

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of Helsinki University of Technology's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Introducing Requirements Engineering: How to Make a Cultural Change Happen in Practice

Marjo Kauppinen, Sari Kujala, Tapani Aaltio and Laura Lehtola

Helsinki University of Technology,

Software Business and Engineering Institute

P.O.Box 9600, FIN-02015 HUT, Finland

{marjo.kauppinen, sari.kujala, tapani.aaltio, laura.lehtola}@hut.fi

Abstract

Introducing requirements engineering appears to involve a cultural change in organizations. Such a cultural change requires that requirements are defined and managed systematically, not only from a technical point of view, but also from the customers' and users' points of view. This paper describes experiences gained from four Finnish organizations that have started to introduce requirements engineering to their product development. The goal of this study was to evaluate which factors support, and which prevent, a cultural change. Linking business goals to technical requirements via user needs and user requirements was one of the key improvement actions that supported cultural change. Eliciting needs directly from real users and representing user requirements in the form of use cases were also key activities. However, bringing about a change of culture was challenging because both managers and product development engineers held beliefs that prevented active user need elicitation and systematic user requirement documentation.

1. Introduction

Requirements engineering is a relatively new term [24]. In system engineering, requirements engineering is the science and discipline concerned with analyzing and documenting requirements [25]. It comprises needs analysis, requirements analysis, and requirements specifications [25]. In other words, requirements engineering (RE) means that requirements for a system are defined, managed and tested systematically.

Even though requirements engineering has a fairly narrow goal – to determine a need and define the external behavior of a solution [5] it seems to be a challenge for organizations. Introducing requirements engineering is a change of behavior and culture and not just a change of process and technology [10].

Two case studies have reported the cultural change towards systematic customer requirements management

as a success factor for RE process improvement. The Digital Equipment Corporation wanted to attain cultural change i.e. it sought to move from a technology-centric view of product development to a customer-centric view because the issue of “right” requirements was found to be a root cause of requirements churn and schedule overrun [7]. In a department at Ericsson Eurolab, an analysis had shown that many of the problems in software development had their root cause in an insufficient understanding of the customer and in unclear requirements [10]. The whole way in which the department was defining and dealing with requirements was made doubtful [10]. Consequently, a new tool, a changed process, or another policy would have not solved the problem [10]. A massive effort was required to effect a change of culture and behavior [10].

According to our earlier research results, the change of culture first requires that product development personnel fully understand the reasons for documenting requirements from the customers' point of view [11]. Secondly, they must commit themselves to defining and managing customer requirements systematically [11]. Cultural change requires people to change both their way of thinking and behaving.

The goal of this study is to evaluate the factors that support, and also those that prevent, the success of a cultural change. This paper describes lessons learned from four Finnish organizations that have started to introduce requirements engineering to their product development. The lessons are based on data collected through observations, informal conversations, formal interviews, official meetings, and document studies.

The paper is structured as follows. The four cases are described in Section 2 and the lessons learned from these four cases are explained in Section 3. Finally, we summarize the conclusions.

2. The Cases

The experience drawn on in this research comes from work with four industrial partners of the Qure (Quality through Requirements) project. The high-level research

goal of the Qure project is to investigate how organizations can develop products that better satisfy user and customer needs.

2.1. Background

All the four organizations were product development units of medium-size or large companies (Table 1) and the number of the employees of these units varied from 25 to 160. The companies focus mainly on market-driven products, but, occasionally, they develop customer-specific systems. The companies represent four different kinds of application domain (Table 1).

Company	Number of employees	Application domain
A	23 000	Transportation systems for buildings (elevators and escalators)
B	1100	Measurement systems for meteorology, environmental sciences and traffic safety
C	500	Information management systems for building, public infra and energy distribution designers
D	3200	Patient monitoring systems for anesthesia and critical care

Table 1. Description of participating companies

All the case organizations have decided to introduce requirements engineering in research and product development (R&D) units and have therefore started an RE process improvement project. Table 2 presents the scope of the project in each organization.

R&D organization	Number of R&D units	Improvement started
A	One	February 1999
B	Two	February 1999
C	Three	June 2000
D	One	August 2001

Table 2. Scope of the RE process improvement project

2.2. Approach to RE Process Improvement

The Requirements Engineering Good Practice Guide (REGPG) [24] was selected as a basis for the RE process improvement of the four case organizations because it is unique in focusing on requirements engineering. The REGPG covers eight RE areas and sixty-six good

requirements practices [23, 24]. The practices have been abstracted from existing standards, reports of requirements practices, and the experience of practitioners [23]. The main components of the REGPG are the REAIMS maturity model and a set of improvement guidelines that are based on the sixty-six good requirements practices.

The REGPG offers only very general suggestions for facilitating process change [24]. To guide the systematic RE process improvement of the case organizations, we defined a simple process improvement procedure (Figure 1). The procedure combines tasks from the IDEAL model [18] and the ISO/IEC 15504 standard [9]. The process improvement procedure does not show all the iterations. In practice, there were no distinct boundaries between the activities. Development and piloting especially were interleaved and there was a great deal of iteration between these two activities.

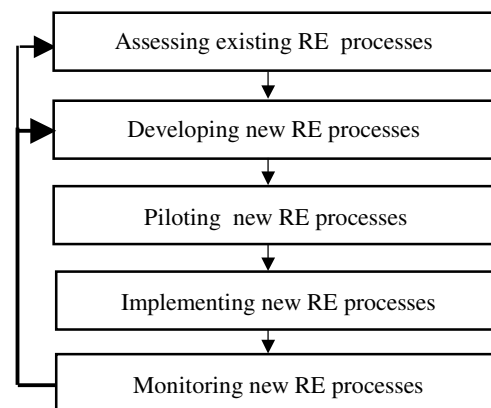


Figure 1. Process improvement procedure of the study

All the case organizations have invested substantial amount of resources in RE process improvement. The improvement groups have included product development engineers, product development managers, product managers, sales persons, usability experts, and process experts. The process improvement projects of each case organization are still in progress. Table 3 shows the status of the RE process improvement in January 2002.

Improvement activity	R&D Organization			
	A	B	C	D
Assessment	x	x	x	x
Development	x	x	x	x
Piloting	x	x	x	x
Implementation	x	x		
Monitoring				

Table 3. Status of the RE process improvement

2.3. Initial State of the RE Process

The maturity level of the RE processes of the organizations was assessed using the REAIMS maturity model. We found projects that had produced a good requirements document, but the RE processes of all the case organizations were at the lowest level of the REAIMS model. Figure 2 illustrates the initial state of the RE process of each case organization.



Figure 2. State of the RE process of each case organization

Requirements were defined mainly from a technical point of view. Most of the requirements described design solutions; only some of them were defined from the customers' and users' points of view. Requirements practices were also dependent on individuals. Domain experts had a good understanding of customers and users. The tacit knowledge of these domain experts was informally communicated in the product development projects. In summary, the case organizations did not have a culture of defining and managing requirements systematically from the customers' and users' points of view.

All the case organizations have developed successful products for years. They have experts that understand the application domain well, and these domain experts have effectively shared their knowledge with others. In other words, the companies and their product development have been successful. However, because the product development environment is changing, they started to improve their RE processes, rather than continue to rely on the tacit knowledge of the domain experts.

- Products are bigger and more complex than before and therefore domain experts are unable to handle all the requirements in their heads.
- Projects are bigger and there are more people that need the tacit knowledge of the domain experts.
- Personnel changes are faster than before and it is possible that domain experts are not available throughout the product development project.
- Product development projects are faster and there is less time to correct requirements mistakes in the later phases of the projects.

3. Findings

The authors participated actively in the RE process improvement projects of the case organizations. In addition to the involvement in the projects, observations, informal conversations, formal interviews, official meetings and document studies were used as methods for collection of the data, on which the findings of the study are based.

The goal of this study was to evaluate the factors that support, and also those that prevent, the success of a cultural change. We identified five improvement actions that especially supported the cultural change in practice:

- linking business goals to technical requirements via user needs and user requirements
- integrating a simple RE process with the product development process
- eliciting user needs actively
- representing user requirements systematically in the form of use cases
- raising people's awareness of RE

We also identified several beliefs that prevented the cultural change taking place in practice. Both the key improvement actions and the obstacles of cultural change are described in more detailed in the following two sections.

3.1. Actions that supported cultural change

Linking business goals to technical requirements via user needs and user requirements: Our main assessment finding was that requirements were mainly documented from a technical point of view in the case organizations. Customer and user related information was tacit knowledge in experienced people's heads; links from business information to technical requirements were missing.

Figure 3 summarizes the good practices that enabled the customer and user related tacit information to be made explicit in the case organizations. First of all, it was useful to make a distinction between customers and users because they are not the same in the business contexts of the case organizations. The IEEE Standard 830-1998 offered suitable definitions. Customers are persons who pay for the product [8]. Users are persons who operate or interact directly with the product [8].

Documenting high-level goals from the points of view of both customers and the product development project was the first step towards making tacit business information explicit (Figure 3). Customers have objectives that define the reasons for buying a new system. Customers may, for example, want to use the new system to reduce costs or to improve the quality of business services. The product development project also has its business goals that define reasons for producing

the new system from the company's perspective. The company may for example want to increase market share by developing a new version of the system for a new user group.

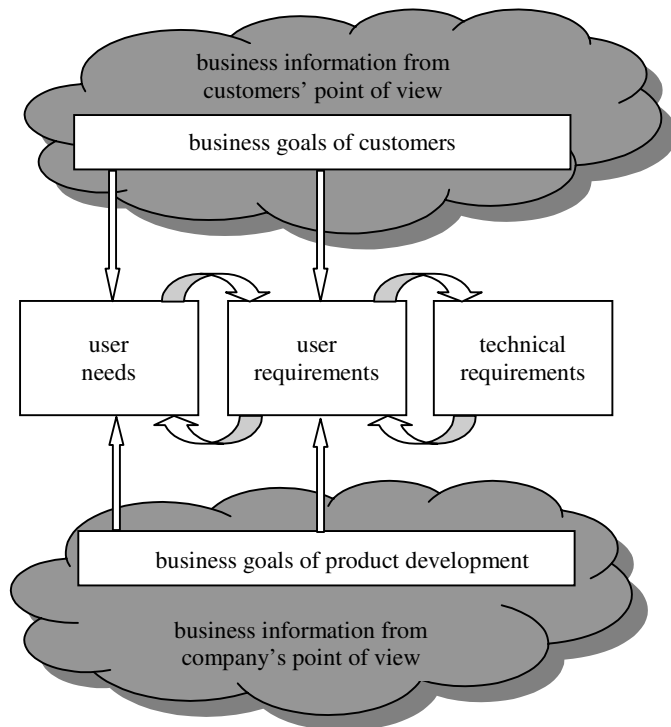


Figure 3. Linking business goals to technical requirements via user needs and user requirements

The customer's primary goal is usually to buy a system, which supports users in their tasks [4]. However, customers do not typically know what users really need and what is essential in their tasks. To connect customers' high-level business goals to technical requirements, two types of user information were found to be useful (Figure 3). Firstly, user needs describe problems and opportunities related to the current situation and context of use. Context of use refers to user characteristics, users' goals, tasks, equipment, and the environment in which a new system will be used. Secondly, the user needs are analyzed and refined to user requirements that define the external behavior of the new system from the users' point of view.

The first step in linking business goals to requirements was to use them as drivers when user needs were elicited and transformed to user requirements (Figure 3). Product development personnel also found it important to create traceability links between user needs and user requirement, and also to technical requirements (Figure 3). A prerequisite for forward and backward traceability was to give a unique identifier for each user need, user requirement and technical requirement.

Integrating a simple RE process with the product development process: The existing RE practices of the case organizations were dependent on individuals; the quality of the requirements documents varied from project to project. All the case organizations had a well-defined product development process but none of them had a documented RE process.

The case organizations defined a simple process to give an overview of RE and to facilitate their personnel's understanding of the basics of RE. Three organizations used the coarse-grain activity model developed by Kotonya et al. [13] as a basis for their first RE model. One of the organizations decided to tailor the requirements sub-process of the Rational Unified Process to its purposes.

Figure 4 summarizes the good experiences of RE process modeling in the case organizations. The main idea was to keep the RE process model simple. Because the personnel of the case organizations were not aware of requirements engineering, the simple model helped practitioners to understand the basics of the RE process. It was also useful to determine the outputs of the activities, as these made the RE process visible and concrete.

All four organizations found it important that RE should not be left as a separate activity. It was important that people could see how the RE process relates to the organization's product development process. The pilot projects showed that a good user requirements document could be used not just by designers, but also by system testers. In addition, user requirements documents can offer valuable information to user manual writers, as well as marketing and sales personnel.

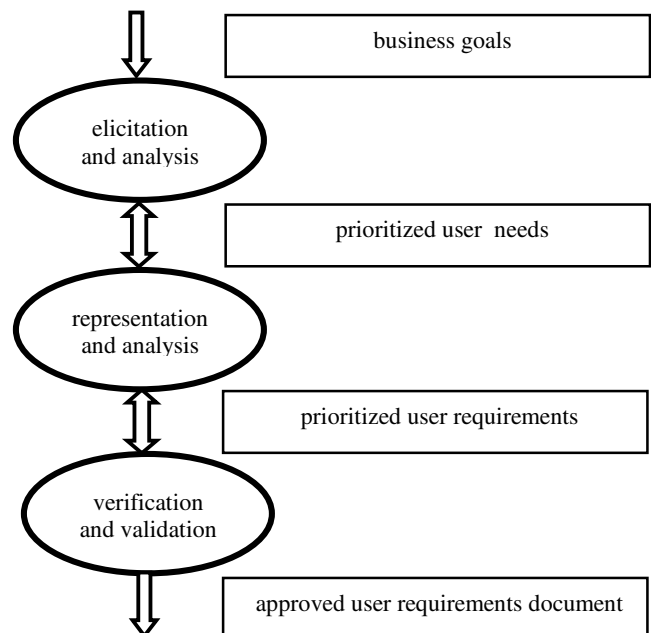


Figure 4. A simple RE process

The pilot projects showed that there is a great deal of iteration and feedback from one activity to another. One of the biggest challenges in the future will be to establish how the case organizations can make the RE process iterative and incremental, while keeping it simple and systematic at the same time.

Eliciting user needs actively: The assessment showed that five out of nine projects had no direct contact with users or that the contact did not take place until the late prototype phase. This means that in many projects user needs were not actively discovered, but requirements were invented and customer information was gained through sales persons and other intermediaries. Developers felt that more information was needed about users, but they did not know how to gather and document the information.

We found that new and useful information about user needs could be elicited even in a short time frame with relatively low costs. Users are not good at articulating their needs. Therefore, in addition to interviewing them, users were observed in their own environment. The focus was not on what users said they wanted or needed, but on understanding their goals and current tasks, while recognizing users as experts in their tasks. As users were observed in their real situation using their tools, many needs and new product ideas were discovered. For example, a warehouseman had difficulties in pushing elevator buttons with his gloves on, while the developers in their offices had no idea that this problem existed.

Moreover, cross-functional teams were a useful approach in eliciting user needs. The teams could utilize the knowledge and views of members with varied backgrounds from sales, marketing, and product development. The discovered user needs were also easier to convey to projects when a project manager participated in team-work.

However, it was not enough to discover user needs. We found that the results must be documented in such a form that product development engineers can use easily when defining user requirements. In addition to verbal reports, video shots and photographs, we developed a user need table to describe user task sequence and link user needs to tasks [15]. The user need tables are near to use case descriptions and so they helped developers in utilizing user need information in defining user requirements by use cases. The task sequences were redesigned in use cases and the identified user needs and problems were analyzed step by step.

Representing user requirements systematically in the form of use cases: The analysis of the existing RE practices showed that most of the requirements documents were based on narrative text, which made it difficult to recognize user requirements from the text. In some projects, all requirements had already been assigned a unique identifier, but they had been put into a long list and

dependencies between requirements had not been documented.

To make requirements easier to write and read, the case organizations defined a standard template for user/customer requirements documents. The document structure of the IEEE Standard 830-1998 [8] offered valuable information but it was too complex to be used as such. The case organizations wanted to have a simple structure that people could learn easily. It was also useful to determine the purpose of a user requirements document because this helped product development personnel to understand why they should write such a document. Real document examples from an organization's own application domain supported the use of the standard template.

Representing requirements from the users' point of view was not easy because software engineers had to change their way of thinking. In the past, they had defined the internal behavior of the system. Now, they were supposed to describe external functions of the system in the user's language. To support the change in the way of thinking, all the case organizations decided to pilot use cases in representing user requirements.

Even though the use case approach is simple, it did not automatically support engineers in describing the external behavior of the system. In some projects, software designers went on to describe the internal behavior of the system, and sometimes they defined details of the user interface using use cases. To overcome this granularity problem, the basic principles of use cases had to be emphasized. The basic principles included in the training material of the case organizations were as follows:

- Use cases are a way of specifying functionality from a user's point of view [22].
- The system is considered as a black box: We are interested in externally visible behavior [22].

Documented user needs helped software engineers write use cases at the appropriate level and from the users' point of view. Descriptions of users' goals and present tasks provided information that was especially useful for use cases.

Real examples from an organization's own application domain also supported software engineers in writing use cases at the appropriate level and from the users' point of view. In addition, standard templates for representing individual steps of use cases were essential. The use case approach also helped software engineers to build hierarchies and dependencies between user requirements.

Raising people's awareness of RE: There were usually one or two people in each of the case organizations that were convinced of the importance of defining requirements systematically from the users' and customers' points of view. The experiences of the pilot projects indicated that organizations need a strategy to discover how to increase awareness of RE, and how to

create a cultural change that affects the whole organization permanently.

The pilot projects showed that the cultural change starts from an awareness of requirements engineering. Firstly, product development personnel needed to understand what RE means and how they can benefit from it. The simple process model helped the case organizations to create a common terminology, while also offering practical guidelines as to how to produce a good user requirements document. The first implementation experiences of the two case organizations showed that introducing the basics of RE requires a day's training course. However, the traditional classroom style of teaching was ineffective when the objective was to integrate RE with daily routines. The pilot projects showed that "Just-in-Time" training combined with learning-by-doing was required to change the way in which product development teams defined requirements in practice. Hutchings et al. have introduced the idea of "Just-in-Time" training [6]. The principles of this training method are 1) the teaching occurs when people are ready, 2) the teaching material is focused on their actual situation, and 3) the teaching is accompanied by expert facilitation and consulting [6].

In addition, raising management awareness of requirements engineering was found vital. Management support for RE was particularly important because RE practices concern not only product development, but also such organizational units as a company's sales, sales support, and marketing.

3.2. Obstacles that prevented cultural change

During the study, we identified four obstacles that prevented cultural change from taking place in practice. Product development personnel believed that they knew what users needed and they did not understand why they should discover needs directly from users. People also thought that discovering needs directly from users was both difficult and risky. The fourth obstacle was that people did not understand why they should document user requirements systematically. Table 4 lists the obstacles and beliefs we came across in the case organizations. The beliefs are examples of factors that prevented product development personnel from changing their way of thinking and behaving.

Developers having long experience in the application domain believed that they knew what users needed. The user studies showed that developers were often surprised by how the users behaved and what they really expected from the system. Another common belief was that developers could themselves act as users. The experiences indicated that developers are not good representatives of users because their technical knowledge affects their way

of using the new system and they may make wrong assumptions.

Obstacles	Beliefs
It is not worth discovering needs directly from users.	We have developed these kinds of products for a long time and we know user needs.
	We developers also use our own products and therefore we can act as users.
	We are developing a new product and therefore users cannot have any needs for it.
It is difficult to discover needs directly from users.	Users are unable to say what they need and want.
	There are so many users that we cannot interview them all.
It is risky to discover needs directly from users.	We cannot show our customers that we do not know the basics of their business.
	Product development personnel can spoil customer relationships by asking stupid questions.
It is not worth documenting user requirements systematically.	Our customers are interested in technical requirements and in details of the user interface.
	We do not have time for user requirements documentation.

Table 4. Beliefs related the obstacles of cultural change

In addition, practitioners found it odd to focus on users' present processes when a new product was planned to provide a new way of carrying out the existing tasks. However, the new product will not be used in vacuum; user needs depend on the context of use. In order to develop a successful product, a developer needs to understand what kinds of persons will use the product, what they will want to achieve and in what context they will use it. We found that it is useful to have a direct contact with a potential user and to understand the present tasks and circumstances. Even though it is a new product that is being planned, users already perform the tasks in a particular way and order, and this affects their expectations of the new product.

Many developers had experiences of users who were unable to say what they needed and wanted. The pilot projects showed that combining elicitation techniques such as observing and interviewing provided a more complete picture of needs without requiring users to articulate them explicitly. Practitioners also found it difficult to discover needs from users when a system is developed for the mass market and can have thousands of users. The pilot experiences showed that it was possible to

identify the various kinds of potential user groups and select representatives from all essential groups. It was better to elicit needs from two or three typical users than neglect eliciting real user needs totally.

Managers and sales personnel were concerned that product development personnel can harm customer relationships when discovering needs directly from users. For example, in Company A, salespersons were slightly suspicious toward allowing development people visit their customers, but later on they found the results useful and sent thanks via e-mail as the customer had evaluated Company A as superior following the visit, and had made a big service contract [14].

Product development personnel explained that customers are interested in the technical features of a new system and therefore, have documented only technical requirements and not the external behavior of the system. Experience showed that users may believe that a technical feature or a user interface detail solves their problem while, in fact, once the underlying need has been discovered and analyzed, designers may well be able to find a better way to solve users' problem.

Both managers and product development engineers assumed that systematic user requirement documentation is too time-consuming. A pilot project showed that it was useful to document user requirements systematically even in the design phase because critical requirement omissions were discovered. In the project manager's opinion, the use cases would have been even more useful at the beginning of the project. The test manager of the project was also pleased because the use cases could be used as test cases, so the test group was able to save a lot of time getting them from software engineers.

To overcome the obstacles presented in Table 4, it was essential to respect the skills and knowledge of product development personnel. Instead of pointing out the weaknesses of the current practice, product development personnel needed information about what they can gain from defining user requirements systematically. Table 5 lists the experiences clearing the mental obstacles. These experiences motivated practitioners when they speculated about the value of active user need elicitation and systematic user requirements documentation. To become convinced of benefits, practitioners also needed to see results from the new RE practices at first hand. Small-scale pilots of the new RE practices offered product development personnel concrete and visible evidence of the benefits.

The practitioners, especially managers, emphasized that it was also crucial that new models and methods were easy to learn and use. One of the managers said that a method is simple enough if product development engineers can learn it in a day. The pilot projects showed that people could concentrate on domain knowledge when the models and methods were easy to learn and use.

Obstacle	Answers
It is not worth discovering needs directly from users i.e. Why discover needs directly from real users	1) The more deeply user needs are understood, the more useful and usable systems can be developed. 2) A useful and usable system supports users' goals and tasks. 3) Users are experts in their tasks, and therefore they are the primary source of real user needs.
It is difficult and risky to discover needs directly from users i.e. How to discover needs directly from users.	1) There are elicitation techniques like interviewing and observing that product development engineers can learn and use easily. 2) By combining interview and observation techniques it is possible to get a comprehensive picture of user needs without requiring users to articulate their needs explicitly. 3) Well-prepared user visits can improve a company's image among its customers, and can create competitive edge.
It is not worth documenting user requirements systematically i.e. Why document user requirements systematically	1) Systematic user requirements documentation ensures that a right product will be developed. 2) Well-documented user requirements are useful information for many groups: designers, testers, user- manual writers, managers and marketing personnel. 3) Systematic user requirement documentation at the beginning of the product development saves time and decreases rework in later phases of a project.

Table 5. Answers to the obstacles

4. Conclusions

In this paper, we have described the practical experience of four Finnish organizations that have started to introduce requirements engineering to their product development. In all the case organizations, the introduction of requirements engineering involved a change of culture. The goal of the study was to evaluate the factors that affect the success of the cultural change.

Cultural change requires product development engineers to change their way of thinking as well as of working. Instead of describing requirements from a technical point of view and relying on the tacit domain knowledge of experts, it is beneficial to define requirements systematically from the customers' and users' points of view also. Two other case studies also

indicate that moving from a technology-centric view of product development to a customer-centric view have significant implications for product development performance [7, 10]. Furthermore, Basili et al. report that most effective process changes are those that leverage the thinking of developers [2].

Organizations can support a cultural change by applying a user-oriented approach for requirements engineering. The basic idea of the approach is to link business goals to technical requirements via user needs and user requirements. User needs describe the users' current situation as they focus on problems and opportunities. User requirements are defined on the basis of needs, and they describe the external behavior of a system from the users' point of view and in the users' language.

The user-oriented approach described in the paper has similarities with three other approaches published in the RE literature [16, 21, 27]. All the approaches have multiple categories of requirements that represent different perspectives and varying degrees of detail. According to our study, multiple requirement categories help practitioners make tacit domain knowledge explicit and structure hundreds, or even thousands, of requirements.

The cultural change starts from an awareness of requirements engineering. The division of the requirements into different categories helps practitioners to get a structured view of requirements engineering. In addition to this, a simple RE process model facilitates an understanding of the basics of RE by personnel and gives them an overview of RE. According to Armour, projects that are lacking awareness can only use metaprocesses [1]. They cannot use a detailed process, because they don't know what process might work [1].

We found two RE activities that especially support a cultural change in practice: 1) eliciting user needs actively, and 2) representing user requirements systematically in the form of use cases. The main principle of the active user need elicitation is that needs are gathered directly from real users in their own environment. Also, Lubars et al. report that most informants in their field study felt that they understood the requirements best when they interacted directly with users [17]. Furthermore, Keil et al. found that the more successful projects employed more links to customers and users than did the less successful [12].

Use cases are a good way of representing user requirements because they support practitioners in specifying externally visible functionality of a system from the users' point of view. User requirements, represented in the form of use cases, also serve as a means of communicating the functions and properties of a new system with designers, testers, user-manual writers, managers and marketing personnel. The experiences of

the case organizations support the findings of Weidenhaupt et al. who report that practitioners apply use cases for several different kinds of purposes [26].

Our four cases showed that cultural change is challenging and requires more than the introduction of a new RE process and new RE technology. The most demanding obstacles to cultural change are the beliefs that both managers and product development engineers hold. To overcome mental obstacles, it is essential to offer practitioners concrete evidence of the benefits of RE practices. Two other case studies [7, 10] also show that the cultural change towards systematic customer requirements management is demanding. Furthermore, Morris et al. report that non-technical issues were identified as fundamental problems encountered on RE R&D projects with industrial uptake [19].

The main contribution of this paper is that it describes the practical problems and solutions found applicable in four R&D organizations. The research started as an investigation of RE process change. The long-term view of the study led us to examine also the cultural change required by the successful introduction of requirements engineering. This paper supplements the case studies of the market-driven RE process published earlier [3, 7, 10, 20, 28]. It provides further insights into the improvement of market-driven RE processes. The paper might serve as an inspiration for other companies that are interested in introducing RE to their product development and making a cultural change take place in practice.

In the future, we will focus on three research questions. Firstly, we will investigate how to connect user needs and user requirements more closely to business information and business processes. We will also continue to analyze how practitioners can create traceability between user requirements and technical requirements. In addition, we will continue to follow up how the case organizations can introduce requirements engineering organization wide and how to make a cultural change take place at an organizational level.

Acknowledgements

The authors would like to thank the participating organizations for their cooperation and willingness to share their experiences and data. We would also like to thank professor Reijo Sulonen at Helsinki University of Technology for his invaluable feedback and comments. This work is mainly funded by the Technology Development Center (Tekes), Finland.

References

- [1] P. Armour, "The Laws of Software Process - Getting from here to there: Putting a boundary around process.", *Communications of the ACM*, Vol. 44, No. 1, January 2001, pp. 15-17.

- [2] V. Basili, M. Zelkowitz, F. McGarry, J. Page, S. Waligora, and R. Pajerski, "SEL's Software Process Improvement Program", IEEE Software, Vol. 12, No. 6, November 1995, pp. 83-87.
- [3] P. Carlshamre and B. Regnell, "Requirements Lifecycle Management and Release Planning in Market-Driven Requirements Engineering Processes", Proceedings of 11th International Workshop on Database and Expert Systems Applications, 2000, pp. 961-965.
- [4] J. M. Coble, J. Karat, and M. G. Kahn, "Maintaining a Focus on User Requirements throughout the Development of Clinical Workstation Software", Proceedings of the Conference on Human Factors in Computing Systems (CHI'97), 1997, pp. 170-177.
- [5] A. M. Davies and P. Hsia, "Giving Voice To Requirements Engineering", IEEE Software, Vol. 11, No. 2, March 1994, pp.12-15.
- [6] T. Hutchings, M. G. Hyde, D. Marca., and L. Cohen, "Process Improvement that Lasts: An Integrated Training and Consulting Method", Communications of the ACM, Vol. 36, No. 10, October 1993, pp. 105-113.
- [7] A. F. Hutchings and S. T. Knox, "Creating Products: Customers Demand", Communications of the ACM, Vol. 38, No. 5, May 1995, pp. 72-80.
- [8] IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998 (Revision of IEEE Std 830-1993), IEEE, New York, 1998.
- [9] Information technology – Software process assessment – Part 7: Guide for use in process improvement, Technical report, ISO/IEC TR 15504-7:1998(E), Switzerland, 1998.
- [10] S. Jacobs, "Introducing Measurable Quality Requirements: A Case Study", Proceedings of the 4th IEEE International Symposium on Requirements Engineering, IEEE Computer Society Press, Limerick Ireland, June 1999, pp. 172-179.
- [11] M. Kauppinen and S. Kujala, "Starting Improvement of Requirements Engineering Processes: An Experience Report", Proceedings of Third International Conference on Product Focused Software Process Improvement (Profes 2001), Springer-Verlag, Kaiserslautern Germany, September 2001, pp.196-209.
- [12] M. Keil and E. Carmel, "Customer-developer Links in Software Development". Communications of the ACM, Vol. 38, No. 5, 1995, pp. 33-44.
- [13] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, John Wiley & Sons Ltd, 1998.
- [14] S. Kujala, M. Kauppinen, and S. Rekola, "Introducing User Needs Gathering to Product Development: Increasing Innovation and Customer Satisfaction", Proceedings of Human-Computer Interaction Conference (INTERACT '01), IOS Press, Tokyo, Japan, July 2001, pp. 856-861.
- [15] S. Kujala., M. Kauppinen, and S. Rekola, "Bridging the Gap between User Needs and User Requirements", Proceedings of the Panhellenic Conference with International Participation in Human-Computer Interaction (PC-HCI 2001), Typorama Publications, Patras Greece, December 2001, pp. 45-50.
- [16] D. Leffingwell and D. Widrig, Managing Software Requirements – A Unified Approach, Addison-Wesley, 2000.
- [17] M. Lubars, C. Potts, and C. Richter, "A Review of the State of the Practice in Requirements Modeling", Proceedings of the IEEE International Symposium on Requirements Engineering (RE'93), IEEE Computer Society Press San Diego, CA, 1993, pp. 2-14.
- [18] B. McFeeley, IDEAL: A User's Guide for Software Process Improvement. Handbook CMU/SEI-96-HB-001. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PE USA, 1996.
- [19] P. Morris, M. Macera, and M. Wilikens, "Requirements Engineering and Industrial Uptake", Journal of Requirements Engineering, Vol. 3, No. 2, 1998, pp.79-83.
- [20] B. Regnell, B. Beremark, and O. Eklundh, "A Market-Driven Requirements Engineering Process – Results from an Industrial Process Improvement Programme", Journal of Requirements Engineering., Vol. 3, No. 2, 1998, pp. 121-129.
- [21] H. D. Rombach, Software Specifications: A Framework, SEI Curriculum Module SEI-CM-11-2.1, Carnegie Mellon University, Software Engineering Institute, January 1990.
- [22] J. Rumbaugh, "Getting Started – Using Use Cases to Capture Requirements", Journal of Object-Oriented Programming, Vol. 11, No. 1, September 1994, pp. 8-12.
- [23] P. Sawyer, I. Sommerville, and S. Viller, "Capturing the Benefits of Requirements Engineering", IEEE Software, Vol. 16, Issue 2, 1999, pp. 78-85.
- [24] I. Sommerville and P. Sawyer, Requirements Engineering - A Good Practice Guide, John Wiley & Sons, New York, USA, 1997.
- [25] R. Thayer and M. Thayer, "Software Requirements Engineering Glossary", Software Requirements Engineering, Second Edition, Edited by R. Thayer and M. Dorfman, IEEE Computer Society Press, 1997, pp. 489-528.
- [26] K. Weidenhaupt, K. Pohl, M. Jarke, and P. Haumer, "Scenarios in System Development: Current Practice", IEEE Software, Vol. 15, No. 2, March-April 1998, pp. 34-45.
- [27] K. Wiegers, Software Requirements, Microsoft Press, 1999.
- [28] A. Yeh, "Requirements Engineering Support Technique (REQUEST): A Market Driven Requirements Management Process", Proceedings of Second Symposium of Quality Software Development Tools, IEEE Computer Society Press, New Orleans USA, May 1992, pp. 211-223.