PD Method and Socio-Political Context of the Development Organization

Kari Rönkkö
School of Engineering
Blekinge Institute of Technology
Soft Center
SE 372 25 Ronneby, Sweden
Kari.Ronkko@bth.se

Mats Hellman
Product Planning User Experience
UIQ Technology AB
Soft Center VIII
SE 372 25 Ronneby, Sweden
Mats.Hellman@uiq.com

Yvonne Dittrich
Software Development Group
IT University of Copenhagen
Rued Langeardsvej 7
DK 2300 Copenhagen S Denmark
ydi@itu.dk

ABSTRACT
Little research is done on how socio-political factors intertwine with method implementation. This paper is a follow up of a reported PD method implementation failure four years ago. For branch related reasons, the implementation fell short. This article tells the story of what happened then. To our surprise, the original reason for introducing the PD method called Personas had disappeared during our one and a half year attempt at method implementation. Internal socio-political developments had solved the power struggle that the PD method was aimed to mediate. We propose that it is time to expand the method focus with studies revealing more about the socio-political internal climate and contingencies in today’s software development practices that influence method introduction and adaptation.

Keywords
Participatory design, method, personas, socio politics, research methodology, industrial cooperation, interaction design, software engineering

INTRODUCTION
This article illustrates how an internal re-organization in a company together with the development of a new product-concept solved an interaction design team’s internal problems; originally a PD method, the Personas method [5 and 8], was designed to solve these problems. Hence, the internal socio-political development did the work of a PD method. In general, little research has been done to show how socio political aspects of the development organization influence methods implementation and vice versa. In software engineering the focus is often on uncovering universal best development practices from a technical viewpoint [18]. Researchers in Systems Development research have been concerned with social aspects to a greater extent than the researchers in Software Engineering. Still an extensive separation of concerns has resulted in intertwined practice where related issues of the type represented in this paper are being missed out [17, p. vii].

In PD, substantial work surrounding method development and support for IT designers has been published. For example, Kensing et al. have developed method support for initial design of complex work in systems development projects [12 and 14]. In relation to continued design in use, Henderson and Kyng [10] discuss additional demands. Many PD projects also report on how the researchers themselves take the role of being IT designers, often in situations where it is a onetime experience for themselves and the users [3].

Kensing et al. have steered their efforts in another direction through supervision of industrial actors. They hope to establish long-term adoption of PD methods [13, 4 and 14]. It has been suggested that PD is defined by a commitment to worker participation, and the re-balance of power relations between users and technical experts, and between users and managers [15]. In the latter, more modest ambitions (the choice of wording is borrowed from the authors themselves), Kensing et al. have long strived to influence the work practice of IT designers. In this way, they are convinced to accomplish both PD method integration and -sustainability in IT organizations [14].

Toerpel et al. [27] argue, based on empirical research in for PD untraditional settings, that the organizational context of development and use influences the applicability of participatory design methods.

In a similar way research results from empirical studies indicate that there may be good organizational reasons for ‘bad’ development practice. Often methods and project models are applied in a way mirroring situational rationale. (See [2, 16, and 22] for examples of this phenomenon.)

Suchman proposes in her article on ‘Working relations of Technology Production and Use’ [26] that design of information technology always takes place in a specific context and builds on existing technology. Thus, design has to be understood as ‘artful integration’ rather than ‘design from nowhere’. The same has to be claimed for method design. The introduction of methods interacts with the specific context of development and use. Few articles take up how methods actually inform development practice.
Mathiassen [17] refers to methods as a learning vehicle that becomes unnecessary when the members of the team become more expert. Kensing and Munk Madsen [25] describe methods as representations and procedures mediating mutual learning and knowledge integration between domain experts and IT professionals. Grudin [7] emphasizes the political aspect of methodologies, when proposing that e.g. personas provide legitimacy for the introduction of usability and participatory agendas into product development. Though this work is based on rich experiences, little empirical work is reported on the impact of methods on concrete development practices. Despite many different kinds of reported successful PD method implementation results, the socio-political developments in industrial practices, as reported in this study, are easily missed out when a technical, systems, or method development perspective is applied. Our study indicates that understanding the organizational context – not only in terms of method implementations, but also as company internal power struggles inherent in the end user related IT design and development processes – is of importance if we wish to fully grasp what participatory, use-oriented and useable systems development is about [18].

The study in this paper took place in a company developing applications and a user interface (UI) framework for high-end mobile devices. To provide support to communicate the result of participatory design processes between interaction designers and users to software engineers, the interaction design team (ID-team) we were cooperating with proposed use of the personas method. The implementation of personas was unsuccessful because of the specific way the development depended on external actors. Revisiting the problems of the ID team showed that the problems were solved by organizational development.

This study is of interest to the PD community precisely because it expands the method focus by addressing the socio-political internal climate and contingencies in today’s telecommunication branch. It provides an example of how an internal socio-politic development can both propose, as well as dismiss, the need for a PD method. From another perspective, it also provides example of an organizational situation where a PD method might not be the most adequate solution. The example provokes questions regarding the interaction between the development context and the implementation of methods: What is actually responsible for the success or failure of a method? We believe that internal socio-political developments – such as those presented in this paper – are not uncommon in industry. The influences of socio-politic development on methods usage are huge. At the same time, we know little about how this intertwining plays out. It is a challenging subject as it takes sensibility to discover, discuss, handle and publish.

The article is based on a long-term case study applying Cooperative Method Development, an approach that applies Participatory Design principals on method development. [6] In such a cooperation, the academic researcher contributes reflections on practitioners’ work and academic surveys to enrich discussions of chosen subjects. The people studied contributed professional knowledge, work experience, and a context for theory building. Interpretive ethnographic analysis of studies provides accounts from the studied practitioners’ own point of view as a basis for the development of method improvements; the latter is performed together with the practitioners involved. Suggested methods are tried out in industrial practice by the people from industry. Hence, the researchers do not take on the role of being the IT designer, and similar to Kensing et al.’s efforts, the researchers rather strive to influence the work practice of the IT designers.

The following section briefly introduces the company and the ID-team. We then discuss the research approach and methods in detail. The section that follows provides a walkthrough of the events leading up to the re-investigation of the communication problems between the ID team and the software engineers. Then we detail the results of this re-investigation. The discussion relates the results back to the research questions raised in the introduction. The conclusion sums up the findings.

**THE COMPANY AND THE INTERACTION DESIGN TEAM**

UIQ Technology AB is a young, international company. It was founded in 1999, has today more than 320 employees in Sweden, and 380 employed in total. It is an owned subsidiary of Motorola and Sony Ericsson. The company develops and licenses a user interface platform for mobile phones using the Symbian OS. The product, UIQ’s user-interface platform enables mobile phone manufacturers to create different kinds of phones for different market segments - all based on one single codebase. UIQ is pre-integrated and tested with Symbian OS. The company develops and licenses user-interface platforms for mobile phone manufacturers. Through its independent position (i.e. it is not directly tied to a specific phone manufacturer) the company promotes the introduction of new advanced mobile phones onto the market. Its main assets are technical architecture, a unique independent product, and skilled, experienced staff. Currently there are 20 phones that reached the market based on UIQ: the Sony Ericsson P800, P900, P910 and P1i, G700, G900, W960i, BenQ P30 and P31, Arima U300, Motorola A920, A925, A1000, Z10 and Z8, to mention some. [24]

As the company’s product is a user interface platform and the interaction design (ID) team comprises the main actor discussed in this paper, it is useful to give a brief overview of the ID team and their approach to usability. At the point in time when the methods implementation was attempted, the ID team consisted of one manager, a user researcher, and seven interaction designers. Most of the ID team’s time was devoted to handling continuous ongoing interface prototyping and ratification of the products’ final UIQ
design. The development takes part in projects resulting at the same time in a product for one or more clients – who often also are owners – and evolution of the platform and the corresponding guidelines. In average a development project lasts for approximately one year. In these development projects, interaction designers devote most of their energy to the initial stages of the software development process. The evaluation of the design later in the process takes second place. All interaction designers took part in social and environmental end-user studies that took place parallel to the projects. The extent to which each was involved depended on the ongoing project cycles. Despite the fact that interaction design people constituted ten percent of the total staff (at that point in time the company had 130 employees in Sweden) there was no time for months-long exclusive and extensive user investigations.

The user researcher worked full-time with both long- and short-term user groups. Advanced mobile technology was lent to long-term users to facilitate monitoring of different ways of using the technology under realistic circumstances. With this approach the company gained access to trained end users who could provide details of how the technology worked in different environments. It was also a way of learning about different stages of usage, from that of a novice to an experienced user in the user’s own environment and context. Understanding the differences between the novice and the experienced user in his/her own context together with observation of users in real life situations were prioritized activities within Interaction Design. These efforts were complemented by studies performed by other researchers, and more than one hundred people forming a user network were contacted periodically when the need to discuss specific aspects of mobile usage arose. The users involved in the network were selected in such a way that they represented a broad group of possible users and usages. The users involved were of different ages, gender and social background and their knowledge of mobile phones, and the UIQ platform in particular, varied.

THE RESEARCH APPROACH

As method, we used a research approach called Co-operative Method Development (CMD). CMD takes existing practice of software development in concrete industrial settings as a starting point. CMD combines qualitative research on everyday software development practices, including the cooperation with users and around the usability issues, with the improvement of methods, processes, and tools. The CMD research process is modelled as evolutionary cycles consisting of qualitative empirical research (in our case ethnography), technical and methodological innovation in co-operation with the involved practitioners, and the implementation of these innovations, evaluated through accompanying empirical research. Co-operative Method Development can be seen as a domain-specific adaptation of action research consisting of three phases that can be repeatedly applied in the same context (see [6] for further details).

The idea of implementing personas evolved from early participatory observations conducted alongside ongoing formal and informal discussions between the academic researcher, the first author of the article, and the ID team at the company. During a twelve-month period preceding the ‘personas trial’, the academic researcher joined interaction design meetings and interviewed interaction designers, the usability researcher, sales & marketing staff, managers and software engineers. Observed meetings and interviews were audio taped and partially transcribed. Besides this more formal side of the fieldwork, researcher observed interaction designers’ and software engineers’ work, had many informal method discussions with interaction designers, their manager and the usability researcher, and functioned as one of many test persons of their product design ideas. The researcher also arranged three seminars addressing the personas method, two with an interaction design team and one with the marketing and sales people.

The failure of the method resulted in a joint publication [21] between the involved researchers and practitioners. The second author of this paper is the user researcher. The third author peer debriefed the research of the personas trial and took part in the discussion and writing process leading to the final form of the article. Writing papers together with the people studied is a way of confirming results and encouraging deeper reflection in the company. It can be regarded as a thorough form of member checking. The results presented in the section following the next one are based on informal discussions, taped semiformal interviews, and taped writing reflection seminars with people from the interaction design team.

The socio-political organizational factors presented and the historical developments have been addressed in this study because of the nature of the methodological study and cooperative approach adopted (CMD). A central feature of this methodology and form of cooperation is the long-term relationship between practitioners and the researchers focused on a specific industrial setting. The researcher continuously struggles with keeping a sensitive and open mind focused on what is going on from the studied peoples’ own point of view [9]. In the interaction between practitioners and researchers, the practitioners are continuously mirrored concerning aspects of their everyday practice from their point of view, though not necessarily explicitly spelled out. The motivation for Bronislaw Malinowski, the inventor of Ethnography, to create the ethnographic account was precisely “Things are not what they seem” and appearances are certainly not the whole of the story [1].In this way, the researchers draw attention to things that are already known, as they are parts of everyday practice, but which we as humans are inclined to overlook from our attention when we reason about development methods and method development. Similarly, the
cooperative decision on methodological improvements and the study of their implementation allows addressing the interaction between methods and socio-political influences of the context. Thus, the investigation of the rationale for the adaptation of methods or, as in this case, the failure of the implementation, is part of the research design.

A NOT SUCCESSFUL METHOD IMPLEMENTATION

In this section, we give a short overview over the cooperative project that ended with investigations of the unsuccessful trial of implementing the personas method, and then revisiting the initial communication problems only to find that the problems had disappeared.

The Motivation: Difficulties Communicating ID Results

When innovative interface design was presented to the software engineers who code the end user applications, the members of the ID team at UIQ often ran into internal problems. When coding the applications, the engineers became very engaged in the visual design of the applications. As a result, the software engineers often had strong opinions as well as suggestions for changes in the initial interface design. Arguments arose between engineers and interaction designers concerning the best way to visually present the functionality in the interface. The software engineers sometimes introduced doubts whether the solution proposed by the interaction designers was technically feasible. In those situations, the interaction designers wanted to remain faithful to the engineers’ good intentions, their creativity and questioning, but direct the discussion towards a shared understanding of users and their requirements based on the user studies performed by the ID team.

The notion of a user was used in a rather elastic way, i.e. a general idea of a user, which might be bent and stretched to adapt to any technical contingencies. This too general idea of a user provided the software engineers with a license to code freely according to their own preferences (see [5 p. 127]). The introduction of a set of personas, archetypical users, both concrete and related to the user studies of the ID team, seemed a good approach to help this situation.

A Short Overview over Personas

As a base for the decision, the researcher involved in the project presented an overview of the academic discussion on personas: In the late ’90s, Cooper introduced personas as part of his goal-directed design concept [5]. Grudin and Pruitt applied Cooper’s initial personas trials to the development of mass-market software [8]. Persona is an interaction design technique to enhance engagement with the future users and a realistic understanding of the use context. A persona is a fictional person with a name, possessions, habits, clothes, family, friends, occupation, and pets. A persona has a specific age, gender, ethnicity, goals, life story, educational achievement, and socioeconomic status. Normally, one to three personas are recommended for one application. One persona represents a selected set of end-users and includes several different documents based on an identified set of users sharing the same end-user goals. It provides a means to communicate both qualitative [19] and quantitative [23] data from user studies. Often a mixture of quantitative and qualitative user research methods is used.

Personas have been shown to reduce cognitive friction in applications, restrict elastic user-views allowing free code, provide precision in the definition of a user-group, provide a shared base for communication, and scenarios for recognizing and challenging political and social assumptions about users [8, 19 and 7]. Personas has also been advocated by an experienced and knowledgeable actor as a participatory design tool in the mass-market context [7].

In design practice, many personas can be presented that later are graded in importance for design decisions. A foundation document is recommended for each persona as a storehouse for all information about the persona (data, key attributes, photos, reference materials, etc). Knowledge about the personas used can be spread in a variety of ways: posters, flyers, handouts and giveaways. A web site is recommended that includes foundation documents, research efforts, and tools for applying the personas (posters and photos, spreadsheet tools, comparison charts, etc) [19]. It is the detailed human attributes and characteristics described in terms of needs, goals, and tasks that are thought to give personas the precision necessary for design work (See also [20] for an extensive overview of the Persona Lifecycle).

Trials to Implement Personas

After several discussions regarding how to interpret the personas concept, a project proposal addressing the introduction of the method was accepted by management. The project specification included the following definition: ‘A persona is a representation of the most prominent joint characteristics within a specific target group gathered and presented as “a none existing” person but still a person to communicate about. … The result from a test could then be traceable all the way back to the intended target group i.e. the personas specified for a specific project.’ The aim of the method introduction project was to define personas for creating use cases, user case tests and performance testing in a pilot development project as well as to refine the personas for Marketing & Sales (M&S) purposes.

Problems appeared when it came to deciding which personas should be used. The product owner was placed in the M&S Department, M&S marketing evaluations and the long-term product vision are important aspects when deciding which personas to target. However, M&S never decided on a set of personas. After waiting for some time, the interaction design team decided on their own initiative, which personas should be used. These personas were considered as suggestions that would help M&S make their decision. The personas developed included a teenage boy, a young female, and a middle-aged businessman. Application
use cases were created for agenda, messaging and contacts. A performance test case was given high priority, and usability metrics were defined for user-system verification. Wall charts and prototypes of foldable information of the three persona characters were produced. The time frame for the personas project was three months. A final report describing the personas and their related use cases was also compiled. M&S was again committed to the idea of using personas in a pilot project. However, making the decision as to which of the personas to use proved once again to be more problematic than expected.

Despite the fact that M&S had not officially decided on or accepted the suggested personas, a pilot project was outlined for implementing the method. Every development project should start with a coherent set of requirements. In the pilot project, the use of personas was included in the high-level requirements specification. When the initial coherent set of requirements finally reached the project team, the initial circumstances had changed. Two additional clients showed interest in the project. The deadline for the project was prolonged and negotiations started; a variety of opinions and visions had to be handled. The initial coherent picture of requirements was reduced. Component oriented thinking resulted in a diverse and, from a user perspective, fragmented requirement specification. This reduction of the initial scope did not relate to end-users’ needs. The main reason for change was new technology, market, and competition-related issues. It was not possible to get competing clients, who also entered a project at different points in time, to accept the same archetypes. The clients themselves might also already use another kind of user classification category than archetypes in their own internal organizations.

Telecommunications is a market driven software development area. In market driven projects there is no distinct set of users, instead there are potential groups of people who fit an imagined profile of intended users. The elicitation of requirements is mainly managed through marketing, technical support, trade publication reviewers and user groups. Recent study in this area has also revealed that the constant flow of requirements caused by the variety of stakeholders with different demands on the product is an issue; an issue which is closely related to how to write understandable requirements. In market driven companies requirements are often invented based on strategic business objectives, domain knowledge and product visions. (see [11] for overview)

**Giving up on Personas**

When the coherent user view was lost, it became very difficult to apply the personas concept. It was not possible to produce a new coherent view from the requirements left. The fact that negotiations were still ongoing did not make things easier. How did this new fragmented and unstable view relate to the persona Simon (one of the personas in the methods project)? Even if it were possible to relate some requirements to Simon, what about the other requirements that did not fit? Should the unrelated technical requirements be treated as a separate part within the same project?

After these unsuccessful implementation trials it became clear that the personas method was inadequate under the prevailing contextual circumstances. The ID team and the researchers involved gave up on it.

The challenge that ultimately hindered the method implementation was not related to internal reasons; instead it was the challenge of having multiple clients acting in the same project and other influential stakeholders in the background that resulted in the failure of the method implementation. The project was performed in a branch trying to precede usage by presenting new possibilities arising from new technology, not the other way around. The personas method, on the other hand, is a method that starts with end-user goals and leads to technology solutions. At the same time as personas constituted a remedy for an internal diversification of user ideas, the precise identification of a user affected external clients and potential future clients negatively. The personas experiment was closed down. (For a more detailed description and analysis see [21].)

**What Happened to the Communication Problems?**

After giving up on the personas implementation, the researcher revisited the communication problems that motivated the whole method improvement trial: Much to the researcher’s surprise the problems had disappeared.

To understand what happened, a series of interviews was conducted. The findings were discussed in ID team meetings. The following result is detailed in the next section: Reorganization, together with a re-actualized company internal vision, resulted in increased status; a new interaction paradigm and technical platform resulted in a unique knowledge position; and a more perseverant attitude developed based on historical knowledge - did the job the personas method was devised for.

**REVISISTING THE FIELD**

Here we present detailed descriptions of the short overview of the results presented above. The reason for presenting this level of detail is to give the reader a good and adequate understanding for the situation that actually changed the power balance between the interaction designers advocating end user needs and the software engineers coding the suggested applications. The understanding of the socio-political developments is dependent on ‘rich’ description.

**Increased Status due to Reorganization and Re-actualized Company Internal Vision**

The company had decided to change their organizational structure due to the ever-increasing number of employees. In relation to the reorganization, the company’s internal vision of achieving the best UI in the world was re-
actualized. These factors affected the interaction design (ID) team’s status within the company in a positive way. The motivation for the new organization was to facilitate faster responses in a continually changing market and new product requirements dictated by clients. Smaller, faster and more frequent projects place new requirements on the organization and the development model. The effect of the change, not only affected marketing and sales and product management, but the whole organization.

In the old organization, the interaction design team was an integral part of the Development unit. At that time a small number of employees, divided according to technical responsibilities and placed in the Development unit, were responsible for the strategic developmental issues concerning the UI platform. In this organization, the members of the ID team worked with their respective projects and updated the design and the guidelines for the UI platform independently. For the interaction designers the placement of Interaction Design under the Development unit had led to a close relationship with engineers, bringing about fast communication processes. This also incorporated the disadvantage of the too strong user interface design opinions from engineers that were discussed in the previous chapter.

After the re-organization, Development was named Software Development, specialized in various technological areas. Interaction Design is one such specialization area placed under Software Development. New specialists were recruited, such as a configuration manager, technical writer and experienced specialized engineers. People also moved around to fit into the new organization. At an early stage one of the interaction designers moved allegiance and started to work for the new Product Management Department; he remained involved in the development of the one-handed concept (explained in next sub-section). The new organizational placement made interaction design visible as an important expertise on the same level as other technical areas, such as e.g. software architecture.

In parallel with the re-organization, upper management emphasized the company’s internal vision - to create the world-leading user interface platform. For the realization of this vision the ID team contributed important competences. This improved the visibility of the ID team’s competence in the organization and increased their status in relation to the other technical software professions involved in the development.

**New Interface Platform and Interaction Paradigm Resulted in a Unique Knowledge Position**

As result of a strategic decision in the company, the interaction design team had been given the prestigious task of developing a new user interface platform. The old pen-based interaction paradigm should be complemented by a one-handed interaction paradigm. The new interface platform actually enforced the development of a new interaction paradigm. The interaction design team had a long tradition of continuously performing contemporary social and environmental studies. As a result of early curiosity about what one-handed mobile interaction might result in, together with surveys of such existing technologies and end-user tests, the interaction design team had a head start when the official decision to develop a new one-handed mobile concept was taken. That is, they were the most knowledgeable employees in relation to the strategic decision.

In the long run the old and divided organization bore the risk of resulting in an inconsistent design and implementation. To this was also added a need for a general style guide to support the ID team, to meet the new demands concerning information dissemination of the new conceptual designs to the organization. A growing need for design structure and control was identified. A sharper, better developed and more acknowledged project language and model is more necessary in order to discuss, check, plan and re-plan development efforts. One ID member expressed it as: —In a sense a paradigm shift has taken place within the new organization. A new way of structuring our work has been applied together with the introduction of the new one-handed product (Interaction design manager). The individual interaction designer had more design freedom in the old platform.

At the beginning of the project, the aim was to make UIQ’s platform support a one-handed user interface. The interaction designers decided a focus on the challenge of navigation and focus handling. Results were continuously communicated and confirmed with potential clients. As a pilot application for the new interaction paradigm, the contacts application was chosen. At this point in time, the aim was still to test if the new ideas were applicable and to discover issues that had been missed. Some of the interaction details had to be decided based on initial prototypes: How would the navigation work? How would dialogues work and look? How should hardware keys be mapped to the UI? How should specific controls act? How do we design this in a way that it would be usable both for a one-handed UI and a Classic touch screen UI?

The decision to have only one style guide document capturing both ways of navigating complicated the work considerably. The old pen-based and the new one-handed ‘style’ had to be as uniform as possible. The new style guide grew in parallel with the development of the new one-handed platform. As a result, the interaction designers had to continuously revisit and change the old platform style guide as the new one-handed concept grew. All existing product documentation had to be checked and revised. Examples of changes in the old pen-based concept concerned building blocks, controls, pure pen-based concepts, and the navigation metaphor.

One early decision in particular influenced all other interface designs; the entire interaction was to be based on
hardware buttons. Hence focus and tabs became important considerations. The absence of a highlighted focus in the old pen-based platform made button-based navigation difficult. The controls that competed with the use of left and right navigation tabs had to be changed. The left and right buttons on the advanced phone were reserved for choosing between tabs, the up and down buttons for choosing and highlighting the focus. As a result, the interface layout became restricted so that it was not possible to put control commands to the left or right of a basic view. It was necessary to activate a choice to be able to use the features in an application. In such a control state, left and right buttons could be used for activating the specific application’s functions. The navigation metaphor ‘back to base’ in the pen-based platform was also changed. In the new platform, the used application is not closed down as in the former platform. If an application is launched and then left, the initial view and state are resumed when the user returns to it.

Having two configurations per specification meant that more states must be handled. An example for a new state is a highlight that is moved around on the screen; depending on where this highlight is placed, a limited number of possibilities and choices can be made. The introduction of soft keys is another example of a feature that did not exist in the old platform. A phone has a number of dedicated hardware keys available that steer the possible number of actions in the present user interface. The functionality activated with these keys changes depending on the state of the application and is indicated with labels placed at the bottom of the screen.

The design space is much more predefined and regulated by larger building blocks and conditions as described in the new style guide. —Earlier we had an entire soccer ground to act in and place our work on. These days we are closer to rugby football, all lines are pre-drawn and the spaces in which you are allowed to move are much more regulated. This means that it is necessary for us interaction designers to apply an entirely new way of thinking (Interaction Design Manager.)

The interaction designers involved in the development of the new paradigm developed an expertise that provided them with a unique knowledge position —We have new specification templates that only a few of us manage to handle as they have become very complex (Chief User Interface architect). A lot had changed, even from the interaction designers’ perspective. Especially the way of thinking during design; there are many more frameworks that control the design space. With the new frames, a stricter interface layout is achieved, something that is considered to ease the work in a larger organization.

The interaction design team recognized early on that the new platform meant a lot of changes, but could not fully estimate the extent of it. Changes in one part triggered changes in other parts in ways that that were difficult to fully predict and control. In a traditional project, half of the user interface specifications would have to be updated, often only with small changes and add-ons in relation to one requirement at a time. In this project, all specifications were affected, often by a diversity of requirement. More than fifty user interface specifications have been entirely rewritten using a new template that includes both one-handed and pen-based design specifications. More than 35 control specifications, describing layout, conditions and effects of a control, were updated.

Very few people could answer the software engineers’ questions: the system architect and the interaction designers who were involved in the new platform project. Few engineers were knowledgeable regarding the changes that the new interaction paradigm implied for the platform. Software engineers became dependent on the interaction designer’s knowledge; as a consequence, the expectation to receive quick answers from interaction designers grew.

At times, the interaction designers themselves would have appreciated advice. Due to the fact that they were at the top of the knowledge chain regarding the new interaction paradigm, they were left to trust their own skill and intuition. Often there was less time to reflect over design details than they would have liked. One example of such a skill-based decision is the idea of ‘tabs’ which implied that the right and left hardware buttons cannot be used to navigate in the user interface. Another is the change of the meaning of ‘Back’ from its original meaning of ‘to save’ to instead mean ‘cancel’. In such cases there is no right or wrong, the interaction designers had to take a stand and continue based on that design decision. It is also clear that not all decisions could be explained easily as they were not based on definite circumstances. The interaction designers were stressed by the situation of a waiting organization with few knowledgeable people who had a grasp of the entire new platform. Project leaders as well as other members of the company faced the same difficulty to understand the new design and in which direction the new building blocks and guidelines are developing. The status of the product was not fixed, it was still under development. Obviously, the entire organization was put in a situation in which the members depended heavily on the good judgment of a small group of interaction designers.

**Developing a More Perserverant Attitude**

Even before the new interaction paradigm was implemented, the interaction designers had arranged discussions where historically suggested interface ideas were compared to existing designs in the platform. Surprisingly, many of the ideas that software engineers in earlier discussions had argued as ‘not technically feasible’ were found to exist in the present platform. In a meeting, a member of the interaction design team asked —Shall we implement a more perseverant attitude? And encourage the engineers to first analyze the technical difficulty of a design suggestion before putting it down (Interaction Designer).
Historically, interaction designers often went back to their desk and devoted time to thinking about other solutions. Too often, the same engineer came back after some days to the interaction designer and told him/her that the initially suggested design could be implemented. The ID team developed a tougher attitude towards software engineers.

Altogether, the factors presented above provided the interaction designer with a ‘license’ from the organization to act more perseverant towards changing their initial design suggestions. Notice that this technical conflict of technical feasibility was not the reason for applying the personas method. As described earlier, the main reason for personas was differences in opinion as regards what constituted the best usability solutions. This latter technical conflict was a second undesired ‘bad’ development habit that cost both time and energy.

**Summing up the Empirical Findings**

Above, we detailed the organizational changes that changed the power struggle between the ID team members and the software engineers. The re-organization and re-actualization of the company internal vision had given the Interaction Design team recognition as playing a key role in the company. Still, the most powerful factor was the development of the new product. In the ‘old’ situation, new applications and features were continuously added to an already existing body of shared knowledge and standard solutions. Together with the more perseverant attitude, this changed the communication between the ID team and other professional experts to the benefit of the user-centered design.

**DISCUSSION**

In the initial study, the personas method was applied as the solution to a power struggle between IT designers advocating end user needs, versus software engineers’ ideas about interaction design. The method is well known and documented. It has been used by actors in the same context as in our case. It has been advocated as a tool supporting participatory design in mass-market contexts by an experienced and knowledgeable actor. Personas is known to reduce cognitive friction in applications, restrict elastic user-views allowing free code, provide precision in the definition of a user-group, provide a shared base for communication, and scenarios for recognizing and challenging political and social assumptions about users. We had a strong interest in all of these aspects of the method, even though our initial interest was to restrict elastic user-views allowing free code. However, due to branch-specific organization of the business the method was not applicable in the specific context.

Moreover, the empirical sections above show how a change in the organization, the development of a new product, and a different attitude, together solved co-operation problems which had stimulated the trial to introduce the personas method, i.e.:

- Reorganization together with a re-actualized company internal vision resulted in increased status
- A new interaction paradigm and technical platform resulted in a unique knowledge position
- A more perseverant attitude developed based on historical knowledge

Comparing our observations with the purposes of methods discussed in the introduction we can see that the personas method was not necessary strongly related to: learning [17] was never a problem for the interaction designers. They were very skilled in their area of expertise and the personas did not contribute to their development; the need for mediating cooperation and communication between interaction designers and software engineers [25] was satisfied by the new style guide integrating the old pen-based and the new one-handed interaction paradigms; and in the process of internal re-organization in the company, the interaction designers gained recognition and credibility for their importance and role with regards to the company’s existing product, the user interface platform [7]. The ID team is an important unit within the Software Development unit to help develop the broad outlines and guiding principles for the new paradigm and platform. Through the renewal of the company’s internal vision to create the world-leading user interface platform, the interaction design team gained increased recognition and respect.

Neither the failure of the method implementation nor the ‘success’ of the socio-political factors tell anything about the reliability or effectiveness of the personas method. It shows how factors external to the scope of the method hindered its implementation and ultimately even solved the problem the method had initially been devised for. Both results address the interaction between a method and the software development context it is implemented in.

That the described changes did the work that the method was supposed to do raises questions regarding whether it is actually the implementation of a specific method that brings about observed improvement. In few, if any, published success stories of software process and method improvement projects, are any reflections presented on whether other factors contributed to the observed improvements. The same is valid for the Participatory Design literature on methods and their application. We ourselves might not have investigated other factors if the implementation of the personas method had been successful.

For a specific company, one might argue that if the software process improvement costs do not exceed the gains, such reflections are of minor importance. From a research point of view, this is not the case. In order to assess a case for a general application of our results, we have to understand how the methods we devise interact with the context and how they actually bring about the
improvement we observe. And from a long-term methods development point of view, this is also true for industry. Our results point to the need to complement studies of successful method implementation with reflections on alternative sources of improvement. Negative results, not successful improvements, or failures of method implementation, can contribute by indicating starting points for further investigations. We have to complement method centered research with approaches that open up for sensibility to precisely what is hidden when focusing on methods. In our case, the cooperative methods development approach together with the ethnographic underpinning ‘member’s point of view’ provided the base to investigate even the sociopolitical organizational context.

CONCLUSION
In our article, we reflect on a case where practitioners and researchers together tried to implement a method – Personas – devised to improve the co-operation between interaction designers and software engineers. The problems were solved in spite of the impossibility of introducing the method. A closer analysis of the development showed that re-organization, a new product, informal visions and increased perseverance did the job of the method.

Little research is done on how such socio-political factors intertwine with method implementation. Socio-political context in industrial practice is not a common research subject in relation to method development. PD addresses politics in the use organization. Methods are seen as learning vehicles for IT professionals, mediators of cooperative design, or as providing legitimization for use-orientation in development. The latter two purposes motivated the trial to introduce personas but became superfluous with the changes in the organization.

Paradoxically, the possibility that internal socio-political developments might facilitate the same results in industrial practices as methods is common sense knowledge recognized by both researchers and practitioners. Still, it seems to be a subject that is mentioned only incidentally in oral discussions. Reasons for this might be that it is not extensively studied, and that it takes a research approach that is sensible enough to discover, present, discuss and handle such interactions.

Where does this leave PD? We believe it is time to expand the method focus with studies revealing more about the socio-political dynamics and contingencies in today’s software development contexts. We argue that research on method development and implementation must reflect these issues to a larger extent than hitherto. Which community other than PD is better suited to discuss how these kinds of power struggles and socio-political developments influence the possibilities of use-orientation in development?

What are the means to draw attention to aspects that are ‘already known as they are parts of everyday practice’, that we as humans are inclined to overlook when we talk about end-user, organizational, software development and system needs? Our study indicates that the analysis of failures of method implementations or the adaptation of methods might sometimes teach us more than their successful deployment. Research needs complementary approaches that open up for the development of a sensibility to how and why methods, tools and processes make sense in a specific situation.

ACKNOWLEDGMENTS
We wish to thank Product Management and Marketing & Sales for providing valuable insights; special thanks go to the remaining staff in the ID team. This work was partly funded by The Knowledge Foundation in Sweden under a research grant for the software development project “Blekinge - Engineering Software Qualities” www.bth.se/besq.

REFERENCES


