

Distributed Participatory Design – A Case Study

Dorina C. Gumm

University of Hamburg
Department of Informatics
gumm@informatik.uni-hamburg.de

Monique Janneck

University of Hamburg
Department of Informatics
janneck@informatik.uni-hamburg.de

Matthias Finck

University of Hamburg
Department of Informatics
finck@informatik.uni-hamburg.de

INTRODUCTION

Many projects nowadays are distributed in several ways, be it that dispersed developers are working together or that users of a product are distributed. The challenges distributed projects face vary according to different dimensions of distribution [11]: individuals, stakeholder groups and other entities can be distributed physically, organizationally or temporally.

Physical distribution refers to different locations of people or other things, e.g. different floors in the same building, different countries or even continents. Organizational distribution is related to the structures people are working in. The term organization is used to refer to companies as well as to any other structure describing the condition or state of organized work. Temporal distribution refers to the synchronicity of working hours, i.e. the time people are available for synchronous interaction [5]. The separation by time can be rooted in physical distribution and thus distribution among different time zones; but also be caused by shift work or working rhythms (e.g. if people work part-time).

Participatory Design is concerned with several issues in three areas [12]: (1) the politics of design, (2) the nature of participation and (3) tools and techniques. The main concern in the second area is the establishment of a close relationship between users and developers to support mutual learning and participation of users in decision making. The strength of PD lies here in bridging the organizational distance especially between users and developers.

The challenges of distributed participatory design are located in two areas:

1) The PD concept of real participation and the concept of physical distribution tend to conflict each other [10]. Most PD approaches are based on the possibility to have face-to-face meetings; physical distribution is rarely in focus. However, more and more projects explore the possibilities of participatory design in physically distributed project settings (e.g. [1,3,4,6]).

2) PD approaches do not address the organisational distribution within the user group or between different user groups, respectively.

In this paper we report on a long-term project facing the challenge that the amount of users and application domains, respectively, have increased continually. Users and designers as well as the user group itself are distributed organiza-

tionally and physically. This distribution in turn challenges the participatory process that has been used from the very beginning when the user group was small. However, we still aim for a participatory design process and use several methods to support participation of distributed user groups. In the following, we introduce the case study and motivate the need for distributed users' participation. After that we present practices we used and that proved helpful to support a distributed participatory design process (summarized in Table 1). We finish this paper with a conclusion.

THE CASE STUDY COMMSY

CommSy is a web-based groupware system developed to support communication and coordination in working and learning groups by facilitating the exchange of documents and the sharing of important notes and dates between users, comparable to e.g. BSCW, phpBB, or Moodle. In contrast to other CSCW systems, though, community support has always been a top priority, hence the name that stands for *community system*. Development started in May 1999 at the Department of Informatics at the University of Hamburg, and the initial system was designed to support the communication and coordination in learning groups at the department – especially as support for highly self-organizing computer science courses. Nowadays the software is used by several universities, schools, commercial networks, and small companies. Furthermore, the software is Open Source – meaning that the design team itself is distributed mainly organizationally, with developers working in different research projects or companies, targeting different user groups and pursuing different interests, ranging from research to commercial interests in a spin-off application service providing company.

In addition, users and user groups are highly distributed physically as well as organizationally: For example, CommSy is used by a variety of departments at the University of Hamburg and also a dozen schools spread throughout the whole city. Furthermore, several universities all over Germany use the system, as well as some virtual networks of freelancers, whose members are themselves dispersed all over the country.

The physical distribution makes it difficult to bring users and developers together face-to-face, even though many users have a keen interest in participating in the design process: Traveling is costly in terms of time and money.

The organizational distribution between different user groups turned out to be an even bigger challenge than the physical distribution. CommSy is applied in very different contexts. Within the university, CommSy is used by very different disciplines like computer science, pedagogy, or linguistics. Besides the university context CommSy is used in schools, freelancer networks, and companies. Due to this wide application field, users come from different organizations and thus have different backgrounds and attitudes. This organizational distribution causes challenges like different and sometimes contradicting requirements on functionality or usability, which in turn threatens a common system vision: Some of the specific requirements elaborated within one user group turned out to be of little or no significance in the other contexts. This grew into a problem for the development team: On the one hand, exploring new contexts of use – and thus, new customers – was vital to the commercial interests of the spin-off company. On the other hand, the development team needed to bundle resources and tried to avoid parallel implementations that would increase the complexity of software and difficulty of administration and maintenance. Furthermore, they wanted to avoid the high risk of fragmentation of the development team and process, which is known as *forking* in open source projects. Also, the different requirements had to be aligned with the original design philosophy.

To meet this new challenge of pooling the interests of different communities and the development team, new ways of bringing users from different contexts together had to be established to balance their respective needs and upcoming requirements.

GOOD PRACTICES

We now describe our experiences with some of the methods used to involve users in the distributed design process.

Mediated Two-Directional Feedback

Mediated feedback relates to feedback that is gathered through various media channels. These become necessary with a physically distributed and/or unknown user group. Media channels could either be technical like E-mail, bug trackers etc. [4], or human [9].

In our project, persons acting as mediators – e.g., central actors within certain user groups, such as moderators – support the participatory process well. They collect feedback of user groups they belong to and pass it on to the developers. This feedback includes usability problems, ways of usage as well as bugs. In addition, these mediators pass feedback from the developers back to their user group, e.g. when bugs will be fixed or why certain design decisions were made. This kind of mediated two-directional feedback requires (a) that the mediators know members of the development team and have a regular exchange with them; (b) that users accept that maybe confidential discussions about the software usage are passed on to the developer team; (c) that

developers provide ample feedback to the mediators in order for them to inform their user group and (d) that the mediators tolerate additional work.

Mediators play a somewhat similar role to *key users* who use software intensively and report first-hand experiences back to the development team: In our project, mediators also tend to be very active users. However, their two-directional role as intermediaries between the design team and their user group clearly exceeds the key user concept.

The practice of mediated feedback addresses the organizational and temporal dimension of distribution. The mediators bridge the organizational gap between users and developers. Users normally do not have access to the developers for several reasons. In turn, it is difficult for the developers to get known to the users. The mediator is much easier to contact. A mediator also bridges temporal distribution since this person may communicate with developers and users at different times. Thus, developers and users are not dependent on working hours and places.

Intercontextual User Workshops

We use so-called *Intercontextual User Workshops* to reduce distance between users and developers and between different user groups. These workshops are conducted in addition to *development workshops* – a blend of the future workshop (cf. [8]) and priority workshop techniques (cf. [2]) – which take place within the development team in close cooperation with special users to analyze and discuss requirements and plans for future developments.

On the one hand, intercontextual user workshops help to reduce physical distance: With these workshops we offer a face-to-face meeting that takes place regularly about once or twice a year to bring users – typically teachers and facilitators of project groups, respectively – and developers together. On the other hand, these workshops serve the purpose of bringing users from different contexts together to initiate an exchange of user experiences in a variety of application domains. When organizing these workshops we need to take into account the different working conditions, time schedules, and constraints (e.g. term breaks, holidays, freelancers' high workload phases) of the organizationally distributed user community.

Typically about ten to fifteen users and three to five developers attend the one-day workshops. These workshops start with an extensive introductory round to help participants to get to know each other and the respective contexts of use. Afterwards, the developers give a presentation of planned and upcoming developments. This is usually the starting point for a longer discussion of use experiences, problems, and requirements. The moderators visualize the contributions for clustering later on. Sometimes the participants wish to work on different issues emerging from the discussion in smaller groups. Topics often address usage problems or phenomena that are shared by many participants,

e.g., how to increase active participation in online communication.

These workshops help to bridge the organizational and physical distance between users from different domains. Judging from the feedback provided to us by the participants, they benefit highly from exchanging experiences with others in this way. Crossing perspectives with users from different contexts allows them to reflect on their own usage from another angle than this was possible within their own community of practice. Challenged by other participants to explain and motivate more clearly why, e.g., certain features were important to them, and contrasting this to experiences from other backgrounds, they start to think through and sometimes question their use routines.

Commented Case Studies

To save the workshop results and to distribute them among a larger group of users we established a new form of documenting user experiences, called *Commented Case Studies*. Just like the intercontextual user workshops, this method is aimed at enabling an exchange between users of different communities of practice and also between users and developers. Furthermore, it serves to communicate design decisions among a larger group of people involved in the development process.

Commented case studies can be compared to use cases or scenario techniques employed in software engineering [13]. While use cases are highly formalized and detailed descriptions relating to the way concrete implementations are planned or carried out, commented case studies describe existing experiences of use in a more anecdotal, less formalized way. Scenarios, on the other hand, which are similar to commented case studies in their narrative descriptions of tasks and ways of use, are meant to capture typical, representative descriptions of use, while commented case studies are highly individual. They consist of indexed and annotated typical descriptions of use written by real users, and an explanation of design decisions on the basis of these case studies. They follow an informally proposed structure, including a short description of the use context and purpose, the participants, the way the software was introduced and adopted in the respective setting, and an extensive report of use experiences and ‘lessons learned’. An extensive introduction written by the editors gives an overview and classification of the cases presented and helps readers to select the ones that are most relevant or interesting for their purposes (cf. [6]).

With this structure commented case studies address both users and developers. For users they reduce the distance to other users groups by offering them access to unaltered description of other users’ experiences. Furthermore, by commenting the case descriptions, developers document design decisions in a transparent way, enabling users to gain insights about how and why the system is developed in the

way it is. By bundling authentic, unedited reports from different use contexts in a comparable way, the commented case studies reduce distance for developers who are not in direct contact with users of certain contexts: get the chance to put themselves in their position and develop a more thorough understanding of their requirements beyond simple feature requests.

In contrast to the intercontextual user workshops, the commented case studies address less the organizational than the physical and temporal dimensions of distribution. This method opens up a way for users to take an active part in the participative design process even when they are not able to participate in a workshop due to organizational or temporal restrictions.

Surveys

In addition to methods aiming at establishing close contact with a – naturally – limited number of users, we also wanted to provide a feedback channel to the bulk of users who were not available for close and personal cooperation. As the user numbers soared, the proportion of users that we could get in touch with personally grew smaller and smaller. Furthermore, certain user groups were more readily available for cooperation than others: In the university context, for example, it was much easier to get in touch with lecturers – who were known to us by name – than with the rather anonymous bulk of students.

Therefore, we use *web surveys* regularly once or twice a year to evaluate usage in different contexts and by different user groups. All users are invited via email to participate in the survey. These surveys address how and for what purposes CommSy is used, what works out well and which problems arise during use. Furthermore, general user satisfaction as well as users’ assessment of certain features and design decisions is measured. In addition, users get the opportunity to describe their wishes for further development of CommSy in as much detail as they like.

The survey data enables us to evaluate the system’s usability and usefulness in different use contexts. Requirements and hypotheses elicited in close cooperation with key users or mediators – e.g. the need for a certain feature – can be validated on a broader scale.

Surveys can be used as a means to bridge physical, temporal as well as organizational distance, with the latter being our main concern in order to reach different user groups who are – like students and teachers – not necessarily physically dispersed.

User Support

Besides the methods described above which aim specifically at involving users in the design process, providing *user support* can also be employed as a means for participation.

For CommSy users, we provide extensive user support, in-

cluding documentation (like FAQs, a user handbook and a moderator’s handbook) and especially email support. Users receive a personal, non-standardized answer to their questions and bug reports, which is highly appreciated: Users often send extensive “Thank you” mails after being helped and also prefer addressing a concrete person (especially one they have been in touch with before) instead of using an “anonymous” email address (like e.g. “support@comm-sy.de”).

The people providing support collect and classify bugs and usage problems and report them back to the development team. For the developers, this is a valuable instrument to get first-hand feedback from users.

Table 1 summarizes the presented practices, showing which dimensions of distribution are addressed by the respective practices. The characters in the table refer to the degree of how much/good a practice address the respective dimension: (+) means ‘well’, (o) means ‘more or less’ and (-) means ‘not addressed’.

Practices/ Dimensions	<i>Physical</i>	<i>Organizational</i>	<i>Temporal</i>
<i>Mediated Feedback</i>	-	+	o
<i>Workshops</i>	+	+	-
<i>Commented Case Studies</i>	+	+	+
<i>Surveys</i>	+	+	+
<i>User Support</i>	+	o	o

Table 1: Participatory practices to address different dimensions of distribution

Conclusions

The five participatory design practices described above helped us to bridge organizational, physical as well as temporal distance within a highly distributed software development project, with *organizational distance* – relating in our case to a variety of different user groups with different requirements, preconditions, and needs – proving to be the greatest challenge for maintaining a participatory process as well as a single product development.

As table 1 shows, methods seldom address all three dimensions equally. As for the methods we propose, this is only true for commented case studies and web surveys – methods requiring little or no physical contact while allowing the involvement of a large number of users. However, this is not sufficient for a successful participatory design process: In our experience, close cooperation with a certain smaller group of users – e.g. mediators or key users – including regular face-to-face contact is essential to guarantee the amount of participation that is necessary for a high quality software product. Nevertheless, requirements and design decisions elaborated within a smaller group need to be vali-

dated on a broader scale, e.g. using the respective methods proposed above. Thus, we argue for the application of a mixture of different methods of user participation in a distributed design process.

For our future work, we are working on still more methods to address distribution and different users’ needs: For example, including schoolchildren in the design process is one challenge we will be increasingly facing. Furthermore, we aim at testing these methods in other software development contexts.

REFERENCES

1. Bleek, W.-G., Jeenicke, M., Klischewski, R. Framing Participatory Design Through e-Prototyping. In: *Proceedings of the Participatory Design Conference, CPSR* (2002), 300-305.
2. Braa, K. Priority Workshops: Springboard for User Participation in Redesign Activities. In: *Proc. Of COOCS 1995*, ACM Press (1995), 246-255.
3. Dearden, A., Naghsh, A., Ozcan, M. Support for participation in electronic paper prototyping. In: *Proceedings of the Participatory Design Conference, CPSR* (2004), 105-108.
4. Divitini, M., Farshchian, B. A., Tuikka, T. Internet-based groupware for user participation in product development. *SIGCHI Bulletin* (2000), 32 (1): 31-35.
5. Espinosa, J. A., Carmel, M. Modeling coordination costs due to time separation in global software teams. In: *Proceedings of the ICSE International Workshop on Global Software Development* (2003).
6. Finck, M., Gumm, D.-C., Pape, B. Using groupware for mediated feedback. In: *Proceedings of the Participatory Design Conference, CPSR* (2004), 45-48.
7. Finck, M., Obendorf, H., Pape, B. (eds): *Fallbeispiele der CommSy-Nutzung – Eine Sammlung von Nutzungsberichten*. Universität Hamburg, Fachbereich Informatik, Bericht FBI-HH-B-261/04 (2004).
8. Greenbaum, J., Kyng, M. (eds). *Design at Work: Cooperative Design of Computers Systems*, Lawrence Erlbaum Ass, New Jersey (1991).
9. Grudin, J. Interactive systems: Bridging the gap between developers and users. *Computer*, Vol 24, Issue 4 (1991), 59-69.
10. Gumm, D.-C. Distributed participatory design. In: *Proceedings of IRIS 29* (2006). Forthcoming.
11. Gumm, D.-C. Distributed software development – a taxonomy. *IEEE Software. Special Issue on GSD*, (2006). Forthcoming.
12. Kensing, F., Blomberg, J. Participatory design: Issues and concerns. *CSCW*, Vol 7 (1998), 167-185.

13. Rosson, M. B., Carroll, J. *Usability Engineering: Scenario-based Development of Human-Computer Interaction*, Morgan-Kaufman (2002).

Scenario-based Development of Human-Computer Interaction, Morgan-Kaufman (2002).