

Research Article

A Case Study of User-Centred Design in Four Swiss RUP Projects

Ljiljana Vukelja,¹ Klaus Opwis,² and Lothar Müller¹

¹ Department of Computer Science, University of Applied Sciences Rapperswil, 8640 Rapperswil, Switzerland

² Faculty of Psychology, University of Basel, 4056 Basel, Switzerland

Correspondence should be addressed to Ljiljana Vukelja, lvukelja@hsr.ch

Received 18 June 2010; Accepted 25 October 2010

Academic Editor: Ian Oakley

Copyright © 2010 Ljiljana Vukelja et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We analysed four Rational Unified Process (RUP) projects in Switzerland that identified themselves as following a user-centred approach. Grounded theory served for analysis of 12 interviews with software developers, project managers, and UI specialists. For each professional group we analysed their work context, motivations, work practices, and strategies used to overcome the obstacles to user-centred design. Results show that end users did not participate in the projects. Instead of working directly with end users, participants used data from marketing research or consulted colleagues from other departments. Prototypes played an important role. We suggest the following remedies: (1) developing methods for easy integration of existing company knowledge about products with usability features, (2) professionalising UI design by educating project stakeholders in standard UI design, (3) creating an approved pool of company's personas for UI specialists' work, and (4) educating customers on their right to get good user interfaces.

1. Introduction and Purpose of Research

Usability and ease of use are an increasingly important quality of software (SW) products. It has been recognised that humans have physical and cognitive limits and human error is not a valid excuse for usability problems with technological products [1]. As a method of choice early works in the area of user-centred design, see for example [2], recommended a high number of iterations and user involvement in the development process. Later, the method was enhanced, and today there is a standard body of literature that describes user-centred design (UCD) in detail [3–5]. We refer to user-centred design as defined in the ISO 13407 standard [6] with the following activities in particular:

- (1) understanding and specifying the context of use,
- (2) specifying the user and organisational requirements,
- (3) producing design solutions,
- (4) evaluating designs against requirements.

A review of the practice shows a poor spreading of user-centred design [7–12]. More research is therefore necessary

to understand why the existing methods are rarely used, and whether they can be adapted to achieve a higher acceptance in practical application or new methods should be developed.

2. Research Question

Our research questions were the following.

- (1) What is the professional profile of persons involved in user-centred design applied in Swiss Rational Unified Process (RUP) projects that claimed to work in a user-centred manner? RUP [13] is an iterative SW development process framework that can be tailored by SW organisations or teams to contain the elements of the process that are appropriate for their needs.
- (2) In which circumstances, why, and how is UCD work conducted in these projects?
- (3) In the context of user-centered activities, how do the practitioners overcome problems encountered?

The final aim is to identify some of the reasons for poor spreading of user-centred design and suggest remedies.

3. Related Work

A lot of research in human computer interaction has its roots in academia. For example, Gulliksen et al. [14] in a survey among usability professionals in Sweden found that 30% of the respondents working with usability consider themselves autodidacts and other 20% received on-the-job training. Thus, only 50% received formal training. They infer from this data that the impact of the cooperative and participatory design approaches has been limited to the research community. In a survey conducted by Vredenburg et al. [15], the familiarity level with UCD methods of practitioners who attended a CHI conference or are members of UPA was very high (median 6 on a 1–7 scale with 7 = very familiar). This might also indicate a strong connection to academia.

In light of this, Wixon [16] argued that usability methods should be discussed as they are used in practise, rather than from an academic viewpoint. He further states that the usability evaluation methods are discussed in academia mostly in terms of qualities that are not relevant in practise. For example, usability tests are judged depending on how many flaws can be found with how many participants. However, in the practise, it is more important whether the method encourages participation, buy in, and collaboration by the development team. He calls for more case studies in real engineering, corporate, and political settings on how real products have been developed. Metastudies of such case studies can reveal more general findings. An example of a case study in practise is given by Følstad et al. [17]. An important finding was that HCI practitioners tended not to evaluate their practice with regard to its impact on the development team and project leader.

There are different types of usability work: organisational usability, strategic usability, and interface usability. We refer to *organisational usability* as work towards fitting the systems to specific organisations, as defined by Uldall-Espersen and Frøkjær [18]. The results of Uldall-Espersen and Frøkjær [18] indicate that organisational usability work is lacking in both HCI and software engineering. *Strategic usability* was defined by Rosenbaum et al. [19] as embedding usability engineering in the organizational processes, culture, and product road maps. Rosenbaum et al. [19] have identified the following obstacles to strategic usability in corporate decision making: resource constraints, resistance to user-centered design/usability, lack of understanding/knowledge about what usability is, better ways to communicate impact of work and results, lack of trained usability/HCI engineers, lack of early involvement, no economic need (customers not asking for usability). Last but not least, *interface usability* was defined by Uldall-Espersen and Frøkjær [18] as design principles to improve the interfaces and account for end-user interests. Similarly, Rosenbaum et al. [19] talk about usability methodologies. Usability methodologies in work of Rosenbaum et al. [19] were usability testing inside or outside of lab facility, field studies, usage scenarios, task analysis, participatory design, contextual inquiry, heuristic evaluation, focus groups, and surveys. Even though there is a call for more strategic usability efforts, Rosenbaum et al. [19] have seen that both large and small HCI groups rated

usability methodologies as a whole lot more strategically effective than organizational approaches.

We follow up on these findings and study projects in companies in environments with no strategic support for usability efforts. Projects in our study were all exceptions in their companies as the companies do not normally work in a user-centred manner. The fact that in each project in our study there was a UI specialist is due to the fact that the UI specialists were usability enthusiasts who volunteered to be involved in the study. It was rather their personal interest than the company's interest in UCD that allowed us to investigate the four projects. During the research we did not intervene with the projects except for conducting the interviews. We looked into the practises of participants of four RUP projects carried out in a user-centred manner and analysed the work context, motivations and strategies they used to overcome the obstacles they encountered. Our study aims to reveal some of the reasons why user-centred design is not widely used in practise and to identify remedies for the future work.

4. Research Method

We used the case study method [20] for the collection and analysis of the data. We deemed the case study method suitable because we studied *how* something is done and *why*, without any predefined hypotheses. The *how* and *why* questions were open, and the enquiry was adjusted depending on the informant's answer.

We conducted 12 interviews with participants of RUP-based projects who identified themselves as working in a user-centred way. In a previous research on user-centred design in Switzerland [7] we asked the respondents whether they would like to report in more detail about their work. Four respondents were willing to do so and allowed us to look into their work and the work of their colleagues closely. Our contact persons were four isolated usability enthusiasts in companies that did not put any emphasis on usability matters. They nominated the other relevant roles who contributed most to the projects. As can be seen in Table 1, each project had a different team constellation. It is interesting to note that no end user was put forward as a main contributor to the UI of the product. A test manager in company B and a marketing representative in company C were acting as user representatives. Even though according to literature end users should belong to a team that works in a user-centred manner, we did not insist on interviewing end users, because they were not part of the project teams. Table 2 gives an overview of the topics that were discussed during the interviews.

Company A is a technology-consulting company. They work exclusively on projects for external clients. The project under consideration was a project where the customer explicitly asked for user-centred design. Companies B, C, and D are companies that sell highly technological products for expert use. Their projects were enhancements of company products.

To demonstrate the coding process, we describe the evolution of a piece of information from the interview

TABLE 1: Participants' roles within the projects.

Company A	Company B	Company C	Company D
UI specialist	UI specialist	UI specialist	UI specialist
SW developer	SW developer	SW developer	
Project manager	Project manager		Project manager
	Test manager as user representative	Marketing representative as user representative	

through to coding and the establishment of the families, and finally to the result. An example of raw data is

End users test it [the product] at the trade fairs.

(Project manager, company B.)

This statement was originally coded as “*Marketing*” and “*Testing*”. After memo writing and considering the codes in the bigger context, it turned out that as a norm testing with end users was conducted in the shadow of marketing activities because it was not possible to contact end users during development, the codes “*Appendage to Marketing*” and “*Unavailability of End Users*” were added. Both codes form part of the family called “*Obstacles to UCD*”. In the final results the above raw data example builds part of the work practice of UI specialists. We proceeded equally with all interview data.

We interviewed different people from the same project on the same topics and checked project documentation. The documentation was rather formal and did not reveal the whole spectrum of the activities undertaken, so we did not analyse it further. We only used the fact that it did not contradict the findings from the interviews.

Our case study is of exploratory nature and serves to understand the context, motivations, and obstacles experienced in UCD in environments with no strategic support for UCD. Each company formed one case.

We applied grounded theory [21] to analyse our data. Grounded theory is a systematic qualitative research methodology in the social sciences that emphasizes generation of theory from data gathered in the process of conducting research. Due to the small sample size and choice of projects studied, we cannot provide a theory. However, we used this systematic approach to handle the richness of the qualitative data we gathered. The main steps in the analysis were coding, memo writing, axial coding, sorting, and writing of findings. After each interview, we conducted open coding, meaning that we indexed the whole interview with codes as they came to mind, independently of previous findings. Memo writing served to condense the findings and identify the first results. Then, concepts and categories emerged from the codes. In the middle and at the end of the interviewing phase we conducted axial coding. In axial coding we looked through all interviews and codes and recoded all interviews such that the emerging categories were well covered. At the end of the process, we had 175 codes and 9 families. Some codes were too specific and did not belong to any family. The codes were then sorted graphically to establish which codes belong together and how they

influence each other. We do not show the codes for spatial reasons. We describe the results in the main part of the paper, see Section 6, on the ground of three main categories: work context (research question 2), motivation, and work practice (research question 1) of each role. Then we indicate the strategies that employees used to overcome the obstacles in their daily work (research question 3). In the end we analyse the impact of the different factors on user-centred design.

5. Generalisation of the Results and Limitations of the Study

Our results can be transferred to other settings where individual teams or team members are enthusiastic enough to develop software in a user-centred manner, even though in the organisation there is no strategic support for UCD. There are other settings where lone fighters cope with challenges of UCD when they get no support from the organisation.

When looking beyond the boundaries of user-centred design, the situation under investigation is a case where positive effects of particular efforts are not directly and immediately visible. They become evident only in the long term or to other people. In case of UCD, the end users profit from additional efforts, rather than project team members. Examples of similar situations could be investments in the maintainability of the software or architectural landscapes in big organisations. In both cases it is not clear to the project members as executing parties why certain tasks are necessary, because the benefits come to light only later or to other actors. As a typical remedy, in big organisations there are policies and dedicated teams that ensure that the applications are maintainable and that the application landscape evolves in a controllable manner.

There are several limitations to our study. The small number of participants and the procedure of their selection limits the ability to understand the diversity and heterogeneity of the findings across individuals and projects. There might be better samples that would have given a much clearer or different picture.

As already mentioned, the main contact persons in our study were usability enthusiasts. It is possible that the mere possibility to show their work to outside parties influenced them and their colleagues to present their practises in a distorted way. Two variants are possible: (1) participants described their work negatively in the hope that their work conditions will get better and (2) participants described their work positively to impress the researchers.

TABLE 2: Interview topics.

Topic	Details
Introduction	Project; informant's role.
Requirements and use cases	How were they defined? Who was involved? Check for accuracy.
User interface	How were look and feel defined? Who was involved? Positive and negative influences; check for accuracy.
Testing	What was tested? Goals of testing; time of testing; usability tests.
Personal assessment	Relationship between RUP and UCD; amount of UCD; behaviour under time and budget pressure; relationship between clients and usability; involvement of end users.
Project management discipline	Number of iterations; when were end users involved? Measurement of user satisfaction; measurement of client satisfaction.
Business modelling discipline	Elaboration of requirements with clients and end users; on-site visits; same workers as in other disciplines?
Requirements discipline	Ease of use as explicit requirement; elaborations with end users or clients; what are advantages and disadvantages of such a process? Different levels of requirements; explicit requirements for visual design of the UI.
Analysis and design discipline	Which artifacts resulted from this discipline?
Implementation discipline	The same employees as in other disciplines? Was implementation based on use cases? Prototypes.
Test discipline	What was tested? Tests against requirements or against use cases? With end users? When was tested? Consequences of tests.
Configuration and change management discipline	How was it handled? Decision processes; consideration of changes for the whole application.
Environment discipline	UI guidelines; business modelling guidelines; use case modelling guidelines; reasons for guidelines; did people follow the guidelines?
Deployment discipline	Education of the users; technical writer; client support.
Influences	Management; technology; customer.

We did not evaluate the success of the products elaborated in the projects that we investigated because we concentrated on the problems of user-centred design. We believe that this method leads to usable products, but we are aware that possibly methods other than UCD can lead to usable products, too.

We did not rate the expertise of the participants and relied on their self-assessment. More experienced professionals would maybe act differently and face other problems or no problems at all.

Last but not least, we as researchers might have biased the results by our values and preconceptions. To let the reader judge how we might have influenced the results, we show our prior understanding why UCD is or is not used.

Self-Referential Design. Due to lack of knowledge about users, team members work in a self-referential manner and design software for themselves, rather than for users.

Tight Conditions. It is not possible to conduct all activities necessary for user-centred design because time and budget allocated to projects are tight.

No User-Centred Software Engineering Method. The software engineering method used is not user-centred and does not provide for extensive involvement of users.

Advanced Technology. The main advantage of the product is in its high-end technology, not in the ease of use.

Awareness of Decision Makers. If the decision makers are aware of the importance of user involvement, then the amount of time and money spent on user-centred activities rises.

6. Results

We identified software developers, project managers, and UI specialists as the most important stakeholders in creating user interfaces. We group the findings around the three main roles and describe the results for each role in a separate section. For each role we describe the work context, motivation and work practice as seen in the four investigated projects. We then name the strategies used by each role to overcome the obstacles. In the end we evaluate for each role how the identified factors hindered or supported UCD.

6.1. Software Developers

6.1.1. Work Context. The main task of software developers in the four user-centred projects was to implement a system according to a specification. We refer to software developers as engineers who design and code computer programmes on both application level and component level. The good reputation of software developers depended on delivering good code in a short time. Even though the projects were declared as user-centred, usability issues were played down.

Developers are interested in technology and are happy not to have to decide about colours.

(Project manager, company A.)

During informal testing, usability problems were not entered as errors, but as change requests.

(User representative, company B.)

6.1.2. Motivation. The main motivation of developers was to handle programming complexity in a timely manner. Their efficiency in terms of amount of work and invested time was directly reflected in the costs of the project. Project success was more important to the developers than product success, since the long-term quality of the product (e.g., usability or maintenance) comes to light only very late. The success or failure of the project was much more evident in the short term.

6.1.3. Work Practice

- (1) Some decisions were made in the team or in the company, for example, decisions about technology or frameworks, so the developers did not have any influence on them. Even if they were not well suited for the problem at hand, there were standards that had to be maintained. For example, in one project there was a policy to develop a web application, which was not the optimal platform.
- (2) Contact to different stakeholders was tunnelled through project managers or UI specialists; thus developers had no contact with users. Yet, developers strongly influenced the product by small decisions they had to take on a daily basis. They often proposed ways to handle user interaction, and if no one objected, their proposals remained with in the product.
- (3) Developers preferred to work independently and not to bother clients, supervisors, or colleagues with their problems. They relied on their experience with other products and projects, copied widely accepted or similar products, and reused existing components.
- (4) Developers followed the guidelines, if there were any.
- (5) To be efficient in terms of amount of work and costs, developers did only what really needed to be done and avoided duplicate work.

- (6) As stated earlier, the developers' main task was to write code based on a specification. In the specification, the UI was often predefined. This was perceived ambivalently. It was positive because it helped to keep the amount of work low. It was negative because a too detailed specification was discouraging to the developers. It hindered the creativity of team members other than UI specialists and developers *could feel as machines* (project manager, company D).
- (7) Testing was not done systematically, and under pressure it was left out altogether. In addition, only what was specified could be tested. Thus, if test cases did not focus on ease of use, usability stayed unverified.

6.1.4. Strategies to Overcome the Obstacles. Software developers did not consider the projects to be different from other non-UCD projects that they participated in. We have not identified any strategies of developers to overcome the obstacles to UCD, because they did not distinguish between user-centred and non-user-centred projects.

6.1.5. Influences on UCD. We state how each work practice from Section 6.1.3 influences UCD. Team decisions (1) can support or hinder UCD, depending on whether the decisions support the current situation. Using existing components and following external well-known examples (3) can support or hinder UCD, depending on how suited they are for the task at hand. If they are well suited, then they support UCD, because well-established components are probably well thought out and familiar to users. Developers tend to work independently (3) and efficiently (5) and have scarce contact with end users (2). These practises hinder UCD because they promote self-referential design. If the UI is predefined (6) and does not have to be changed later, then it supports UCD, because it will probably be consistent as a whole and the same underlying concept will be kept even if different developers work on different parts of the system. If there are late changes, a predefined UI hinders UCD, because developers are reluctant to change the code once it is programmed. Following sensible guidelines (4) supports UCD because guidelines ensure a high recognition value and consistency of the products. Testing (7) supports UCD if the specification includes usability aspects. However, if usability requirements are not specified, then testing can give a wrong impression of the quality of the product.

6.2. Project Managers

6.2.1. Work Context. We refer to project managers as the employees who are responsible for the planning, execution, and closing of a project. Thus, they focused on organising the project and surveying its progress. They prioritised and controlled the amount of work spent on different tasks. They kept an eye on the UI, because this was what the customer saw. The contract had to be fulfilled, so the project managers invested resources to achieve high usability only if the customer had asked for it.

6.2.2. *Motivation.* Good reputation achieved through keeping the projects on time and within budget was the main driving force of project managers. Due to lots of contact with clients, project managers saw not only the project success but also product quality as important. If the customer was satisfied, this allowed introducing the product to a wider user base. This helped the project manager to achieve a good reputation inside the company, but it also helped to promote UCD at least in cases where the customer put emphasis on it.

6.2.3. *Work Practice*

- (1) Project managers tried to fulfil the official contract by controlling work progress, controlling adherence to guidelines, and keeping the project on time and within budget. Staying on time and within budget included prioritising tasks, deciding on variants, and doing only what really needs to be done and thus not doing too much usability work:

There is a danger. Usability aspects are never finished and one must be well versed in deciding when to stop.

(Project manager, company A.)

- (2) Project managers needed to have an overview of the situation and shared it with other team members: ideally everybody was supposed to know what was going on, nothing was supposed to go astray, and everything was supposed to be traceable from high-level requirements to the implementation code and vice versa. Different team members worked on different tasks, but the project manager had to have an overview of the whole application. The project manager needed to decide many things and therefore had to *feel* what is going on, because not everything could be expressed as requirements. Often also the UI specialist had the role of evaluating and understanding the project situation as a whole.
- (3) Mostly only the project manager or the UI specialist had ties to the customer world, and there was no direct contact between developers and clients. This was in contrast with the wish that everyone in the team should be well informed. The project managers overcame this difficulty by informing the team members.
- (4) The contact with the client was not systematic and was often used for marketing purposes as well.
- (5) Prototypes helped clarify the needs and progress of the project, because they enabled efficient communication with the customer.
- (6) Attitude towards usability testing was pragmatic:
 - for a rich client [few users]: it would be ridiculous [superfluous], we would not keep to the budget. For a web client [many users]: it would also be ridiculous, because you do not have to keep the users with your application. They simply just have to use it!

(Project manager, company B.)

- (7) If usability was not explicitly requested by the client, then it did not count as a quality metric of the product.
- (8) The project managers put emphasis on independent work and internal problem solution by not bothering others: a user survey for every problem was not feasible. Therefore, self-referential design by all team members, external role models, and implicit knowledge influenced the design of the UI.
- (9) Following guidelines helped achieving better consistency in the product, but project managers had to ensure that they are followed. Sometimes, complying with guidelines was seen as a sufficient effort to achieve high usability.

6.2.4. *Strategies to Overcome the Obstacles.* To overcome the problem of scarce contact between project team and end users or clients, project managers tried to pass the knowledge they had about the customer to the project team. Controlling that the guidelines were followed was another strategy of project managers that helped promote UCD.

6.2.5. *Influences on UCD.* We state how each work practice from Section 6.2.3 influences UCD. Adhering to the official contract (1) can help or hinder UCD, depending on what it states. If high usability is explicitly asked for, then keeping the contract supports UCD. Otherwise, the additional effort to achieve high usability does not pay, because the usability does not count as a general quality of the product, so keeping the official contract becomes an obstacle to UCD. If only the project manager has the overview of the whole application (2, 3, 4) and filters the information according to his own needs, then this hinders UCD. Other team members are forced to make some decisions independently (8), but they cannot make informed decisions due to lack of complete and adequate information. A negative and depreciating stance towards usability (6, 7) hinders UCD. Prototyping (5) and adherence to guidelines (9) support UCD because they enhance consistency. In addition, prototyping (5) allows efficient communication with the client.

6.3. *UI Specialists*

6.3.1. *Work Context.* We refer to UI specialists as employees who concentrate on the user interface of products in a broad sense: they assessed and made recommendations to improve usability of products (usability engineering) and/or design the interaction with the system (interaction design). The main task of the UI specialists was to analyse the requirements from the user perspective and propose an adequate UI. Their effort to work in a user-centred manner was supported if the customer explicitly wanted high usability. Otherwise they had to persuade the project manager that the efforts invested in ease of use would pay. Customers were external contractual partners or internal clients in cases of in-house software development. High usability was an important sales argument in three cases: first, when there was no difference to competitors' technology and the only advantage could be

achieved through a better UI; second, if product was for a wide user base; third, if product was for poorly educated and thus low-paid employees. Usability was perceived as important for consumer goods, not for investment goods.

6.3.2. *Motivation.* The main motivation of UI specialists was to produce a good UI and thus gain or keep a good reputation as UI specialists. They were attached to the UI that they elaborated and followed up on it even when they were no longer involved in the project.

6.3.3. *Work Practice*

- (1) No systematic approach to realise usability work was taken, even when usability was explicitly asked for. Usability tasks were mostly an appendage to marketing; for example, contact to end users happened on marketing events, fairs, or visits to distributors. The knowledge about users was often based on data from market research.
- (2) UI specialists learned about users through user representatives. User representatives were heads of departments, customer service, support, internal testers, sales persons, product managers, and even developers:

Everyone is a user. I as a developer see also what is cumbersome. System testers also see the user perspective.

(Developer, company C.)

Especially valuable were innovative clients who were open to new technologies and liked to experiment. They were few but useful for getting feedback about new products. Input from real users came normally only after the product launch.

- (3) Besides piggybacking on marketing, another way of conducting usability work was as an appendage to SW engineering. For example, if real contextual enquiry was not possible, then meetings with users were appended to normal project meetings instead. Another example of usability being dependent on SW engineering is that some UI elements arose by chance during development and were reused if they proved useful (see point (2) from Section 6.1.3).
- (4) Contact between members of the team and users was scarce. The reasons were manifold: it was not known from the beginning who the end users would be, they were difficult to reach due to geographical distance or different languages, clients did not have time to invest in the project, detail questions were of no interest to them, or they did not know the answers to problems either.
- (5) Prototypes developed by UI specialists were an efficient means of communication with customers and were on a level that the customer understood. Presentations of prototypes were seen as valuable because they helped marketing purposes and boosted

familiarity of the client with the system. Prototypes or drawings of the future product were highly valued also because they served as a proof that the UI specialist was being productive. During the requirements phase they helped different people to come to a common understanding and helped the discussions between the customer and the engineering department. Prototypes helped to politically persuade the decision makers, to simplify the product and to ensure consistency. See point (5) from Section 6.2.3 for the project managers' point of view.

- (6) UI specialists designed the UI, but they had to take care not to specify too much. Overspecification was discouraging to the developers. On the other hand it helped to avoid rework; see point (6) in Section 6.1.3 for the developers' point of view.
- (7) Direct contact with end users was not highly valued by the UI specialists. Several UI specialists even said that communication with end users was not necessary, because anyone can be a user, system testers and developers in particular.
- (8) Experience of team members was an important source of knowledge and was more highly valued than contact with end users. Experience with the product under construction and with similar products constituted domain knowledge.

A good mix of experience and innovation is necessary.

(Project manager, company B.)

Furthermore, project experience in which problems to address first was seen as more important than direct contact with end users.

Innovation or experience... Basic set-up should stay the same, but internally you change to a new technology. When several things have become better, then they [users] are delighted.

(User representative, company C.)

- (9) Usability work was perceived as unrewarding. Usability was only seen when it was missing.

If usability isn't correct, then the upper management has something to say about it, when it is correct, then it is not of interest.

(UI specialist, company B)

If feedback from end users is that *The application is very easy* then this is disappointing, because the whole complexity of the work is not seen. At the same time it is the biggest compliment.

(UI specialist, company D.)

- (10) The only features tested were those that have been considered in the specifications.

- (11) Explicit usability testing happened internally with user representatives in an informal manner or externally by a specialised company. External usability testing served to get a neutral opinion and was conducted late when the application was in a stable form. At this stage not many changes could be introduced into the product.
- (12) Tests with customers for acceptance of the product served the purpose of finishing the project and could not be used as input for improving the product.

6.3.4. *Strategies to Overcome the Obstacles.* The strategies of UI specialists to overcome the obstacles were to work with special clients who were willing to experiment with new products, testing at official events with customers and informally with colleagues. Prototypes served as a visualisation of ideas for different stakeholders, brought common understanding, and were used for informal tests with colleagues. The development of prototypes was the biggest promoter to UCD. Also the UI specialists' emotional attachment to the product even after their engagement in the project helped the continuous focus on users' needs.

6.3.5. *Influences on UCD.* We state how each work practice from Section 6.3.3 influences UCD. Usability efforts are appended to either marketing (1) or software engineering (3). This hinders UCD, because their goals differ and no real emphasis can be put on usability tasks. Scarce contact with end users (4, 7) hinders UCD because feedback from real users is missing. Working with user representatives (2) is a workaround to this problem, but user representatives have different goals than users and cannot provide the same information. This workaround gives the wrong impression of working in a user-centred manner. Reliance on experience rather than on contact with end users (8) hinders UCD because input from end users is missing. Prototyping (5) and testing (10, 11, 12) help UCD because different solutions can be elaborated and verified with the users or clients. Prescription of UI (6) supports UCD because it helps achieve consistency of the application. However, it can hinder UCD if late changes need to be undertaken, because developers are reluctant to change the code. Depreciating usability efforts (9) hinders UCD, because it is demotivating to the team, especially to the UI specialist.

7. Discussion of Results

End users did not participate in the projects. They were consulted in some cases, for example, on trade fairs, or represented by substitutes, for example, marketing representatives or testers. Most participants regretted that end users could not be more involved in the projects, but no one made greater efforts to involve them. UI specialists relied more on their own experience and understanding of the products than on input from end users. We identified the difficulties of contacting the end users due to geographical distance or language barriers as reasons for such behaviour. However, it is also possible that the UI specialists doubted

the contribution of the end users to the product and therefore did not insist on more contact with end users. This interpretation would support the finding by Buschermöhle et al. [22] that too much user involvement is harmful. They defined the optimal range of effort on user involvement to be between 10% and 24% of total effort. Heinbokel et al. [23] even identified user participation as an obstacle to SW engineering: it resulted in low flexibility, few innovations, and low team effectiveness. Keil and Carmel [24] recommended that users and developers be in touch, but not too excessively. They suggested four to six personal links between users and developers. In our projects, team members used data from marketing research or consulted colleagues from other departments instead of working directly with end users. They relied more on the existing sources such as general knowledge, marketing data, their own experience, or other systems than on information they would have gathered from the end users.

Marketing, support and others inform themselves on business of the future, on competitors, and so forth.

(Usability specialist, company B.)

Projects in our study were conducted in environments with no strategic support for UCD. Gulliksen et al. [25] have identified twelve key principles of user-centred systems design. The key principles were developed from a large number of projects so they are well anchored in practise. We contrast our results with those key principles to ascertain how they fit with the practise in environments with no strategic support.

In environments with no strategic support five of twelve principles were met (*simple design representations, prototyping, explicit and conscious design activities, professional attitude, usability champion*), six were not met (*user focus, evaluate use in context, active user involvement, evolutionary systems development, process customisation, user-centred attitude should always be established*), and one was not applicable (*holistic design*). Principles that were met are all executable by single persons, whereas the principles that were not met or were not applicable deal with support of the whole organisation. This is on one hand not surprising, because the environments were without strategic support. On the other hand it supports the finding by Rosenbaum et al. [19] that usability methodologies are more strategically effective than organisational approaches. Usability methodologies can be conducted by one or a few persons, whereas organisational approaches need involvement of bigger teams over a longer period of time.

Wixon [16] has argued that usability methods should be analysed in terms of whether they encourage participation, buy in, and collaboration by the development team. We identified a further essential characteristic. In user-centred design, it is normally assumed that nothing or very little is known about users and their context of work. Users' context of work and preferences need to be investigated as a first step towards a usable system. However, most products today are further developments of what already exists, and a lot of knowledge exists in the company. Product managers, marketing professionals, and support teams are indeed well

knowledgeable about their products. UI specialists in our study utilised this knowledge heavily. What is missing in the current user-centred design are techniques on how to utilize this knowledge for usability work.

Motivation of the different stakeholders emerged as important during the interviews. Our results show that different stakeholders had different motivations during the projects. An alignment of their motivations might lead to a better cooperation and thus to better results of projects. For example, timely elaboration of the UI by the UI specialists may relieve the programmers from dealing with such distractions as positioning the buttons on the screen. Another example of alignment could be the development of prototypes. Prototypes can serve the project manager for more effective communication with the customers.

For the future we suggest the following remedies.

- (i) Developing methods on how to integrate existing knowledge about a product from product managers, marketing professionals, support teams, and other staff with usability aspects. Techniques on how to extract facts relevant to usability and interaction aspects from available information should be established.
- (ii) Building a pool of personas for a company as suggested by Khalayli et al. [26], such that UI specialists can fall back on them when they are denied access to real end users. This way, if the market is not changing too often, the most important characteristics of end users can be taken into account and independent work of UI specialists is ensured.
- (iii) Lobbying on the customer side should strengthen users' and clients' right to get highly usable products. As we have seen, if the clients put emphasis on usability, the development team can find means to work in a user-centred way, but without an explicit request for high usability, the development team has no incentive to work in a user-centred manner. Rosenbaum et al. [19] in their analysis of obstacles to strategic usability found that one of the most often cited obstacles call for more education of clients and partners. If customers are not demanding greater usability, development organisations do not have much interest in investing in usability.
- (iv) More project stakeholders should be educated in standard UI design and usability. The standard body of knowledge on common platforms can be reused and the major flaws in UI design can be prevented.

8. Conclusions and Future Work

We have analysed the work context, motivations, and work practices of practitioners in four Swiss RUP projects working in a user-centred manner in environments with no strategic support for UCD. We have shown strategies used by the practitioners to overcome the problems and analysed how different work practices influence the user-centredness of projects. We identified remedies that can help future projects

in mitigating the problems faced in practice of user-centred design in environments with no strategic support for UCD.

In the future, we would like to see our study replicated and our findings tested in other settings where user-centred design is not the method of choice for developing software, but rather isolated teams or team members try to apply user-centred design. Other examples where we would like to see our study replicated are situations where (1) confidentiality is important and users cannot be involved beforehand or (2) company does not make money with usable products, for example internal software development where employees must use the software anyway and there are no visible benefits from usable products. Furthermore, we would like to test whether our suggestions for improvement can be implemented in practice and which results they would yield.

Professionals in the projects preferred to work independently and utilized existing knowledge. They did not rely on what single persons (users) said, but on other sources. In the future we would like to experiment with the credibility of the sources: what is more reliable, a statement of a single person, a statement supported by many persons, marketing data, experience of experts, or other systems?

A long term goal is to investigate how existing knowledge from staff can be used in usability engineering.

Acknowledgment

The authors thank the participants of the study and the reviewers of the paper for their contribution to this research.

References

- [1] UPA. Welt usability tag charta.
- [2] S. W. Draper and D. A. Norman, "Software engineering for user interfaces," in *Proceedings of the 7th International Conference on Software Engineering*, pp. 214–220, IEEE Press, Piscataway, NJ, USA, 1984.
- [3] H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*, Morgan Kaufmann Publishers, San Francisco, Calif, USA, 1998.
- [4] A. Cooper and R. Reimann, *About Face 2.0*, John Wiley & Sons, Chichester, UK, 2003.
- [5] L. L. Constantine and L. A. D. Lockwood, *Software for Use*, Addison Wesley Longman, Reading, Mass, USA, 2000.
- [6] International Organization for Standardization, *Human-Centred Design Processes for Interactive Systems*, Geneva, Switzerland.
- [7] L. Vukelja, L. Müller, and K. Opwis, "Are engineers condemned to design? A survey on software engineering and UI design in Switzerland," in *Proceedings of the 11th International Conference on Human-Computer Interaction (INTERACT '07)*, vol. 4663 of *Lecture Notes in Computer Science*, pp. 555–568, 2007.
- [8] G. Coleman, "An empirical study of software process in practice," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05)*, p. 315.3, IEEE Computer Society, Washington, DC, USA, 2005.
- [9] B. Bygstad, G. Ghinea, and E. Brevik, "Software development methods and usability: perspectives from a survey in the software industry in Norway," *Interacting with Computers*, vol. 20, no. 3, pp. 375–385, 2008.

- [10] G. Venturi and J. Troost, "Survey on the ucd integration in the industry," in *NordiCHI*, R. Raisamo, Ed., pp. 449–452, ACM, 2004.
- [11] M. Peissner and K. Röse, "Usability engineering in germany: situation, current practice and networking strategies," in *Proceedings of the 1st European UPA Conference on European Usability Professionals Association Conference*, pp. 105–114, British Computer Society, Swinton, UK, 2002.
- [12] N. Moha, L. Qing, A. Gaffar, and A. Seffah., "Enquête sur les pratiques de tests d'utilisabilité," in *Proceedings of the the 17th International Conference on Francophone sur l'Interaction Homme-Machine (IHM '05)*, pp. 115–122, ACM, New York, NY, USA, 2005.
- [13] P. Kruchten, *The Rational Unified Process: An Introduction*, Addison-Wesley Longman, Boston, Mass, USA, 2003.
- [14] J. Gulliksen, I. Boivie, J. Persson, A. Hektor, and L. Herulf, "Making a difference: a survey of the usability profession in sweden," in *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction*, pp. 207–215, ACM, New York, NY, USA, 2004.
- [15] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," in *Proceedings of the Conference on Human Factors in Computing Systems*, vol. 4, pp. 471–478, ACM, 2002.
- [16] D. Wixon, "Evaluating usability methods: why the current literature fails the practitioner," *Interactions*, vol. 10, no. 4, pp. 28–34, 2003.
- [17] A. Følstad, I. Bark, and J. Gulliksen, "How hci-practitioners want to evaluate their own practice," in *Proceedings of the 4th Nordic Conference on Human-Computer Interaction*, pp. 417–420, ACM, New York, NY, USA, 2006.
- [18] T. Uldall-Espersen and E. Frøkjær, "Usability and software development: roles of the stakeholders," in *Proceedings of the Human-Computer Interaction. Interaction Design and Usability*, Lecture Notes in Computer Science, pp. 642–651, 2007.
- [19] Stephanie Rosenbaum, Janice Anne Rohn, and Judee Humberg, "A toolkit for strategic usability: Results from workshops, panels, and surveys," in *Proceedings of the 36th Hawaii International Conference on System Sciences (CHI '00)*, pp. 337–344, ACM, 2000.
- [20] R. Yin, *Case Study Research: Design and Methods*, Sage Publishing, Beverly Hills, Calif, USA, 2nd edition, 1994.
- [21] B. G. Glaser and A. L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Aldine, Chicago, Ill, USA, 1967.
- [22] R. Buschermöhle, H. Eekhoff, and B. Josko, "Success: erfolgskfaktoren aktueller it-projekte in deutschland," *Objekt Spektrum*, pp. 42–47, 2006.
- [23] T. Heinbokel, S. Sonnentag, M. Frese, W. Stolte, and F. C. Brodbeck, "Don't underestimate the problems of user centredness in software development projects—there are many!," *Behaviour and Information Technology*, vol. 15, no. 4, pp. 226–236, 1996.
- [24] M. Keil and E. Carmel, "Customer-developer links in software development," *Communications of the ACM*, vol. 38, no. 5, pp. 33–44, 1995.
- [25] J. Gulliksen, B. Göransson, I. Boivie, S. Blomkvist, J. Persson, and Å. Cajander, "Key principles for user-centred systems design," *Behaviour and IT*, vol. 22, no. 6, pp. 397–409, 2003.
- [26] N. Khalayli, S. Nyhus, K. Hamnes, and T. Terum, "Persona basedrapid usability kick-off" in *Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI '07)*, pp. 1771–1776, ACM, New York, NY, USA, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

