

Participatory Design Activities and Agile Software Development

Karlheinz Kautz

► **To cite this version:**

Karlheinz Kautz. Participatory Design Activities and Agile Software Development. Jan Pries-Heje; John Venable; Deborah Bunker; Nancy L. Russo; Janice I. DeGross. IFIP WG 8.2/8.6 International-Working Conference on Human Benefit through the Diffusion of Information Systems Design Science Research, Mar 2010, Perth, Australia. Springer, IFIP Advances in Information and Communication Technology, AICT-318, pp.303-316, 2010, Human Benefit through the Diffusion of Information Systems Design Science Research. <10.1007/978-3-642-12113-5_18>. <hal-01060407>

HAL Id: hal-01060407

<https://hal.inria.fr/hal-01060407>

Submitted on 4 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



18 PARTICIPATORY DESIGN ACTIVITIES AND AGILE SOFTWARE DEVELOPMENT

Karlheinz Kautz
Department of Informatics
Copenhagen Business School
DK-2000 Frederiksberg, Denmark

Abstract

This paper contributes to the studies of design activities in information systems development. It provides a case study of a large agile development project and focusses on how customers and users participated in agile development and design activities in practice. The investigated project utilized the agile method eXtreme Programming. Planning games, user stories and story cards, working software, and acceptance tests structured the customer and user involvement. We found genuine customer and user involvement in the design activities in the form of both direct and indirect participation in the agile development project. The involved customer representatives played informative, consultative, and participative roles in the project. This led to their functional empowerment—the users were enabled to carry out their work to their own satisfaction and in an effective, efficient, and economical manner.

Keywords

Participatory design, agile software development eXtreme programming, customer and user involvement, case study research

1 INTRODUCTION

A central tenet of participatory design (PD) is the direct involvement of people in the co-design of things and technologies they use. Although not that different from approaches already discussed in the 1980s under the label *prototyping* (Merisalo-Rantanen et al. 2005), agile (software) development (ASD) has recently gained immense popularity. In agile development—based on intensive direct communication and a few, short specification documents—interface, functional, and technical design, and the more technical development activities such as coding and testing are much more intertwined than in traditional information systems development (ISD) (Gross et al. 2008). In an agile manifesto, the advocates of agile development state their now well-known four pairs

of values, namely (1) individuals and interactions over processes and tools; (2) working software over comprehensive documentation; (3) customer collaboration over contract negotiation; and (4) responding to change over following a plan (see www.agilemanifesto.org).

Agile development practices and principles insist on the customer taking control and being constantly involved and stress a collaborative partnership based on daily interaction between the developers and the customer (Highsmith 2002). Studies of agile practice have, however, shown that customer representatives might have decision power, but only a limited understanding of the users' needs; they might not be the actual users of the software to be developed, who in turn might have the necessary knowledge, but not the authority to decide on system features (Robinson and Sharp 2005). Users rarely take the role of the customers (Martin et al. 2004). The customer role might even have been carried out by substitutes from the development organization such as product managers or marketing staff (Robinson and Sharp 2003).

It is surprising that agile software development and its relation to user involvement and participatory design has hardly been a topic in the ISD community in general, and in the PD community in particular. This is even more astonishing in light of the fact that Cockburn (2002), a prominent member of the agile movement and author of one agile method, has described how much his approach is indebted to Ehn's work, himself one of the prominent proponents of PD by explicitly relating to Ehn's (1992) article, "Scandinavian Design: On Participation and Skill," as a major source of inspiration.

Investigating user involvement and participation very much follows a request from Markus and Mao (2004), who argue for a need to revisit the concept and to study it in novel, exciting environments such as agile development. In addition, Bratteteig (2007) calls for studying design practice and emphasizes the need for practice studies with regard to user involvement and participatory design. This is in line with Dybå and Dingsøy (2008), who in a comprehensive study surveyed nearly 2,000 ASD related articles published prior to 2006 and who only found 33 scientifically sound, empirical studies of the phenomenon. They conclude that more such studies are needed.

On this background, in the following we will report from our research, which attempts to answer the question how customers and users participate in agile development and design activities in practice. In the next section, we will provide a literature review of related research publications and describe the theoretical background for the study. We then introduce our research method and the setting of our study, present our analysis, and discuss our findings; we finish with some conclusions.

2 RELATED WORK AND THEORETICAL BACKGROUND

Some research originating in the area of agile development, such as the previously mentioned work by Robinson and Sharp (2003, 2005) and by Martin et al. (2004), studies the task of the customer and the relationship of the customer and user roles, but less the actual participation of customers and users in the design and development activities of projects that use an agile development approach.

As we were interested in the relation between participatory design activities and agile software development, our literature search concentrated on studies that focused on

both areas. Our literature search of the participatory design literature, however, only led to a few contributions related to the question of user involvement in agile design and development activities.

In the only explicit PD paper we found on the topic Rittenbruch et al. (2002) provide a conceptual comparison of an agile development method, in their case eXtreme Programming (XP; for more information on XP, see section 4, “The Case Setting”), and PD. On this basis, they developed their own approach, which integrates techniques from both agile development and PD. They tested their method in a research project, where the customers were fellow IT-literate researchers, to provide a proof of their concept. Their research provides a number of practical recommendations to deal with the (from their perspective) limited conception of user participation in XP. However, their work does not provide any insights about user involvement in practice.

Hansson et al. (2006) present a practice study of how to include users in the development of off-the-shelf software and claim theirs is a case of complementing PD with agile development. However, the organization under investigation neither explicitly uses an agile method nor participatory design. The authors map the practices they find, namely the users’ feedback on the product through meetings, support sessions, and courses onto the two concepts. But they also admit that the users never actually participated in any design activities and that the developers totally controlled the process of selecting any of the proposals for design and development. Thus, while this is an interesting account of practice and a valuable description of user involvement in the further development of packaged software, the work does not investigate user involvement in agile practice.

In another study of possible interest in this context, Ferreira et al. (2007) investigated the relation between agile development and user interface and interaction design. They report about the cooperation of software developers and interaction designers from two cases in practice. Their descriptive account, based on a grounded theory inspired research method, surfaces a number of important themes such as up-front interaction design, continuous designer involvement, designer–developer interaction, and the valued role of interaction designers, but the work does not study user involvement at all.

Chamberlain et al. (2006) set out to develop a framework for integrating agile development and user-centered design. Analyzing the similarities and differences between the two approaches, they conclude that these are compatible. However, in a practice study, which investigated the use of agile methods and user-centered design, they identified a clear distinction between designers, who were responsible for user-centered activities, and developers, who produced code. They also found little, if any, customer and user involvement in the form of users being interviewed and asked for their opinion in limited tests sessions. On this basis they propose five principles for successful integration of the approaches including an explicit demand for user involvement. Unfortunately, the study does not provide a proof of concept based on empirical data to verify the viability of the framework.

There definitely seems to be a lack of empirical studies of participatory design activities in agile software development. We attempt to contribute to filling this gap and provide a case study, which we analyze based on the following theoretical backdrop.

To study participatory design, and within the design activities—more precisely, customer and user involvement—in agile software development, we will apply the concept of user focus. Iivari and Iivari (2006) distinguish between individual, average, and fictive user focus: With an individual focus the designer tries to satisfy each actual user and emphasizes each individual's capabilities and needs. With a focus on an average user, interaction designers typically apply some heuristics or general human factor principles. Focusing on a fictive user, HCI specialists or interaction designers base their design (proposals) on, for example, personas, which are descriptions of hypothetical archetypes of actual users.

Our further analysis will then be based on the following important concepts of user participation: (1) forms of involvement and participation, (2) user roles in user involvement and participation, and (3) the purpose of user involvement and participation as introduced by Mumford (1983), Damodaran (1996), and Clement (1994), respectively.

Clement argues that the purpose of user participation is user empowerment and he distinguishes between functional empowerment and democratic empowerment. The former means that the users should be able to carry out their work to their own satisfaction and in an effective, efficient, and economical manner. Their participation in the design process supports reaching this objective. Democratic empowerment means that they should have the mandate to participate in decision-making in their workplace including the design and development of software and IT-based systems.

Damodaran differentiates three user roles in the design and development process. The user can play an informative, consultative, or participative role. As informants, users merely provide information about their work and might be the objects of some observation. In a consultative role, they are asked to comment on preset design solutions. In a participative role, they actively participate in the design process and have decision-making power regarding the solution.

Mumford, who provided the first distinction of user roles, further classifies two different forms of involvement, namely direct user participation and indirect user participation, where the user is represented by some kind of intermediary. Direct and indirect participation are defined through the users' direct participation in the project (team) or their direct or indirect contact with project staff from the development organization.

3 RESEARCH METHOD

Our research follows the approach of engaged scholarship (Van de Ven 2007), which is a participative form of research for seeking advice and perspectives of key stakeholders to understand and theorize about a complicated problem. Given the limited literature concerning our research topic, understanding the participatory design activities, and customer and user involvement, our investigation is based on an exploratory, qualitative, single case study (Creswell 2003) of an ASD project. While it is often stated that it is not possible to generalize and certainly not to theorize from a single case study, Walsham

(1995) suggests that it is possible to generalize case study findings among others in the form of a contribution of rich insight. So inspired, we have used our theoretical backdrop, which consists of important concepts that make up user participation, as one background for our data analysis.

The research presented is part of a larger project that aims at understanding agile development in practice and at contributing to sustainable theories of ISD in general and agile development in particular (Kautz and Zumpe 2008; Matook and Kautz 2008). In the context of this paper, we focus on the role of the customers and users and how they are involved in the design and development process. Our research is based on an empirical case study of a commercial agile development project in a large German public sector organization, called WaterWorks, performed by a German software company, called AgDev, which specializes in agile development.

The empirical data for the case study was collected in semi-structured, open-ended interviews that were conducted by a team of two researchers in a three-day period at the development site. The research team performed 12 interviews with 11 individuals (the AgDev project manager was interviewed twice). This included nearly a third of the development team and a representative sample of key players and future users in the customer organization. The interviews were tape-recorded and subsequently transcribed. For the qualitative data analysis, a software tool (NVIVO7) was used. The interview data was supplemented with company and project documents such as method, requirements, and release descriptions, as well as project plans.

The data collection, the coding of the data, and the data analysis have been guided by the four pairs of values underlying agile development previously introduced in this paper. As stated earlier, for the research presented here, the data was in particular analyzed with regard to participatory design and customer and user involvement. As a prerequisite for the analysis, we produced an account of our case setting, which focuses on the structural profile of the investigated project. It consists of descriptions of the information system under development, of the formalized method (Fitzgerald et al. 2002) to be used—as opposed to the method-in-use—of the project's structural context, and of the structural characteristics of the involved development team, its members, and associated staff such as customers and end users. This is presented in the next section.

4 THE CASE SETTING: THE OMS PROJECT

The project under investigation was concerned with the development of an operations management system (OMS) for the WaterWorks of a large German city. Founded 150 years ago, the organization is now partially privatized with the city council holding 50.1 percent of the ownership. The system was developed with a web-based graphical user interface and a backend to interface the technical infrastructure as defined by an underlying ERP system.

The project was organized in four subprojects to provide IT support ranging from customer management to the maintenance of the sewer and duct system. After several attempts of traditional ISD based on a standard ERP system that had not led to the desired results, the organization opened a tendering process. It was won by a small software company, AgDev.

At the time of the project, AgDev consisted of about 25 employees 20 of them being developers, and based its development approach on the agile method XP (Beck and Andres 2004). The formalized method includes planning techniques for releases and iterations called planning games, user stories and story cards to specify user requirements (in XP formally the customer writes the stories onto simple index cards), onsite customers to support customer–developer communication, daily stand-up meetings of the whole project team to support team communication, pair programming, re-factoring, collective ownership, continuous integration and testing of code to develop the software proper, and regular tuning workshops to improve the development processes. The company extended the method with some project management processes to cater for larger projects such as an elaborate overall project plan, formal reporting mechanisms, and a formal contract based on a requirements specification called a realization concept, which had been produced by the customer.

The project was organized in two phases. In a first 12-month exploration phase, prototype catching requirements and possible solutions were developed. This led to the development of the realization concept by the customer organization and their decision to contract AgDev also for the development of the OMS proper.

In this main development phase, a team of about 12 development staff with multiple roles such as project manager, subproject manager, analyst, customer contact, and developer worked onsite in a building owned by WaterWorks. A sophisticated management structure with one subproject manager acting as contact person from AgDev and one subproject manager acting as onsite customer from WaterWorks for each of the four subprojects was established, in addition to two overall project managers, one from each company. The WaterWorks subproject managers and onsite customers were managers and team leaders in the operational divisions of the application areas and as such also actual users of the OMS. These customers were, by and large, not onsite the entire time. The project also comprised a varying number of other users, representing operational staff, from the different divisions. These users were, for the most part, actively involved in feedback and testing activities, as will be described and discussed in more detail in the next section.

When this study was performed, phase one had been successfully closed and, after a break of over a year due to internal politics at WaterWorks, phase two had been going on for four months. Responding to an inquiry call during our analysis, the AgDev project manager stated that the project ended 10 months later, on time and budget, with all parts of the OMS being operational. The collected data is mostly related to phase two.

With this structural profile in mind, we will now analyze the project in more detail with regard to participatory design and user involvement.

5 AN ANALYSIS OF THE OMS PROJECT

The OMS project was described by both the customer and the supplier as a success. The emerging information system afforded, in the words of one of the WaterWorks subproject managers, the ability to identify synergies among the various departments; in particular, in the duct department it enabled improved planning that resulted in the possibility to dispose cleaning vehicles and reduce related staff. With regard to the focus

of this paper, one of the WaterWorks subproject managers explicated that their end users had been very satisfied.

However, various comments were made about reaching the right balance of customer collaboration and user involvement. AgDev's project manager commented on onsite customer behavior: *"Yesterday he said something and today he says something else. Requirements have to be clear at the beginning of an iteration and cannot change right in the middle of it. We are agile, but not on a daily or hourly change request rate."*

Customer and user involvement took place on an ongoing basis and the planning games and story cards, the presentation of working software, as well as the acceptance tests structured the continuous day-to-day-contacts, communication, and collaboration. In addition, the project showed one very particular instance of user involvement. One AgDev subproject manager reported on one user, who—based on his background and experience—had become a full member of the development team.

In the following we will use the planning games and story cards, the presentation of the working software, and the acceptance tests to arrange our description and analysis of the user involvement in the project's development and design activities.

5.1 Planning Games, User Stories, and Story Cards

At the start of phase two, a number of different documents existed, which were all comparable short and concise. From a customer perspective, these were related as follows: An overall realization concept built the basis for the development contract with the customer. The realization concept was refined into requirements lists. These lists governed what should be the outcome of an iteration and what should be the basis for the acceptance tests. Individual requirements or groups of requirements were then described as a story and each story was written down on a story card. The story cards represented the final detailed plans and specifications for the developers' work tasks and processes.

The planning games at the beginning of each iteration were based on the overall realization concept and the requirements lists. These were largely produced by AgDev's project manager and some of the subproject managers. They developed these documents with input from the onsite customers. The story cards were solemnly produced and estimated by the developers in developer team work sessions, because the subproject managers as well as other possible end users at WorkWorks were mostly operational staff, which had some impact on their abstraction capabilities and their ability to write texts. Yet, the developers and the customers together prioritized the cards.

In this context, WaterWorks' subproject managers saw their role as facilitators and communicators. To back up the development of the operations management system based on an agreement with the staff council and with management, some of them had been assigned full-time to the project to be available and involved in the project whenever necessary.

The subproject managers did not develop the requirements at their own discretion, but had significant contact with the employees in their divisions and carried the requirements from their divisions into the project. They also prepared the prioritization of the requirements according to their importance and the available budget.

An AgDev subproject manager talked about the difficulties of converting the requirements into design and declared that design was the task of the developers: *“It nearly becomes our design task as contact partners...it’s not easy to find out from the WaterWorks people what they want; when I say ‘do you want it this way,’ they say ‘yes,’ and when I ask ‘do you rather want it that way,’ they also say ‘yes’... and they say ‘we have this and this problem, but to design an interface out of this information is our problem.”*

However, the design was always developed with close participation of the WaterWorks subproject managers and other users and always under the mandate of the WaterWorks subproject managers. While the AgDev subproject managers had the liberty to make proposals, the customers could always say no, and with really important issues, AgDev would always get back to the users before they would go forward. This also included direct contact between those developers, who did not act as contact persons or subproject managers, and the onsite customers.

Some AgDev subproject managers even felt that, through the WaterWorks involvement, the project participants from the WaterWorks somehow developed the OMS themselves. One of them thus explained that AgDev had not developed something, which they had invented themselves without communication with WaterWorks. This form of customer collaboration provided some structure to cope with the complexities of a comparatively large ASD project, while leaving room for less structured, but necessary, collaboration as well.

That is to say, when implementing the story cards, it became obvious that some additional collaboration was needed. One WaterWorks subproject manager stated that contact with the onsite customer was necessary for nearly every story card. He put forward that maybe two thirds of a card’s contents were clear, but that the other 40 percent had to be directly coordinated—20 percent in the middle of an iteration and 20 percent in the end when the customers and users looked at the working software and found that their requirements weren’t understood the way they meant them. This illustrates the importance of the working software for the development and design process, which we will discuss in more detail in the next subsection.

5.2 Working Software

The presentations of working software were identified as another basis for customer and user involvement. In the OMS project, a first software release was provided after three months with the others to be delivered every three to six months. Each release was organized in iterations of three to six weeks duration, meaning that at the time of our investigation each subproject had at least gone through two iterations.

Feedback about and change requests for the software design were brought forward by the onsite customers in weekly feedback loops, which were built into an iteration, based on presentations and demonstrations of working software. This always led to smaller changes.

The AgDev project manager described how the working software, which was produced by story card, attracted the WaterWorks subproject managers and how they

seamlessly participated in the development process. He confirmed that beyond the scheduled weekly meetings for all subproject managers, some of the WaterWorks subproject managers turned up at the project nearly on a daily basis while others came at least on a regular basis and looked over the shoulders of the developers. This allowed for fixing problems and providing working software before an iteration was officially released. This was very important for the WaterWorks subproject managers, as they continuously could see progress. The developers, however, had to get acquainted with this close involvement as they were not that used to sitting at the customer site and having customers pass by every day, which they sometimes considered as quite disturbing.

Beyond these contacts with the onsite customers, the working software was also presented to larger groups of prospective users (presentations to one onsite customer were not considered sufficient). Thus the set up, with at least four subproject managers, who also acted as onsite customers, was supplemented with other user representatives. The AgDev project manager summarized the situation: *“Well, at the latest when an iteration is finished, sometimes already in the middle of it, or whenever, presentations are run for users. Well, not always in front of many users, but the customer subproject manager gets some people together and says: ‘Here, look, do we develop in the right direction?’”*

In addition, using a similar format, the onsite customer representatives regularly performed “road shows” with the working software in the user departments to collect feedback and ideas and proposals for improvements.

The AgDev subproject managers also sought direct contact with the users and one of them reported that he had seated himself for two weeks in the duct operation station with the objective to extensively test onsite and to look at how well the software actually fitted the operation. This resulted in the direct involvement of those employees onsite, who arranged or actually performed cleaning the ducts. Another AgDev subproject manager had chosen the same strategy and even involved some of his developers in the process. After installing a release at one division, they went there for several days, discussed the release with the users, registered bugs, and then rebuilt the software accordingly.

The frequent feedback loops were taken very serious and immediately responded to with action. This had the effect that minor misunderstandings were caught and dealt with as changes early before they could grow into larger problems. As a consequence, the users developed trust and a feeling that they had an impact on the development of the system and even the employees in the divisions, who were not directly involved, were quickly integrated into the project.

But the working software also brought to light some initial problems related to the distribution of roles in the project. The WaterWorks subproject managers expected that AgDev’s developers would bring more of their own ideas into the project and that they would come up with smart technical solutions. They were frustrated that the AgDev developers always just did what the customers told them to do, because they saw this as a sign that the developers were not good enough to develop their own proposals. After a clarification of the roles in an agile development project, the cooperation between the different groups of project participants continued without further problems.

5.2 Acceptance Tests

The AgDev project manager also explained that, between two iterations, there was always a test phase that was a “post” activity of the preceding iteration and that there was a concept phase that was the preceding activity of the next phase. He confirmed that the acceptance tests were run by the customer, meaning that the customer had the responsibility and decision power for and in these tests. The tests were similar to the formal presentations of the working software, but they were performed according to a protocol and they always comprised end users. Thus, customer and user involvement also took place during and in the form of the acceptance tests. The requests for changing the software design that came up during the scheduled acceptance test sessions were dealt with in the next iteration.

Before an acceptance test was performed, less formal preparations took place, often triggered by a WaterWorks subproject manager, who would try out the new functionalities as described on the story cards. As valid feedback was considered important, end users—not just the customer as represented by the subproject leaders—were performing the tests. An acceptance test was then always led by a responsible WaterWorks subproject leader, who also approved or rejected the new version of the system. A typical acceptance test lasted just one day with four to six people participating. Two or three divisional managers and other employees, who owned the task and who had to work with it, were present and tested. There was an official test protocol where the requirements for the iteration were listed. The responsible WaterWorks subproject manager’s approval, conditional approval, or rejection as a test leader was documented, together with what was missing, and where, to achieve a full approval.

The AgDev subproject managers were always present during the tests, which otherwise were the responsibility of WaterWorks. The AgDev subproject managers also monitored the acceptance test and recorded the errors and bugs on “private” bug lists. The developers usually did not participate. There were, however, exceptions from this rule as we learned from an AgDev subproject manager: *“In the end, we participated in the test, me, our project manager, and the respective WaterWorks subproject manager, and in this case another developer as well.”*

One AgDev subproject manager described that this way many mistakes were found and that encouraging feedback was provided. The employees involved in the tests stated that they could imagine working with the system and that they liked it better than the ERP-based solution that had been proposed earlier.

The WaterWorks subproject managers declared that this approach, with the acceptance tests before an iteration was approved and the iterations as such, helped them to quickly and regularly get something tangible and to get in touch with the end users. They confirmed that in this context the test sessions, which were also coordinated with the staff council, were decisive. By and large, WaterWorks’ subproject managers were content with this form of participation and considered themselves as part of the development. They also reported on content end users and based this appraisal on the feedback they had received from the end users after the tests. The acceptance tests had a significant influence on the further design of the system components, as requests for changes and new requirements always came up as a result of a test.

However, despite the general positive reception of the acceptance tests, still more involvement and more tests were also demanded, as can be seen by the following final statement made by one WaterWorks subproject manager: *“The tests were good, but there were too few.”*

6 DISCUSSION

Prior research into user participation in agile development has shown that end users rarely take the role of the customers, and sometimes the customer is even represented by a development organization’s own staff (Martin et al. 2004; Robinson and Sharp 2003, 2005). In contrast, in our case we found genuine customer and user participation in an agile development project.

We can to some degree empirically confirm Chamberlain et al.’s (2006) conceptual work that agile development and user-centered design are compatible; user participation and agile development go well together. But user participation in our case was an integral part of the chosen methodology and thus no special attempt to integrate the two approaches, as Chamberlain et al. demand it, was necessary. There were not two approaches, nor was there a distinction between designers and developers, which others (Chamberlain et al. 2006; Robinson and Sharp 2003, 2005) have found in practice.

This might have been the case in the OMS project, because of its user focus. In the OMS project the focus was not on any average user nor was it on a fictive user. It was on the actual individual customers and users. Our empirical data show that they—even if they were not always content with the fact that they had to take that role—through their continuous feedback acted as designers themselves.

Our case also illustrates how, in agile development, design and development are intertwined. This confirms earlier work on agile development (see Gross et al. 2008), but has also been known for other ISD and software development approaches for many years (see Swartout and Balzer 1982).

Returning to participatory design, customer and user involvement and participation, we base the further discussion of our findings on the concepts introduced in our theoretical background.

In terms of Mumford (1983), we found, in the OMS project, both direct and indirect participation: the onsite customers were WaterWorks staff, who themselves would work with the future system to a certain extent. As such, they exerted direct participation. They were also intermediaries for other users, for example, operative staff in the customer division and the duct net division. These employees indirectly participated when they commented during presentations or when they provided their viewpoints or descriptions of their work processes to the onsite customers. However, they also participated directly when they tested the results of the iterations or when they were observed or conferred the developers, who were in contact with them directly in the divisions, such as the developer, who visited and stayed in the duct net division while the result of an iteration was in operation.

Although one of the WaterWorks subproject managers went so far as to report to management that only things that the employees wanted were done, a more differentiated

picture appears when we analyze the various user roles (Damodaran 1996). Those users in the divisions, who were not also onsite customers, had the role of informative or consultative users. They provided both the onsite customers and the development staff at AgDev with information about their work, their needs, and their preferences. Some of them were observed during their work by some developers and they provided their comments during presentations and tests. Although this information and the comments were taken into account and much of it was realized, it goes too far to consider this a participative role as it was primarily the onsite customers, who had a decision-making mandate. Thus, in so far as their positions as (a minority among the) future users were concerned, they were clearly in a participative role, although they did not themselves write the user stories and the story cards. However, no design decision could be taken without their agreement. Beyond their participative role, the onsite customers also had informative and consultative roles.

Finally, with regard to empowerment (Clement 1994), we can determine that functional empowerment has been achieved. All stakeholder groups reported that they were content with the outcome of the project. We have, however, no evidence for any democratic empowerment other than the staff council's peripheral involvement and the onsite customers' authority to take decisions concerning their staff's and their own workplaces. This leads us to a concluding statement.

The PD movement has traditionally advocated workplace democracy, a participative role for the users, and direct involvement (Bjerknes and Bratteteig 1995), while in large parts of the mainstream IS literature, functional empowerment with users in informative or consultative roles, directly or indirectly involved, has been the focus of research (Kujala 2003). Given the results for our analysis, it could be argued that our research does not really show a case of participatory design in agile development. It can, however, be counter-argued whether this is relevant given that all involved stakeholders in the case were satisfied with the outcome of the OMS project.

Furthermore, in 1995, Bjerknes and Bratteteig had already shown how the focus on workplace democracy had shifted to a cooperative design approach, at least in the Scandinavian PD community. On this basis, they then put forward the challenge for future PD research to contribute to democracy in a changing environment for working lives and workplaces. They claimed that other kinds of actions and institutions, different from traditional PD approaches, might be necessary to reintroduce the democratic dimension in PD and ISD research. It may thus be that agile development, with its local focus on constant cooperation, reviews, and feedback, and with direct communication as its preferred development method, is, after all, one possible way to tackle this challenge.

7 CONCLUSION

In this paper, we investigated the question of how customers and users participate in agile development and design activities in practice.

Our work demonstrates the applicability of concepts developed to understand participatory design to analyze customer and user involvement in ISD projects that apply an agile approach. As such, it follows Markus and Mao's (2004) call and revisits the concepts and shows that they are still useful.

We also contribute with a practice study of a design process as requested by Bratteteig (2007) to broaden the perspective on design science research, and we provide a sound, empirical study of agile development as demanded by Dybå and Dingsøyr (2008).

Our research contributes to the body of knowledge in information systems development with rich insight (Walsham 1995) and provides a link between the otherwise often-disconnected research areas and research communities of participatory design, agile development, and design science. Further research is necessary to allow for more theorizing about the relation of the three fields.

Acknowledgments

I would like to thank the participants of the OMS project for their time and insights in agile development practice. I am also indebted to Ralf Klischewski, who was a member of the original interview team, and Sabine Matook (né Zumpé), who performed parts of the data analysis with me, as well as to Helen Sharp, who directed our attention to the body of knowledge and provided literature, which documents empirical studies of user involvement in agile development.

References

- Beck, K., and Andres, C. 2004. *Extreme Programming Explained: Embrace Change* (2nd ed.), Boston: Addison-Wesley Professional.
- Bjerknes, G., and Bratteteig, T. 1995. "User Participation and Democracy: A Discussion of Scandinavian Research on System Development," *Scandinavian Journal of Information Systems* (7:1), pp. 73-98.
- Bratteteig, T. 2007. "Design Research in Informatics: A Response to Iivari," *Scandinavian Journal of Information Systems* (19:2), pp. 65-73.
- Clement, A. 1994. "Computing at Work: Empowering Action By 'Low-level Users,'" *Communications of the ACM* (37:1), pp. 52-63.
- Chamberlain, S., Sharp, H., and Maiden, N. A. M. 2006. "Towards a Framework for Integrating Agile Development and User-Centred Design," *XP 2006*, pp. 143-153.
- Cockburn, A. 2002. *Agile Software Development*, Boston: Addison-Wesley.
- Creswell, J. W. 2003. *Research Design – Qualitative, Quantitative and Mixed Methods Approaches*, Thousand Oaks, CA: Sage Publications.
- Damodaran, L. 1996. "User Involvement in the Systems Design Process: A Practical Guide for Users," *Behaviour and Information Technology* (15:6), pp. 363-377.
- Dybå, T., and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and Software Technology* (50:9/10), pp. 833-859.
- Ehn, P. 1992. "Scandinavian Design: On Participation and Skill," in *Usability: Turning Technologies into Tools*, P. S. Adler and T. A. Winograd (eds.), New York: Oxford University Press, pp. 96-132.
- Ferreira, J., Noble, J., and Biddle, R. 2007. "Interaction Designers on eXtreme Programming Teams: Two Case Studies from the Real World," in *Proceedings of the Fifth New Zealand Computer Science Research Student Conference*, Hamilton, New Zealand.
- Fitzgerald, B., Russo, N. L., and Stolterman, E. 2002. *Information Systems Development, Methods in Action*, London: McGraw Hill.

- Gross, J. B., Daughtry III, J. M., and Lee, J. C. 2008. "Heuristics of the World Unite! Merging Agile Methods in Software and Interaction Design," *Agile Journal* (3:2).
- Hansson, C., Dittrich, Y., and Randell, D. 2006. "How to Include Users in the Development of Off-the-Shelf Software: A Case for Complementing Participatory Design with Agile Development," in *Proceedings of the 39th Hawaiian International Conference on System Science*, Los Alamitos, CA: IEEE Computer Society Press.
- Highsmith, J. 2002. *Agile Software Development Ecosystems*, Boston: Addison-Wesley.
- Iivari, J., and Iivari, N. 2006. "Varieties of User-Centeredness," in *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, Los Alamitos, CA: IEEE Computer Society Press.
- Kautz, K., and Zump, S. 2008. "Just Enough Structure at the Edge of Chaos: Agile Information Systems Development in Practice," in *Agile Processes in Software Engineering and Extreme Programming—Proceedings of the International Conference XP 2008*, P. Abrahamsson, R. Baskerville, K. Conboy, B. Fitzgerald, L. Morgan, and X. Wang (eds.), Berlin: Springer, pp. 137-146.
- Kujala, S. 2003. "User Involvement: A Review of the Benefits and Challenges," *Behaviour and Information Technology* (22:1), pp. 1-16.
- Markus, M., and Mao, Y. 2004. "User Participation in Development and Implementation: Updating an Old Tired Concept for Today's IS Contexts," *Journal of the Association for Information Systems* (5:11-12), pp. 514-544.
- Martin, A., Biddle, R., and Noble, J. 2004. "The XP Customer Role in Practice: Three Studies," in *Proceedings of the Second Agile Development Conference* (June 22-26), Washington, DC: IEEE Computer Society, pp. 42-54.
- Matook, S., and Kautz, K. 2008. "Mindfulness and Agile Software Development," in *Proceedings of the 19th Australasian Conference on Information Systems (ACIS): Creating the Future: Transforming Research into Practice*, P. Cragg and A. Mills (eds.), Christchurch, New Zealand, pp. 638-647.
- Merisalo-Rantanen, H., Tuunanen, T., and Rossi, M. 2005. "Is Extreme Programming Just Old Wine in New Bottles: A Comparison of Two Cases," *Journal of Database Management* (16:4), pp. 41-61.
- Mumford, E. 1983. *Designing Human Systems for New Technology: The ETHICS Method*, Manchester, UK: Manchester Business School.
- Rittenbruch, M., McEwan, G., Ward, N., Mansfield, T., and Bertenstein, D. 2002. "Extreme Participation: Moving Extreme Programming Towards Participatory Design," in *Proceedings of the Participatory Design Conference (PDC) 2002*, Malmo, Sweden, June 23-25, pp. 29-41.
- Robinson, H., and Sharp, H. 2003. "XP Culture: Why the Twelve Practices Both Are and Are Not the Most Significant Thing," in *Proceedings of the Conference on Agile Development* (June 25-28), Washington DC: IEEE Computer Society.
- Robinson, R., and Sharp, H. 2005. "The Social Side of Technical Practices," in *Extreme Programming and Agile Processes in Software Engineering, Proceedings of 6th International Conference, XP 2005*, H. Baumeister, M. Marchesi, and M. Holcombe (eds.), Berlin: Springer, pp. 100-108.
- Swartout, W. R., and Balzer, R. 1982. "On the Inevitable Intertwining of Specification and Implementation," *Communications of the ACM* (25:7), pp. 438-440.
- Van de Ven, A. H. 2007. *Engaged Scholarship: A Guide for Organizational and Social Research*, New York: Oxford University Press.
- Walsham, G. 1995. "Interpretive Case Studies in IS Research: Nature and Method," *European Journal of Information System* (4), pp. 74-81.

About the Author

Karlheinz Kautz is a professor in Systems Development and Software Engineering in the Department of Informatics at the Copenhagen Business School, Denmark. His primary research interests are in information systems development, knowledge management, software quality and process improvement in the IT industry, and the diffusion and adoption of information technology innovations. Karl is a founding member and former chair of IFIP TC8 WG 86 on Diffusion, Transfer, and Implementation of Information Technology. Karl's e-mail address is Karl.Kautz@cbs.dk.

