

Community Driven Development as participation? – Involving User Communities in a Software Design Process

Jan Hess, Sinja Offenberg

University of Siegen (IS and New Media)
Hoelderlinstr. 3, D-57068 Siegen
+0049 271 740 2098
(jan.hess;sinja.offenberg)@uni-siegen.de

Volkmar Pipek

University of Siegen (CSCW in organisations)
Hoelderlinstr. 3, D-57068 Siegen
+0049 271 740 4068
volkmar.pipek@uni-siegen.de

ABSTRACT

In this paper we report on a case study of a participatory design process we call *Community Driven Development* (CDD). Together with a German software company we developed a socio-technical environment to motivate users of an existing online user community to participate in the further development of a product. For the CDD approach developers and users worked together to create a functional description and different prototypes of a target product. By conducting interviews and analyzing formal and informal processes we could identify different motivations, fears, problems and benefits the participants perceived during the process. Our study shows that an existing online community can be a valuable basis for a PD process, but it calls for a careful design of the organizational and technological setting and a sensible process moderation. It also shows that existing professionalization structures in technology design arenas may always lead to an imbalanced 'participation'.

Keywords

Virtual Community, End User Development, Home Entertainment Software, User-Driven Innovation

INTRODUCTION AND MOTIVATION

It has always been far from obvious what exactly 'participation' is when it comes to information technology design. In the early days of 'personal computing' in the 1980's, the conflict lines at the workplace (employer's interest in efficiency/rationalization vs. employee's interest in good working conditions/ergonomics) provided some orientation on different levels of participation [3], and how certain process types or user-developer interaction arenas ('PD methods') influence them.

Today, arenas of IT design look different. IT has conquered more and more areas of our everyday lives and it is hidden in more and more devices and technological infrastructures. General computer literacy has increased among IT users, and the WWW as well as the Open Source movement offer new ways of use- and development-related articulation (support forums, user wish lists, etc.). These articulations

may also have become more competent regarding the potentials and limitations of IT. New technologies, products or usages encounter an existing base of technologies and usages they have to blend into, and often face competing socio-technical arrangements. IT development strategies adapted to these market dynamics by becoming 'user-oriented', maybe even 'user-centered', but not necessarily 'participatory'. The user got a voice in these design processes, but it is not clear who listens to it.

Taking these developments into account, PD researchers are faced with new challenges and potentials. The two stereotypes of the 'user-unaware developer' and the 'computer-illiterate user' are replaced by more gradual mixtures of competencies. When suggesting PD arenas, we have to consider these various mixtures as well as the ongoing learning processes that accompanies a design interaction. As home/leisure settings complement traditional work settings as domains for PD interaction, the different degrees of motivation, involvement and dedication to the PD interaction have to be considered. Design-time and use-time cannot be separated anymore, since IT artifacts have become more flexible and adaptable to different uses, and because they mutually influence each other's use and indirectly (e.g. by debating feasible technology potentials) each other's development. As a result, it may always be 'design-time' for dissatisfied users, or users choose a different socio-technical arrangement (i.e. a different product). As we suggest in [20], PD research may react to this development by focusing on developing infrastructures-in-use further rather than on developing IT artifacts.

The framing conditions for technology development offers new potentials for PD research, as well. The IT infrastructures we have today provide more ways to articulate and exchange needs, ideas and opinions, and offer participation opportunities beyond traditional views on technology design (e.g. with regard to political issues like standardization). Practical experiences and the competition with the Open Source movement may encourage more and more professional IT developers to take the step from 'user-centered' to 'participatory' design, giving PD research more practical relevance and resulting in more opportunities for practice-oriented research.

In this paper, we explore a PD interaction in practice that shows many aspects of the mentioned developments. A software manufacturer (Omega) for Home Entertainment

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee if copies are not made or distributed for profit or commercial advantage. Copies must bear this notice and full citation on the first page. To copy otherwise in any way requires prior permission and must be requested in writing to Indiana Univ. Conferences. Proceedings Participatory Design Conference, CPSR/ACM Copyright © 2008 Trustees of Indiana University ISBN 978-0-9818561-0-0

Software wanted to develop a new Media Centre software with the help of an existing online community. We observed and supported the practice of developer-user relations and the initiative towards redesigning the product for about 18 months. Focusing on *Community Driven Software Development*, we conceptualized and gradually improved a PD arena in order to explore the dynamics of the perceived and actual values of participation as well as the associated expectations and fears of the participating stakeholders. Our experiences can thus inform other PD approaches that operate with virtual user communities.

We will now briefly address related work that focuses on in-use PD concepts and community-driven concepts. After that, we will describe our case setting and concept in more detail. We will identify significant effects we were able to observe and discuss the course the PD interaction took, relating our findings to other PD approaches to delineate different understandings and practices of participation.

PARTICIPATION IN USE

Several studies on PD research already account for different modes and levels of participation ([3], [17]), but they merely reflect the historical context of participation in workplace design. One can find the emancipatory direction (users should be an active part in workplace design) as well as the production-oriented description (users have to be integrated in existing design practices, e.g. by using ethnographic methods [18]), but in most approaches the design process precedes the actual use of the product. Traditional techniques include interviews, surveys, questionnaires or protocol analysis. In the 1990's alternative techniques like observation and video analysis emerged. Such 'contextual' techniques help to better understand users' working environments. A company can employ different additional techniques which can encourage users and developers to exchange information. Keil and Carmel call them customer-developer links [14], which may include a support hotline, a bulletin board or a trade show. By comparing different projects they found out that more successful projects employed more customer-developer links than less successful ones.

Muller et al. [17] classified participative techniques into two dimensions. One dimension is the level of user involvement; a user can either be observed or he can actively participate in discussions. The second dimension is the user's participation in the development process. Fischer and Scharff [6] distinguish between 'design-time' and 'use-time'. They point out that problems in the subsequent usage cannot be completely anticipated while designing a product. Users will discover mismatches when they actually use the product. One of the main goals of End User Development (EUD) research is to bridge the gap between design- and use-time. Participation in the sense of EUD will 'empower end-users to develop and adapt system themselves' [16]. These adaptations can only be realized with high-flexible software architectures ([24]). Because traditional PD approaches are focused on the development

at design time, Fischer argues for a meta-design that involves users over a long period of time. *'Meta-design extends the traditional notion of system design beyond the original development of a system to include an ongoing process in which stakeholders become co-designers – not only at design time, but throughout the whole existence of the system.'* Visser [22] stated that the same users should participate not only at a single stage of the design process, but also later. Such 'returning participants' would give a more effective feedback, because they already have a relatively deep knowledge of the application's concepts.

As a result of globalization and the spread of new technological facilities, development processes can be managed in more distributed settings [10]. Such projects differ from traditional processes of development and require a rethinking by different stakeholders. On the one hand, the process of implementation can be 'distributed'. On the other hand, user involvement can be stimulated by the use of Internet tools. Farshchian [4] reported of a case study in which users participated in an international software development project via email and the Internet. Because informal communication mainly took place asynchronously by the use of mailing lists, prototypes were the main formal reference for stimulating discussions and improvements, which may lead to new forms of technological process support. Another challenge is the design of organizational aspects, in order for user involvement to work. The concept of a continuous user involvement can often be found in open source software communities [2]. Users can utter wishes for improvements (e.g. in special mailing lists) or can reprogram the software themselves. Different roles also exist in open source projects (including project leader, core team, developers, active and passive users), and the process of deciding which of the users' suggestions can be considered and which can not is not completely clear in all cases. Our CDD approach tries to provide some means for support by the initiation of a control instance, which consists of representatives of different stakeholders, both users and professionals.

VIRTUAL-COMMUNITIES AND PARTICIPATION

As members of an online community meet only virtually, spatial limitations lose some of their importance. The motivation for being part of such a community is very often a shared interest. Howard Rheingold characterizes virtual communities as *'social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace'* [21]. However, everyone has his own interpretation what connectivity in such social aggregations means: *'We all have our own notion of what an online community is. It isn't hard to understand, but it is slippery to define and tricky to measure'* (Lazar and Preece in [15]).

Communities can be classified according to different aspects. One classification is given by Armstrong and Hagel [1]. They distinguish between four different types of com-

munities, including transaction oriented, interest oriented, fantasy oriented and relationship oriented communities. Barry Wellman [23] proposes different categories of virtual communities, one of these is called *Communities of Consumers*. Such communities differ from other communities because they are controlled by an organization. Users who are engaged in such communities often bring in many innovative ideas for product improvements [9]. Another advantage of communities of consumers is the fact that users can contact and help each other. A potential customer will probably trust a user's recommendation more than a company's advertising. Companies are aware of that and also see positive commitments as an effective form of marketing [15]. With the idea of Use Discourse Environments as platforms for 'built-in' communities for technological artifacts we fostered user-user-collaboration to support appropriation work ([19]).

An important theory for involving users in the design process is given with the Lead User Theory [8]. Based on several studies, Hippel [8, 12] defines Lead Users as users who fulfill two characteristics. On the one hand they are intensively engaged with the particular product and the associated market. This is how they can discover new trends and demands in an early stage. On the other hand, Lead Users should anticipate advantages that lie in new technologies for themselves. These characteristics will lead to a high engagement for participation. The motivation for taking part in such design processes was explored by Füller et al. [9]. Users hope to help creating an improved product that meets their personal needs better. Further factors are fun, curiosity, desire to learn, personal interest, acceptance from others and the access to exclusive information. Users feel more accepted and build up a deeper relationship to the producer. Hippel [12] splits the process of Lead User involvement into four phases. In the first stage a new trend is identified. Based on the users' requirements and experiences some Lead Users are chosen. Afterwards, the Lead Users' demands are analyzed, which results in new product concepts. Finally, these new concepts are projected on a larger market.

The involvement of users in the design phase is not trivial. Users as well as employees have to be prepared for such a process. On the developer side, programmers often resist to contributions from external stakeholders. One solution is the involvement of so-called 'gatekeepers' [11]. 'Gatekeepers' are persons who enjoy the users' as well as the employees' confidence. They connect a company with external sources by filtering relevant information in a structured way. Such persons often exist in open source software projects. Barcellini [2] calls them 'cross-participants' because they participate in parallel discussion spaces and therefore may have the best overview of ideas and improvements.

Füller et al. describe a concept, which enables the involvement of members of virtual communities in a structured way [9]. The approach they developed is called *Community*

Based Innovation (CBI). CBI can be applied in four phases. In the first phase attributes of the users are identified which fit the task requirements at its best. As a second step a community is identified where the key users can be found. In the third step, a virtual interaction design has to be developed to support communication with the users. The last step implies the real involvement of the users, starting with establishing contacts and resulting in design participation. This way, users can participate already in early design phases: '*Members of online communities who are characterized by high product and activity involvement, represent an ideal resource for co-designing products when confronted with those new methods*' [9]. Füller et al. have proven the concept in cooperation with the Audi AG. In the three-month lasting case study users were involved in the planning of a new infotainment system. Users were able to participate in the design process by using a so-called *virtual lab*. The lab consisted of a product configuration tool and some text entry fields for own ideas. One of the major findings of the study was the circumstance that users are capable and willing to participate on such a process.

COMMUNITY DRIVEN DEVELOPMENT

In the previous chapters we described concepts where users were somehow involved in the design and innovation process. However, none of the known studies treats users and employees with equal importance. Users can utter wishes and take part in the development process, but they do not have any influence on the decisions that are finally made. Furthermore, earlier research was mainly interested in the developers' point of view, while users' opinions were not explored to such an extent. Even if our approach has many things in common with the concept of Füller et al. [9] we wanted to focus on the participation of users in the whole development process, not only within a single process stage. In order to support a development that is really driven by users, we designed the concept of *Community Driven Development*.

Concept

In order to include a user community in a software design process, room for discussions is needed. Virtual platforms like forums, where all interested users can share and discuss their ideas and opinions, provide an alternative to physical meeting places. In the CDD approach, the group of users involved in design is called the 'User Parliament' of the community. If too many people take part in the discussions, developers can limit the number of members of the User Parliament and establish an application procedure. The concept's second institution is the 'Central Committee', which consists of elected users and developers who collect information and make the final decisions. The members of the Central Committee play a very important role in the process; they should consequently enjoy the full confidence of both users and developers. In accordance with the gatekeeper principle it is therefore reasonable that User Parliament and developers elect the members of the Central Committee together. The members of the Committee have the possibility to take part in the discussions of the

User Parliament as private users, but in their function as Committee members they should be neutral and act as moderators if necessary. The Central Committee should mainly be constituted of users to guarantee a consumer-friendly development. The Committee hosts regular conferences, either in person or by telephone, to discuss users' ideas and interests about previously defined topics in order to come to mutually acceptable decisions. The results of the User Parliament's meetings are summarized in a public space to create a central requirement specification that forms the basis for the software development process. An initial prototype should be built and given to all interested users as soon as possible, so that they can constantly test and improve it. This design cycles should the STEPS model. In a STEPS process developers and users work closely with each other to improve early versions [7]. Central aspects of the CDD concept are the continuous involvement of the users, the development and improvement of a prototype and the users' power to make decisions, which is secured through their majority in the Central Committee.

USE CASE: CDD IN A SOFTWARE DESIGN PROCESS

The company *Omega* adopted the CDD concept. Omega is concerned with the development and sale of products that help to connect PC and TV. The development of a new media centre *OmegaTV* was studied as an example of the involvement of a user community in the design process. A Media Centre software typically offers functionalities including pausing and recording live TV, managing existing video-, audio- and image-files and streaming media files to other clients.

Setting

Omega provides an online community space for its users. The portal consists of a wiki system, which allows users to share their knowledge about Omega's products, and a forum, which serves as a platform for information exchange between users and Omega employees, which usually revolves around problems as well as improvements.

An active community has been established over the last three years; circa 200 of the more than 15.000 registered users – on average, there are ten new registrations a day – regularly take part in discussions. The ten most active users, including one member of the Omega team, contribute approximately 30% to the discussions. The whole team consists of ten persons, who provide roughly 12% of all articles. Moderators as well as beta testers are private users who take on additional duties voluntarily and enjoy the confidence of the Omega team. There were nine moderators and eleven beta-testers at the beginning of the project. As the users showed an immense interest in the discussions and were willing to take on additional duties, this community seemed to be suitable for the exploration of the possibilities to involve users in the design and development processes of new software.

Preparation

The Omega team had introduced the CDD concept both in forum and weekly newsletter several months before the project started. The members of the forum had the opportunity to apply for seats in the User Parliament and the Central Committee via an online form. The original plan included only 30 persons for the User Parliament, but as each applicant seemed to be highly motivated and reliable, all 70 applicants were allowed to enter.

The members of the Central Committee were elected by the Omega team and the moderators of the forum. The moderators are nine private users who work on a voluntary basis to ensure compliance with the rules of the forum. Omega has been cooperating with these persons for a long time. All received applications for the Central Committee were presented in the internal moderators' forum. The moderators and the Omega team quickly agreed on four users, who were convincing due to competence and their ability to lead discussions and de-escalate situations. Furthermore, it was important that the members of the Central Committee would have enough time to work on the project. These demands quickly led to a unanimous decision. From the Omega team's side the Central Committee was complemented by the product manager, the product supervisor and the quality manager.

The co-operation began with the Central Committee's kickoff workshop where all members met in person. This first meeting aimed at introducing the technical framework and unalterable definitions by the developers, as well as formulating a basic concept for the project procedure. The User Parliament started working when the first technical preview, which provided the already defined principles of the software, was published.

Progress

The participation project was divided into two phases. The first phase started with the beginning of the project and ended after four months, when the first alpha version was completed. The second phase started with the publication of the alpha version and lasted until the final version of OmegaTV was published. This phase took about four months, too. While discussions and the creation of requirement specifications were of central importance in phase 1, phase 2 was mainly concerned with improving the alpha versions: five alpha and one beta version were presented to the users up to the end of phase 2 (see Fig. 1).

The discussions of the User Parliament were held in a special forum where write access was restricted to its members. New entries could be written as text, as text with attachments or as surveys. By initiating a survey, a user was able to ask other users to vote on different alternatives. According to the guidelines, every entry had to be labeled as question, idea, survey or discussion. The members of the Central Committee had their own forum, as well, although it was mainly used for making appointments. They contributed their ideas and opinions to the User Parliament forum. Every member of the Central Committee specialized in cer-

tain topics depending on personal interest. They took part in discussions and worked as moderators in these areas.

The product manager summarized important forum discussions and sent them to the members of the Central Committee as a basis for the weekly conference call. The average duration of those conferences was two hours, in which previously defined topics were discussed intensively and decisions were made in the according areas. The results of every conference call were published in the wiki system. The requirements listed there served as the basis for the requirement specification the developers used to implement the system.

Methodology

The use case was studied both qualitatively and quantitatively. The quantitative analysis was concerned with the participant’s forum entries. The gathered data allowed us to make statements about the community itself (how many users, how many entries) as well as about the participation of individual users with regard to certain topics over a particular period of time. To offer an additional feedback tool, we incorporated a plug-in module into the first beta version of the software that enabled users to assess single modules by several aspects using scales.

The qualitative analysis consisted of the evaluation of the entries in forum and wiki, and the conduction of semi structured interviews with 14 representatives of the different committees. Both users and employees were interviewed: six members of the Omega team, two of whom were members of the Central Committee, and eight users, four members of the User Parliament and four of the Central Committee.

Each person was interviewed twice. The interviews held at the beginning were mainly concerned with the participants’ motivation and the conditions for the project. Later interviews targeted at possible alterations in the participant’s opinion: Did the project meet the expectations and did the attitude towards a CDD process change? The interviews with employees took up to 30 minutes, those with users up to 23 minutes. All of them were recorded for later analysis. While the members of the Omega team and the Central Committee were interviewed in person in the Omega office or during the workshop of the Central Committee, the members of the User Parliament were interviewed by telephone.

Experiences gathered in phase I

Especially at the beginning it was striking how motivated and enthusiastic the users started their discussions. About 2000 entries were written in the first three weeks, and several of these were rather long. However, only 49 of the 70 User Parliament members took part in this first discussions, and the intensity of participation among them differed. After five weeks, more than half of the participants had written less than 15 articles each, on contrast to the twelve persons who had written more than 80 each, one of them even more than 200. The threads included usually between five and 50 articles, one contained even more than 300.

Most users participated in the project because they got the opportunity to influence the development process by bringing in their own opinions and ideas. Some users called it

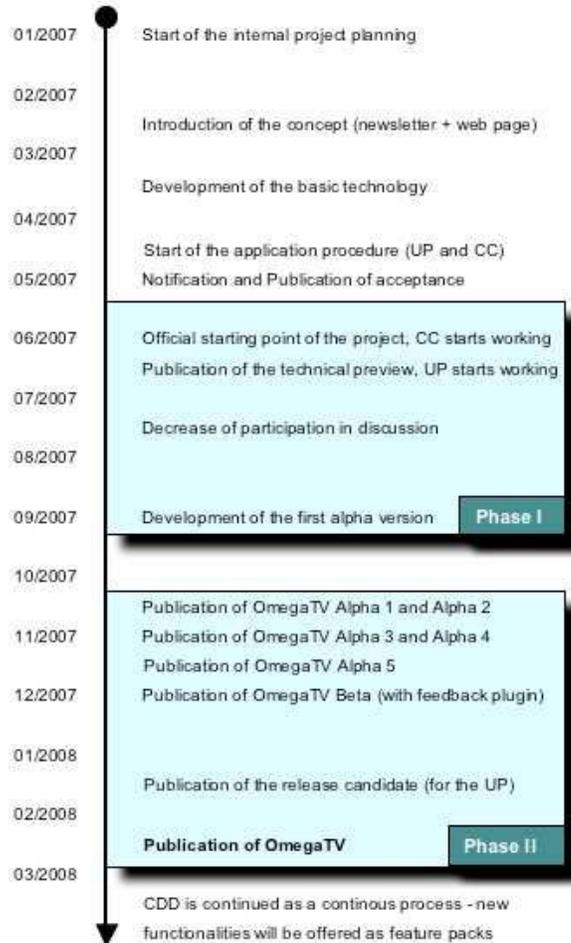


Figure 1: Timeline and milestones

egoism, but others were convinced that they helped the developers. This mirrors the fact that the involvement of users has advantages for both sides, users and developers. The consumers’ experience of the every-day use of the products could provide perspectives the developers were not familiar with before. Some users knew already from the beginning which particular functions they wanted to realize, others were concerned with more general factors such as stability and usability. Several participants enjoyed the cooperation with other users and developers. No user had financial reasons for his/her contributions, all worked on a voluntary basis. Some earlier versions of the software had not met the market’s needs, which resulted in the implementation of features that were not used afterwards. This should now be avoided through the incorporation of the users in the development process. A good precondition for the CDD was the mutual trust between users and developers that was based on positive experiences in the cooperation with beta-testers

and moderators. Particularly the team manager and the spokesman saw the involvement of the users as 'essential' and an 'inevitable development' caused by the market. On the other hand, the employees knew that this procedure had certain risks and therefore encountered the project with different degrees of motivation. While the developers were mostly skeptical, team and product manager were more open-minded. One developer described a 'queasy feeling that it might turn out to be a bottomless pit', but argued in the same sentence 'on the other hand one has the chance [...] of getting good ideas and suggestions' [employee]. Because the employees were paid by the company, their motivation was completely different compared to the users'. They were merely doing their job, and not experimenting with new methods (except for the management). Apart from this observation a problem almost every employee mentioned was the time factor. The existence of time limits was denied by the persons in charge, but deadlines that should be met were known internally.

The project management feared that users might have too high and too naive expectations and might therefore be frustrated if their ideas could not be considered. However, it was not predictable beforehand how instructive the project would be and if it would be a profitable or an unprofitable investment. Ideally, CDD could provide a perfect system to meet users' needs and therefore allow Omega to acquire the leading position in the market. From the users' point of view, there were hardly any risks. A successful project could supply them with software that would be closer to their needs, but even a failure would not bring any disadvantages apart from the loss of time.

A major problem in the Central Committee was the amount of ideas and opinions that were uttered in the User Parliament and that had to be reduced to a clear list of decisions. Especially at the beginning it was extremely difficult to keep up with the speed of discussions, because the Central Committee only had two hours a week to make decisions. Discussions were further complicated by the fact that nobody in the Central Committee wanted to be responsible for decisions at the beginning; thus, discussions were frequent, but results rare. This difficulty was solved after a couple of weeks, but when the project manager became ill then, no member of the Central Committee was willing to write the entries for the wiki system. Because of this, it took a long time until decisions were published. In several cases, the members of the Central Committee disagreed on the accuracy of the entries, because the decisions were made too long ago. A further negative effect of the delay was that the members of the User Parliament felt excluded from the development process; they explained this with a lack of the employees' motivation. In reality, this phenomenon was caused by a lack of time and human resources. Nevertheless, almost everybody in the User Parliament worried about this at the end of phase 1. This feeling increased when the first alpha version was delayed again and again and the members of the User Parliament did not get any new information. 'I was hoping for faster and newer mate-

rial. Something should have been implemented much faster to keep the boiler steaming. Always feeding in so that people have something to talk about' [member of UP]. Perplexity and dissatisfaction of the users were the result.

Experiences gathered in phase II

The users' reactions to the late publication of the alpha version were diverse. The general impression in the public forum was positive; users were looking forward to the final version. In the User Parliament on the other hand, the situation was tense, some persons even claimed that the CDD had failed. The small amount of changes in comparison to the technical preview and the outward appearance disappointed some participants and intensified their previously awakened suspicions. The technical preview, which had been introduced at the beginning of the project, had some unchangeable aspects that restricted the possibilities for innovation, especially regarding design and control. From the company's point of view it seemed reasonable to provide a first impression that should help the users to be more creative. Many participants however considered particularly these aspects as important and came up with many ideas and suggestions that were disregarded in the first phase of the project. At the beginning of phase 2, those discussions were started again and resulted in a new flexible user interface. Contrary to some users' suspicions, this had been planned from the beginning and had also been published in the wiki system.

The second point that made users doubt their influence on the development process was the fact that a major part of the realized suggestions were ideas of the product manager. He was able to express his ideas to the users in a very concise and convincing way, so that the majority followed his ideas. Although this behavior was not intended by the concept, a member of the Omega team and a user both agreed that if the product manager had not participated in the creative process, there would not have been enough coherent ideas. The product manager's participation had originated in the need for ideas, because the other User Parliament members were too reserved.

The situation within the Central Committee was tense, too. The employees complained that most tasks of the Central Committee were left to them and that they did not have enough time to perform their actual duties. In the interviews, all members of the Central Committee agreed that there was no clear task assignment within the commission. The employees felt left alone with their problems, while the users thought they were excluded from the development process even inside the Central Committee. 'We sometimes felt uninvolved although we were an internal group' [user in CC]. An Omega employee replied: 'If one wants to do User Driven Development one has to communicate, to gather information and give information back. We nearly had to employ a fulltime worker, this could have worked reasonably then' [employee]. Apart from problems regarding communication, there was a lack of clear guidelines and of a continuous moderation on the part of the project man-

agement. Even the Omega employees complained about the fact ‘that there were no guidelines, so that even a member of the Central Committee did not know what he had to do’ [employee in CC].

The use of the wiki system was problematic concerning the communication behavior of all participants. The decisions of the Central Committee were meant to be summarized there as a central requirement specification to enable every user to inform himself about the status of the decision making process. Unfortunately, few users inspected this document regularly. The product manager said: ‘Nobody wanted to make the effort to sit down and read that thoroughly. I inquired every now and then and the answer was always ‘Yes - everything is fine’; and then, when the first version was published, all the prophecies of doom came’ [employee in CC]. Although the wiki system was accessed 120.000 times, actual reactions to it were rare. But when the same topics were entered in the forum, long discussions developed. The change of media between forum and wiki system seemed to be problematic. The users explained their behavior with the lack of notifications when new entries were written. In fact, the wiki system has a changelog, but the requirement specifications were written in a continuous entry that the participants had to search completely when they wanted to find the new information. Because of the extremely slow creation of new entries due to the bad task assignment in the Central Committee, many users avoided the wiki system and relied only on the forum. However, the Central Committee did not use the forum for the announcement of decisions, because that was meant to be done in the wiki system. The fact that very few users observed the development of the requirement specifications regularly became clear when the first alpha version was published. Many users complained about certain aspects of the software that had been described month before in the wiki system. The members of the Central Committee reacted to these complaints with anger and frustration. Some members of the User Parliament said that the decisions of the Central Committee did not mirror the participating users’ opinion. A User Parliament member replied: ‘I don’t understand the whole excitement. 1. Everything is in the wiki and it looks exactly like what is described in the wiki. 2. I haven’t seen complaints about what is written in the wiki [...] Then let us change the wiki’ [member of UP].

A further rather complicated aspect of the CDD is the long-term cooperation of many persons who are not familiar with that way of working. Discussions were often very emotional. A User Parliament member expressed that: ‘A community lives through discussions; otherwise, it wouldn’t be a community. But even when discussions heat up, everybody will hopefully get along with everybody else in the end’ [member of UP]. From time to time, the users’ unstructured procedure was a problem. It was difficult that ‘nobody was experienced in the development of products, in structuring the process, and then everything goes haywire’ [member of UP]. On the other hand, users’ unconventional thinking can lead to surprising ideas: ‘Somebody

made this half-baked suggestion and three people polished that and at some point it turned out to be a very good idea’ [member of UP].

Some developers drew attention to the fact that the User Parliament had come up with many interesting ideas, but the participation within that commission was considered to be too low. Only 15 of originally 70 persons contributed regularly during the whole project. According to the Omega employees, this was not enough. ‘If we had had more persons, the discussions might have been more lively and other people would not have lost interest so quickly’ [employee in CC]. 70 users, as originally suggested, would have been a good number for this intention. Besides the advantage that more users could have come up with more ideas, more opinions would have specified the relevance of the ideas for the whole target group, not only for individuals. ‘The smaller the number of users and the number of people who contribute to the discussions, the harder it is to say if something is reasonable or not. And then, too many personal arguments become important’ [employee in CC]. Low participation in the User Parliament did not put enough pressure on the members of the Central Committee to adjust their decisions to the public opinion. Therefore, interests of the Central Committee might have influenced the process above average.

Another problem was the composition of the Central Committee. Most participants considered it to be too small and especially from the users’ point of view inadequately staffed. Due to the illness of a user, users lost the majority in the Central Committee. When arguments about the further process started in phase 2, another user left the Committee with the consequence that three employees and only two users were left. One developer hardly participated in discussions and conference calls, although he, as a software engineer and project manager, could have provided helpful information regarding the practicability of ideas. On the part of the users, professional and private commitments did occasionally not leave enough time to work on the project. The organization and cooperation of the Central Committee were difficult, too. Appointments for conference calls and meetings were often complicated and resulted in only two face-to-face meetings during the whole project. Although the weekly conference calls lasted mostly about two hours, some persons were not able to make their voices heard.

While the members of the User Parliament reacted irritated and emotional to the first alpha version, the users in the public forum were more enthusiastic. Many aspects of the software were praised and the overall impression seemed to be positive. Even when a member of the User Parliament quoted some arguments that had led to heated discussions in the User Parliament forum, these arguments were put down in the public forum. A major part of the discussions shifted to the public forum in the second phase of the project. This enabled all users to describe their experiences with the alpha and beta versions. 141 users participated in these discussions in addition to the User Parliament, al-

though most of them only reported mistakes. No more than 30 persons participated regularly and made suggestions for improvement. Although every alpha version was downloaded about 500 times, the number of reactions decreased heavily from the first to the fifth version. While the reactions to the first alpha version were quite emotional, users were more satisfied with the faster realization of decisions afterwards. The reactions to the later versions were almost exclusively problem reports.

FEEDBACK PLUGIN

Alpha and beta versions of the resulting software could be downloaded by everybody who was interested. In order to get a quantitative feedback from the whole community we integrated a feedback plug-in into one of the prototypes. Due to the limited feedback possibilities with a remote control, we decided to use a method which allows very easy and fast feedback and at the same time distracts the user from his media consumption as little as possible. By using a scale based rating, users could rate all of the different modules. We separated each module into the categories 'functionality', 'usability' and 'design'. Every category could be ranked on a scale from 'very good', 'okay', and 'sufficient' to 'bad' as a clear statement of dissatisfaction. After the module had been used for a predefined time (we chose the fifth start of a module), a pop-up window appeared which requested a scale based rating from the user. This easy evaluation design should result in a high user involvement. If the user did not want to participate in the voting s/he could cancel the process easily. The ratings were automatically stored on the server-side. The results of the survey were displayed afterwards on automatically generated web pages. This feedback option mainly fulfilled two central goals: (1) determining whether the prototype fulfilled the expectations of the users and (2) helping team members to draw attention to the low rated aspects for problem-focused discussions.

The possibility to submit a feedback was appreciated by the users. Many users who did not participate in the CDD process gave their opinion that way. 707 users rated the TV module in the seven-week long test period. This module was rated most, because nearly all participants used it. In contrast to this only 35 persons rated the picture module.

While the first reactions from the User Parliament were negative, the results of the feedback plug-in gave a positive overall picture. 'Design', 'functionality' and 'usability' of the separate modules were rated by two thirds of the users as 'very good' or 'okay'. Even though the results could only reflect a first tendency, it seems that the majority was satisfied with the first results.

DISCUSSION

Motivation: The users' motivation to take part in a CDD process was very high, especially at the beginning of the project. The possibility to be part of the development process and to bring in own ideas was considered to be valuable. However, from 70 members of the Parliament only 49 persons participated in the first design process. Only 15 us-

ers took part in the project continuously. On the other hand, 30 new participants joined the project (informal) with regular contributions to discussions after the first prototype was released. By interpreting these results we can say that a CDD approach cannot be planned with a fixed number of (user) participants. Users drop out, new ones want to get in – the membership in the User Parliament should therefore be open and dynamic. It could be useful to define a minimum size, which can be filled up with new participants.

The team's motivation to participate in such a project was difficult to access. On the one hand the management saw a great potential and on the other hand the developers were quite reserved, especially at the beginning of the project. This separation also could be observed in the Central Committee – while the product manager was the main driving force, the developer participated only occasionally in discussions. From his point of view, the CDD only disturbed his usual work. The manager on the other hand has been familiar with media center systems for many years and initiated many discussions in the User Parliament. Because his ideas were accepted very often, he had a strong influence on the design process. However, we believe that this steering intervention from the company's side is absolutely essential and cannot be avoided. Users participate only partly in topics they are interested in. But someone has to stimulate discussions for every relevant topic and when the feedback is sparse then this person has to make a decision as well.

The whole second project phase (see also Figure 1) ran more smoothly. After most of the mentioned criticisms were considered, the amount of discussion contributions for the next alpha version were phased down. After the release of the last two alpha versions there were only marginal contributions, namely the submission of program errors. Interpreting this finding we can say that the participants' motivation and engagement are on the highest level in earlier design stages when the boundaries are open.

Technology: One aspect we regarded as critical already in the starting phase was the used technical infrastructure to support communication. We optimistically expected that users could handle a CDD process with the use of the existing and familiar infrastructure (forum and wiki). Unfortunately, this assumption was wrong. The existing infrastructure was not applicable for a highly dynamical process like CDD. The separation in discussion (forum) and functional specification (wiki) resulted in an environment in which both tools were seen as independent instances with different responsibilities. For the members of the Central Committee, the wiki was the center of reference; for the members of the User Parliament, the statements in the forum discussions were regarded as important. One option to deal with this problem could be the use of cross links from discussion area to the specification area. However, a similar function offered by Fischer in his DynaSite prototype [5] had not been accepted well in his study. Besides the bad linking of discussion and specification another problem

was the presentation of the specifications in one document. Although the wiki has a changelog function, it was obvious that the readability of the specification did not fulfill users' needs. Furthermore, the document contained many images and screenshots, and it therefore took quite a long time to download it, especially for users with a low bandwidth. By taking these observations into account, every module and every subfunction should be specified in separate documents, which should also be linked to the discussion area and/or vice versa.

Organisation: The subdivision of the CDD concept into User Parliament and Central Committee has proven to be a suitable in principle. However, there were reservations from the User Parliament against the Central Committee, because members in the Committee were seen as favored in the direct communication with the team staff. Members of the Parliament thought that members of the Committee kept information secret or held information back. Concerning these matters, users of the Parliament stated that it would be helpful if the members of the Committee took turns regularly. Furthermore it could have value to make the communication within the Committee more transparent, e.g. by letting members of the Parliament participate in the weekly telephone conferences in a passive way or by recording the conferences and presenting the results in the online area afterwards. Another suggestion for early design processes is the use of personal group discussions. Users as well as team members were interested in module-oriented face-to-face workshops. However, such workshops are time consuming and demand organizational efforts. The planning of physical Committee meetings already failed because of temporal incompatibilities. Web conferences were therefore seen as an alternative to the weekly telephone conference sessions.

Yet another important aspect concerns a clear separation of roles and tasks of the members of the Central Committee. Different tasks like communicating with members of the Parliament, summarizing requirements or coordinating milestones should be assigned to Committee members in a predefined way. These clear and transparent assignment of roles should be made public, so that the members of the User Parliament know who is responsible for a particular task. This can help to correct misunderstandings faster. The temporal factor for the supervision as well as for the whole process is a critical issue, too. The time effort to supervise the project was much higher than expected at the beginning of the project. A full-time employee would have been needed only for the communication with the User Parliament. The process as a whole required more time than traditional software development. The duration of a CDD process cannot be predicted exactly, because users assign time for contribution individually. Nearly all participants of the CDD process mentioned that the provided time frame for the project was too short. One of the developers estimated in an interview that CDD would take approximately 50% longer than development without direct user participation.

Satisfaction: The comparison of the different reactions to the first prototype is quite interesting. While members of the User Parliament were disappointed to a certain degree, persons from the uninvolved online community gave positive feedback in the public forum. This circumstance combined with the quantitative results from our feedback plugin let us assume that CDD will lead to a good overall result. The negative comments by members of the User Parliament were probably a result of the fact that the developed prototype could only be a compromise between the different suggestions (as it was written in the wiki documentation). Users who participated in the project may be more dissatisfied because they invested more or less time and effort making a contribution to the process. Dissatisfaction will come up, when they see one of their own suggestion not considered.

Both users and employees appreciated their participation at the end of the study. Even though the process of the CDD was problematic at certain points, it was nevertheless 'a bigger success [...] that the method can apparently work and might work even better for other projects' [employee]. Except for one person, all employees were willing to conduct another CDD, but they would try to solve the now known problems. It was particularly important for them to allow more time for the project. The users that were interviewed by us were not as enthusiastic about participating in a similar project anytime soon, but they explained this with private commitments. In the concept itself they saw a lot of potential. 'User Driven Development works when certain things are clearly defined, tasks are clearly distributed, the team supports it and the communication with the users is good' [member of UP].

SUMMARY

In this paper we explored the involvement of users within the whole development process of a Home Entertainment system based on an existing virtual user community. We suggested a design of a participation arena with a User Parliament and a Central Committee as steering institutions. Our concept to include existing online user communities into a *Community Driven Software Development* process, in which participants are not only 'used' for giving feedback but also have the power to make decisions, was overall well accepted. But our study shows that especially the structures of professionalization lead to a power imbalance towards the designers' side even if that is not intended: what started as a nice private activity for the users in the Central Committee, felt like unpaid real work afterwards. The 'work character' of user participation was also illustrated by the demand for explicit and transparent roles for certain tasks, by the perceived need for self-organization among users, and by the efficiency concerns with regard to the technological infrastructure that was used. While Wikis and Web forums are sufficient to run a user community, a participation process demands more specialized technological support (e.g. with respect to references to other parts of the discussion or to design aspects under consideration) even if users are already familiar with the community infrastruc-

ture. Our experiment with a feedback module built into the software showed that the cost of articulation that comes with a comment in a forum is more likely to be paid by dissatisfied users – the user judgement based on the feedback module we provided was more positive than what could be read in the forums.

Using existing virtual user communities as a starting point for a PD process seems to be obvious, but the advantage of benefiting from an existing discussion culture needs to be exploited carefully. The normal discontinuities of participation in online communities can become a problem when they appear among user representatives in a process model like ours. As a consequence, we would suggest that more people participated in the steering committee on the users' side, and we would assign every role and responsibility to more than one user to create transparency and redundancy. With respect to duration and result quality, the process itself is hard to calculate even for the designers, but we would advise designers to articulate the framing conditions they see more. And finally, although users may be familiar with general community tools, specialized tool support may increase the quantity of participation as well as the quality of articulations (e.g. by referring to representations of the technology).

REFERENCES

1. Armstrong, A.G., and Hagel, J. The real value of online communities. *Creating value in the network economy, Harvard Business Review Book Series* (Boston, MA, USA 1995), 173-185.
2. Barcellini, F., Détienne, F., and Burkhardt, J.M. Users' participation to the design process in a Free Open Source Software Online Community. *Proc. of the 18th workshop Psychology of Programming PPIG'06* (2006), 99-114.
3. Bjercknes, G., and Bratteteig, T. User participation and democracy: a discussion of Scandinavian research on system development. *Scandinavian Journal of Information Systems* 7 (1995), Issue 1, 73-98.
4. Farshchian, B.A., and Diyitini, M. Using Email and WWW in a distributed participatory design project. *SIGGROUP Bulletin* 20 (1999), Issue 1, 10-15.
5. Fischer, G., and Ostwald, J. Seeding, Evolutionary Growth, and Reseeding: Enriching Participatory Design with Informed Participation. Binder, T., Gregory, J., and Wagner, J. (eds.). *Proc. of the Participatory Design Conference* (Malmö University, Sweden, CPSR, Palo Alto, CA, June 2002), 135-143.
6. Fischer, G., and Scharff, E. Meta-Design – Design for Designers. Boyarski, D., and Kellogg, W. *Proc. of the third International Conference on Designing Interactive Systems* (New York City, August 2000), ACM, 396-405.
7. Floyd, C., Reisin, F.-M., and Schmidt, G. STEPS to Software Development with Users. Ghezzi, C., and McDermid, J.A. *ESEC '89* (Berlin/Heidelberg, Germany, 1989), Springer Verlag, 48-64.
8. Franke, N., von Hippel, E., and Schreier, M. Finding commercially attractive user innovations: A test of lead user theory. *Journal of Product Innovation Management* (MIT Press, MA, USA 2005).
9. Füller, J., Bartl, M., Ernst, H., and Mühlbacher, H. Community based innovation: How to integrate members of virtual communities into new product development. *Electronic Commerce Research*, Volume 6 (Norwell MA, 2006), Issue 1, 57-73.
10. Gopal, A., Mukhopadhyay, T., and Krishnan, M. Virtual Extension: The Role of Software Processes and Communication in Offshore Software Development. *Com. of the ACM* 45 (2002), Issue 4, 193-200.
11. Heiskanen, A., and Similä, J. Gatekeepers in the action structure of software contracting: A case study of the evolution of user-developer relationships. *ACM SIGCPR Computer Personell 14* (New York, NY, USA 1992), Issue 1-2, 30-44.
12. von Hippel, E. Lead Users: A Source of novel product concepts. *Management Science* 32 (1986), 791-805.
13. Kannan, P.K., Chang, A.M., and Whinston, A.B. Electronic Communities in E-Business: Their Role and Issues. *Information Systems Frontiers 1* (Hingham, MA, USA 2000), Issue 4, 415-426.
14. Keil, M., and Carmel, E. Customer-Developer Links in Software Development. *Communications of the ACM* 38 (1995), Issue 5, 33-44.
15. Lazar, J., and Preece, J. Social Considerations in Online Communities: Usability, Sociability, and Success Factors. Van Oostendorp, H. *Cognition in a Digital World*, LEA Inc. 2002.
16. Lieberman, H., Paternò, F., Klann, M., and Wulf, V. *End-user development*. Springer, Netherlands 2006.
17. Muller, M.J., Wildman, D.M., and White, E.A. Taxonomy of Participatory Design Practices. *Posters and Short Talks of the 1992 SIGCHI Conference on Human Factors in Computing Systems* (Monterey, California, USA 1992), 34.
18. Nuseibeh, B., and Easterbrook, S. Requirements Engineering: A Roadmap. Finkelstein, A. (eds.). *The Future of software engineering, ICSE* (2000), 35-46.
19. Pipek, V. From Tailoring to Appropriation Support: Negotiating Groupware Usage; PhD Thesis, Faculty of Science, Department of Information Processing Science (ACTA UNIVERSITATIS OULUENSIS A 430), University of Oulu, Oulu, Finland, 2005, 246 p..
20. Pipek, V., and Syrjänen, A.-L. Infrastructuring as Capturing In-Situ Design, 7th Mediterranean Conference on Information Systems, Association of Information Systems, Venice, Italy, 2006.
21. Rheingold, H. *The Virtual Community – Homestading on the Electronic Frontier*. Addison Wesley, Reading, MA, USA 1993.
22. Sleeswijk Visser, F., and Visser, V. Re-using users co-create and co-evaluate. *Personal and Ubiquitous Computing* 10, 2-3 (2006), 148-152.
23. Wellman, B. Virtual Community: Introducing a New SIGGROUP Focus Area. *ACM SIGGROUP Bulletin* 9 (New York, USA 1998), Issue 1, 18-20.
24. Wulf, V., Pipek, V., and Won, M. Component-based Tailorability: Towards Highly Flexible Software Applications. *International Journal on Human-Computer Studies (IJHCS)*, 66 (1) (2008), 1-22.