7.1 INTRODUCTION

Requirements engineering is the first and the most critical step in software development. One of the basic questions in requirements engineering is what is it that customers and users really need. The success of the product depends on its ability to provide the right solution for the customers and the users. The internal functioning of the system is not really of great concern to the users and customers, but they do want to perform their tasks with the system in a specific way.

Requirements engineering can also be identified as an essential activity from the usability point of view. As John Karat (Karat, 1997) writes, the acceptability of any software product is no longer seen as being dependent solely on user interface features, but on the way a system fits into its context of use. Thus, user-centered design and user interface design cannot be separated from the rest of the system development. Usability should be considered from the very beginning of the development when a "ground plan" of the system is decided.

There is increasing evidence that involving users as the main source of information in requirements engineering is a vital prerequisite in successful projects (Kujala, 2003). However, user involvement should not be viewed as being trivial. Requirements elicitation is often seen as being a difficult problem, as communication between developers and users may be poor. Developers may not even be motivated to communicate with users, as the developers do not know what it is that they should be asking. In addition, users may not know what they want and may also have difficulties in articulating their needs. One difficulty is that part of the users' knowledge has become tacit through automation (Mitchell and Chi, 1984; Wood, 1997). In well-learned tasks, much of the relevant knowledge is no longer consciously available for the individual and non-verbal skills and everyday self-evidences are difficult to articulate.

The human-computer interaction community has developed a variety of approaches and methods for involving users. User-centered design, participatory design, ethnography, and contextual design may be considered the main approaches, although the roots and methods of these approaches are closely linked and overlapping (Kujala, 2003). In addition, task analysis covers a wide range of methods in order to analyze a system function in terms of user goals and the sub-goals inherent in performing the task (Johnson, 1989; Kirwan and Ainsworth, 1993; Hackos and Redish, 1998). Much of the task analysis literature is devoted to the analysis of data, but task analysis also involves the users as informants (Jeffries, 1997).

Moreover, field studies are particularly focused on discovering tacit knowledge from users. Field studies provide a collection of techniques for studying users, their tasks, and their environments in the actual context of those environments (Wixon et al., 2002). Hackos and Redish (Hackos and Redish, 1998) describe an extensive range of field methods from observing to ethnographic interviewing. Field studies can be seen as overlapping many approaches, but the general idea is not just to ask what users want but to study their actual behavior and context of use.

Thus, field study results help to understand those tacit user needs that users cannot articulate directly. The new system is not used in a vacuum; users have needs relating to the new system depending on the context of use. For example, user needs man-

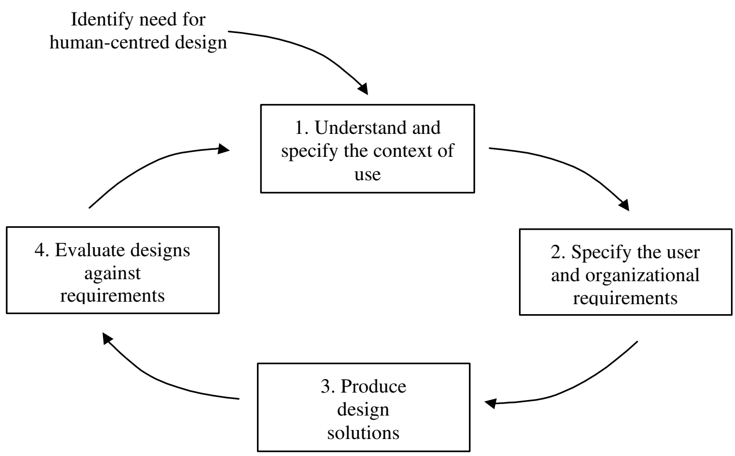

Figure 7.1 The human-centered design activities in ISO 13407 (ISO/IEC, 1999)

ifest themselves as either problems that hinder users in achieving their goals, or as opportunities to improve the likelihood of users achieving their goals.

Field studies support the first activity of human-centered design: "understand and specify the context of use" as described in the international standard ISO 13407 (ISO/IEC, 1999). Figure 7.1 shows the interdependence between human-centered design activities. According to ISO 13407, the characteristics of the users, tasks and the organizational and physical environments define the context in which the system is used; users are seen as a valuable source of this knowledge. Furthermore, user involvement and participation is seen to increase user acceptance and commitment. Understanding and specifying the context of use also helps in identifying relevant usability goals and test cases.

Understanding context of use and discovering user needs is, however, not in itself enough. Analyzing an overwhelming amount of raw data is a frequently mentioned problem of qualitative field studies; this was brought out in Kujala's (Kujala, 2003) literature review. Moreover, fieldworkers have been found to have problems with communicating results to system developers and with effecting design work (Plowman et al., 1995).

In user-centered design, a context of use description is a starting point for the user and their organizational requirements (ISO/IEC, 1999). However, the context of use description is separate from user requirements and these two documents should be linked to facilitate the information flow and the transition from activity one to activity two in Figure 7.1.

Use cases are often seen as supporting the process of capturing requirements from the user's point of view (e.g. Rumbaugh, 1994). However, in our industrial cases, use cases were often written by software engineers who had not met the users, so the use case documents were not shown to the users as Jacobson (Jacobson, 1992) recommends. The software engineers were therefore not familiar with user needs.

As their use cases described the internal functioning of the system and the technical details, they were nearly impossible for the users to understand.

On the other hand, use cases are widely accepted among developers and they provide an opportunity to transmit user needs to requirements engineering and many researchers have already presented their ideas of reconciling user-centered design (e.g. Seffah et al., 2001) or user interface design (e.g. Constantine and Lockwood, 2001) to use case-driven requirements engineering.

This book chapter presents an approach to how user-centered requirements analysis can be effectively integrated to use case-driven requirements engineering. Firstly, user needs are gathered directly from users using semi-structured, small-scale field studies. Secondly, the results are summarized in user need tables to ease their utilization and their linking to use case descriptions. Thirdly, the user need tables are transformed into use case descriptions. Finally, we describe how the approach has been validated by several industrial cases in real development contexts.

7.2 UNDERSTANDING USER NEEDS

As previously stated, field studies provide a way of understanding tacit user needs. Field study techniques go beyond gathering just verbal data by incorporating observations made in the user's environment (Wixon et al., 1990). At the same time 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).

In our experience gained from several industrial cases, field study methods can be very useful even when the investment is modest. The field studies need to be simple and cost-efficient enough to be practical in real-life development projects characterized by tight schedules. In our approach, efficiency is gained by combining simple basic methods and using a top-down approach to focus the study.

Field studies are new for many companies and have to be introduced for the first time. An effective strategy seems to be to use small-scale pilots to introduce simple and easy to learn field study methods (Kujala et al., 2003). In addition, guidelines, checklists, training and personal support facilitate the adoption of new methods.

A good strategy is to start with basic interviewing and observing, as described by Hackos and Redish (Hackos and Redish, 1998), Redish and Wixon (Redish and Wixon, 2003) and Wood (Wood, 1996; Wood, 1997). Interviewing may not be the best method for eliciting non-verbal, tacit information, but it is very cost-effective in understanding the high-level context of use and the users' main goals and problems. Interviewing is easy-to-learn and important information can be discovered directly and effectively from users in a short time. However, interviews should be carried out in the natural setting of the potential users and using their own task-related language. The natural setting helps the user to remember details by seeing and maybe showing and trying the tools and artefacts being discussed. In addition, observing or talking-aloud supports interviewing by providing non-verbal information (Kujala and Mäntylä, 2000b). Beyer and Holzblatt (Beyer and Holtzblatt, 1998) offer good basic principles for facilitating the interviewer-interviewee relationship.

In order to be cost-efficient field studies need to be focused. A study team sets objectives for the study and identifies the most critical themes for each study. Sometimes it may be difficult to find the critical themes if the team does not know the users' world. Wood (Wood, 1997) describes how Grand Tour questions can be used to encourage the user to verbally "show the analyst around" the physical, temporal, and conceptual space of the work domain. In this way, a high-level picture of the users' world is gained and this information can be used to guide the rest of the interview.

In addition, a top-down approach helps in identifying interviewing themes and keeping the amount of data at a manageable level (Kujala and Mäntylä, 2000b). In the top-down approach, certain details of understanding may be lost as it does not start from scratch; however it is easier to learn and an overwhelming amount of raw data is avoided. The top-down approach means that we use semi-structured interviewing in which the most important interviewing themes are predefined and used in preparing questions. The goal is to gather critical information from each topic and keep the topics in mind while observing users and their environment. A basic set of top-down interviewing topics are shown in Table 7.1. The idea is not to follow the prepared questions strictly, but to use them as a checklist and to try to understand the users' perspective.

7.3 LINKING USER NEEDS TO USER REQUIREMENTS

Understanding user needs is in itself not enough. 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 analyzed, prioritized, and described. Finally, informal user needs must be expressed by structured user requirements if they are to be useful to system developers.

Contextual information is often represented in a textual form, such as stories (Imaz and Benyon, 1999). In our first industrial cases, we also used written reports with figures, photographs and video recordings (Kujala and Mäntylä, 2000a; Kujala and Mäntylä, 2000b; Kujala et al., 2001b; Kujala et al., 2001a). Developers evaluated the reports, photographs and videos as useful. However, in one company, we found that it was not so easy for a technically oriented developer to use written descriptions in product development (Kujala et al., 2001a). He could not see how to use the documents in user requirements definition, even though he had written the documents himself. Thus, we realized that a slightly more formal way of representing user needs was needed, so that developers could use information in analyzing and rationally selecting a good combination of user needs for inclusion in their future systems.

We therefore developed user need tables to offer a link between context of use descriptions and structural user requirements (Kujala et al., 2001a), see Table 7.2. The technically oriented developer derived insights from a user need table which we created for him. He got enthusiastic and wanted to make such tables from all of his field study findings. His project manager assessed that he could describe 70% of the preliminary requirements of the project using the user need tables.

User need tables represent user needs as users' problems and also as possibilities, and link them to a task sequence which is an essential part of the context of use (Table 7.2).

Several kinds of user information can be summarized in the form of user problems and possibilities. Problems are obstacles that arise from users' characteristics,

Table 7.1 Interview topics

Topic	Description
Background information	Background information helps the analyst to interpret the results and classify users. Typical questions are about personal characteristics such as age, sex, profession, technical orientation, previous computer and work experience. In addition users' task related characteristics such as motivation, work role and frequency of use or geographic and social characteristics such as location, culture and social connections may be asked.
Users' goals and preferences	The goal is to understand what users want to achieve, what is important for them, and how an intended application can support their tasks and create better ways of achieving the goals.
Users' knowledge, skills and experiences	The goal is to discover what users can and cannot do, and how they employ objects and symbols in accomplishing their goals. Thus, it would be possible to utilize their existing knowledge, skills, and conceptual models in product development.
Current processes	Understanding current processes helps in identifying task hierarchies and task sequences that are natural for users, and gives timing and other benchmarks for the performance criteria of a future solution.
Context of use	It includes user characteristics, tasks, equipment, and a physical and social environment in which a product is used (ISO/IEC, 1999).
Pros and cons of current processes and tools	In redesigning the current process it is necessary to identify advantages that users are unwilling to give up. An intended system should include most of the benefits and solve the current problems.

Table 7.2 An example of a user need table

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 center 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 center operator does not notice the emergency alarm call.
Step 3: Service center operator completes transmission of information to the system and sends it to the area serviceman.	■ Problem: Laborious phase for the service center 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 Center to the elevator.
Step 4: Service center operator calls the serviceman and reads the description of the failure.	■ Problem: Extra work for the service center operator.

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.

In addition to a task sequence and problems and possibilities, a high priority column can also be added to the table, so that it becomes possible to attach priority information to the user need tables. It may be difficult to specify any priority order for the needs, but usually the most essential needs are often identified.

User need tables are not able to present all user needs; other representations such as user profiles and photographs can be used in parallel. However, the purpose of user need tables is to summarize several kinds of user information and to facilitate the use of this information when user requirements are defined.

7.4 WRITING USER REQUIREMENTS FROM THE USER POINT OF VIEW

User need tables form the basis of writing user requirements and in particular they help developers to write use cases from the user point of view. A use case driven approach is one way of defining user requirements. Originally Jacobson (Jacobson, 1992; Jacobson, 1995) introduced use cases as a part of object-oriented methodology. Rumbaugh (Rumbaugh, 1994) describes use cases as the possible sequences of interactions between the system and one or more actors. Thus, use cases provide a more holistic and dynamic view of user requirements than the traditional single-requirement statements alone.

Jacobson (Jacobson, 1992) employs a graphical use case model which shows the system as being bounded by a box, with each actor being represented by a person outside the box, and use cases represented as ellipses inside the box. Rumbaugh (Rumbaugh, 1994) complemented the model by proposing a written description of the use case including name, summary actors, preconditions, description, exceptions, and post conditions.

Use cases can be written in a wide variety of forms and at different levels, but we have found that the original black-box view is the most useful one from the user point of view. Thus, only the external functioning or services to the user are described. The idea is to give high level descriptions of the basic functions and not to describe user interface details. In Table 7.3, the example use case description includes some details because the system in question was a new version of an existing system and it was known that some of the details were not going to be changed.

We have found that these kinds of high-level use case descriptions have value in facilitating communication among the project team. Use case descriptions help the project team to gain a coherent view of the system. Definition work did not proceed too quickly along technical lines.

We have used Rumbaugh's (Rumbaugh, 1994) description of use cases, except that we organized the written description of the use case in steps with numbers and connected the exceptions to steps identified by numbers. We also describe the goal of the user in preconditions-part: what users are trying to accomplish and why (Constantine, 1995).

Table 7.3 An example of a use case description

Use case:	**Making An Emergency Alarm Call**
Summary:	An entrapped passenger pushes the emergency alarm button in order to get help. A service center 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 center 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 center operator gets a visible notification of the emergency alarm call on the screen with an optional audio signal. Step 3: The service center operator accepts the emergency alarm call. Step 4: The system opens a voice connection between the service center operator and the passenger. Step 5: The system indicates to both the passenger and the service center operator that the voice connection is open. Step 6: The system guides the service center operator as to what information to ask of the passenger. Step 7: The service center operator
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 center 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 center operator will contact a serviceman who will help the passenger out of the elevator safely as soon as possible.

Inventing use case steps is difficult if developers do not know the users' tasks and needs. For example, in one of our industrial cases, the use case descriptions lacked the necessary level of detail and also the user-point of view when user need information was not available (Kujala et al., 2001a). In addition, we found that gathering user feedback with use cases is not enough. Users still interpret use cases on the basis of their present way of performing the tasks. If something is missing from the use case, they assume that it will nevertheless be implemented in the product. These implicit assumptions undermine the mutual understanding between users and developers.

User need tables inform developers as to how the task should be carried out and what the basic problems to be solved are. In addition, the tasks and objects are described in the users' language. In table form the information is in an organized form and the developer can consider user problems step by step and avoid the perception of having to deal with an overwhelming amount of data.

User need tables and use case descriptions complement each other, thus it is easy to move from one to another. The difference is that a user need table describes a specific present state user situation and context of use and there can be several versions of it, whereas the use case describes the general solution to how the task is performed with the new system. The idea is not to copy the task sequence as such, but to redesign all the necessary parts in order to solve user problems or realize the opportunities. Otherwise, the task sequence familiar to users is retained merely for the sake of convenience.

7.5 EVALUATING THE APPROACH IN INDUSTRY

The approach of linking user needs to use case-driven requirements engineering was developed and evaluated in several published, and a few unpublished, industrial cases in realistic product development settings. A summary of the published studies and related research problems and data gathering methods are described in Table 7.4. The case-study research strategy and multiple sources of evidence were used, as recommended by Yin (Yin, 1994). The costs and benefits of the approach were evaluated by using documentation, participant-observations, interviews, and questionnaires.

In most of the cases the approach was piloted in companies by real developers and the role of the researchers was that of an expert or a consultant who provided information, instructions, training, and support for the practitioners. Thus, it could be seen that the approach was practical enough to be used by real developers in real product development context.

The products under development were a PDA-device, a portable communications device, elevators and escalators, an information system for building designers, and weather measurement instruments. The size of the involved companies varied from small to large.

As a result of the studies, a practical field study approach was synthesized and evaluated. The general results of the studies are summarized in Table 7.5. It was found that the approach was useful even in a short time frame with relatively low costs. The total cost of the field studies varied from 46 to 277 person hours. Developers, a usability expert, and salesmen evaluated the results of the field studies as being very

Table 7.4 The research problems and data gathering methods

Study	Name	Problem	Data gathering method
I	User involvement: A review of the benefits and challenges (Kujala, 2003)	What are the benefits and challenges of user involvement in product development?	Literature review
II	Studying users for developing usable and useful products (Kujala and Mäntylä, 2000b)	How can field studies be applied in product development?	Participant-observation, interview
III	How effective are user studies? (Kujala and Mäntylä, 2000a)	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 (Kujala et al., 2001a)	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 (Kujala et al., 2001b; Kujala et al., 2003)	How can the proposed approach be introduced to product development cases?	Participant-observation, questionnaire, interview

Table 7.5 The results of the studies

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 field studies be applied in product development?	A field study approach was developed. The approach was tested in one industrial 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 field study approach was evaluated to provide useful information for product development. Preliminary evidence suggested that field 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 developers 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 cases?	In introducing field studies to product development small-scale pilot studies motivated the developers. Developers and salesmen found user studies useful. Innovation and customer satisfaction were increased.

useful in interviews and questionnaires that were conducted. In addition, the field studies provided new product ideas and improvements to existing products.

Customer satisfaction seemed to increase, although it was not directly measured. In Study V, the customer evaluated the product development company as being superior compared to others after the field study. The product development company achieved direct financial benefits as the customer signed a service contract with the company.

Furthermore, user need tables were found to help developers to utilize the field study results and to write more complete and correct use cases. The developers could more easily understand user needs and write use cases from the user point of view. If user need tables were not available, the use case descriptions missed the necessary level of detail as the developers were not aware of all the steps necessary to achieve the users' goal.

Use cases helped designers to gain a coherent view of the product. Undefined missing details were identified and definition work did not proceed too quickly along

Table 7.6 The results of the comparative usability test in Study III

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

technical lines. The developers said that they could use the use cases as checklists to guide the definition work and write instructions. They also noticed that use cases could be used as test cases.

In addition to the case studies, empirical data was gathered from an experiment in Study III (Kujala and Mäntylä, 2000a). The field study approach was evaluated by redesigning the functionality of an existing product based on a field study and comparing the process and results with the baseline design process, in which the functionality was first developed. The field study included six users; the total time spent on the study was 46 person hours. The baseline design process was very iterative and rapid prototyping was used. The only direct link to users was through usability tests. An estimation of the time spent on design was not available, but a rough estimation of the resources allocated to the usability test can be derived from the fact that 33 users participated in them.

The Study III provided preliminary evidence that field studies represent a more effective way of improving the usability of the product than iterative usability testing. For example, the results of both field study process and baseline design process were evaluated in a comparative usability test. As shown in Table 7.6 four experienced users of the baseline product spent slightly more time performing the tasks than the four novices using the changed product.

Usability evaluation techniques "react" to an existing design and thus are aimed at "improving" rather than "creating" (Wixon et al., 1994). It seems to be easier to do this improving with usability testing if the user needs are properly understood at the beginning and the system is aimed in the right direction. In addition, the success of a usability test depends on how representative the test task and environment are and this information is gained from field studies. However, even if the user needs are understood their meaning needs to be interpreted and translated to a system. Thus, field studies provide essential information about user needs and use scenarios; usability testing has its own role in validating interpretations and evaluating the usability of the practical solution.

In summary, we have presented how user-centered requirements analysis can be integrated to use case-driven requirements engineeringw. The proposed approach describes step-by-step how user needs can be discovered and utilized in requirements engineering. Our case studies indicate that the approach is simple and practical enough to be used in real development contexts.