Teaching Participatory Design

K. Maike Hecht
Informatik Department
University of Bremen
28359 Bremen GERMANY
+49 421 218 64392
hecht@informatik.uni-bremen.de

Susanne Maass
Informatik Department
University of Bremen
28359 Bremen GERMANY
+49 421 218 64391
maass@informatik.uni-bremen.de

ABSTRACT
In this paper we reflect on how to teach methods used in participatory design (PD) to graduate students. We describe a course concept that has proved very successful both in terms of student satisfaction and course results during the last years. Lectures introduce the concept of PD and a wide variety of participatory methods for analysis and design; exercises prepared and facilitated by small groups of students permit hands-on experience and further the development of necessary group moderation skills. As there are only few descriptions of teaching concepts in the literature, we would like to invite the PD community to share their experiences and start a discussion about teaching participatory design.

Keywords
Participatory Design, Education, Teaching, Methods

INTRODUCTION
The concept of participatory design (PD) has been around for quite a while. It aims at giving users a voice in decisions about the design of their working environment and tools. PD brings together different stakeholders, trying to make them meet at eye level and work together on the analysis and the design of a computer system. Every stakeholder thereby gets a chance to understand the others' views and to benefit from their expertise.

While this sounds fairly simple and reasonable, often there are social and communication barriers and power imbalances between the different experts. For this reason, PD researchers advocate methods that facilitate co-operation in teams with mixed expertise. These participatory methods provide a "third space" [7] for designers and users to meet in. Artifacts or representations [11] that make sense to everybody facilitate cooperative work; they further mutual understanding and help making implicit knowledge explicit in the process.

Many determining factors have to be taken into account when the framework for a participatory design project is set up and methods to be applied are chosen; time and space, organizational constraints, number of potential participants, and the interests of different groups involved are only some of them. Wagner and Piccoli recently reported on their studies of project failures and pointed out that "… it is not enough to involve users. Software design teams must give careful consideration to the *timing* and *focus* of user participation" (emphasis in the original) [10, p.53]. Methods must be chosen well with regard to domain, participants, constraints etc. But how can this be accomplished?

Still there are only few publications that explicitly describe methods to be applied in the PD process on a step-by-step basis, few offer help with the choices to be taken, how to organize the whole design process, how to get access to the field, and to set up the interaction with users [3][2]. We wonder how people new to the idea of PD can learn to take their first steps. What kind of basic knowledge do they need? Is it all 'learning by doing', working on real projects and finding out about the 'dos and don'ts'?

In addition there is little discussion about course concepts and teaching approaches in the community (but see [6]). Course books are missing, too. A lot can still be learned from e.g. Schuler/Namioka [9] and more recently from Bødker et al. [2] who offer a wide collection of methods they classify by design phase and the kind of knowledge assessed[1]. But how should these methods be taught? What do people need to fully grasp the concept of PD and feel able to apply those methods? We think it is high time to start a discussion within the PD-community about how to teach useful methods to the next generation of software designers.

A COURSE ON PARTICIPATORY DESIGN
Since the year 2000 we teach a course on participatory design to students of 'Informatik'[2] and 'Digital Media' at Bremen University. The Informatik program in Bremen is known for its wide spectrum of courses in the field of socio-technical systems design. Students learn not only to build technically good systems but also to take into account the perspectives of organizations and individual users in particular roles, discover their needs and requirements, and transform those into appropriate (user-oriented) designs.

[1] What we refer to as 'method' is called a 'technique' there.

[2] Computer science or computing science in English.

Before attending our PD course students have all been exposed to the idea of user-orientation. Most often, however, they would not know how to get access to the field, communicate and cooperate with the users, and they have not much reflected on their role as designers yet. In our course we explore and discuss a spectrum of PD methods in order to empower our participants to use them in later projects.

Course Concept

The course is not compulsory but one of the core elective courses in the field of socio-technical system design. It is composed of 13 meetings, taking place once a week for approximately 3 hours each. We present the history and the general ideas of PD in Europe and the US and cover a series of methods in detail. These are all tried out in practice during the course. This makes up a mixture of lectures, independent reading, and practical exercises. As teachers we define a project that will be worked on in the exercises during most of the course, e.g., the development of an Internet travel portal.

Each class meeting dealing with a particular PD method is composed of three parts. *First*, the teacher explains the method and its application in general to provide a conceptual basis for the exercise. *Then* three 'expert students' who had to prepare the exercise in advance take over. They divide their fellow students into three smaller groups and each of them becomes responsible for one group. They explain and facilitate the upcoming exercise. Each group works in a separate room; the teacher visits the groups by turns, helps out and offers advice if necessary. *Finally*, the whole class meets again to discuss their experiences, the method, and in particular the role of the facilitator. This three-steps approach has turned out to be a quite successful way to gain methodical knowledge for user-oriented software development. At the same time, the students in charge acquire soft skills and practice the facilitation of group processes. They deepen their understanding of participatory design while learning to apply relevant methods.

Due to practical and conceptual reasons, we involve almost no 'real' users in our course activities. Practically it is not very likely to find users who are willing to engage in an educational project for three hours a week over a period of approximately eight weeks time. Furthermore it became evident that facilitating new methods while being 'among themselves' was hard enough for a lot of students, who have little other opportunities to follow similar 'hands-on' courses at university. As the course focuses on the understanding of methods and the enhancement of practical moderation skills for participatory processes, it also seemed conceptually reasonable to leave space for experimentation and to work without real users.

Course Schedule

The schedule of the course has evolved over the last eight years and is subject to minor changes each time we offer it; these changes include the adaptation or alteration of methods and the replacement of some methods by new and promising ones.

The course is composed of three introductory lectures, followed by three sessions focusing on ethnographically inspired methods, and six meetings dealing with particular methods from the field of participatory design. The last meeting serves to compare and discuss all methods. The course schedule normally maintains the order as shown in Table 1.

1-3	Introduction to PD
4-6	Ethnography & Contextual Design
7	Metaphors Game
8	CARD Game
9	Personas
10	Scenario Based Design
11-12	Paper Prototyping & Usability Tests
13	Comparison & General Discussion

Table 1: Course Schedule

Starting with ethnographic methods that allow designers to enter and explore "the users' world" and user requirements, we proceed to methods that provide opportunities for designers and users to meet on common ground and develop requirements into design ideas, ending with usability tests of system prototypes that draw users into "the designers' world" (the choice of topics was inspired by the spectrum presented by Muller et al. in [8]). The methods we introduce in class were not originally tailored to fit each other; however, they build upon each other fairly well. As students work on one common project throughout the whole semester, the artifacts produced during one exercise can be taken as input for the next one.

In the last lecture of the course we have a general discussion about participatory and user-centered design and the methods introduced. The aim is to reflect on experiences and the learned concepts and to think about ways to integrate those ideas and techniques into the traditional software development lifecycle students learn elsewhere.

A SUCCESSFUL CONCEPT?

The final discussion with the students also gives us a good impression of what they have learned during the course and what impact the study of PD had on their view of software development. Now many of them self-confidently talk about 'their favorite methods' and the importance of cooperating with users. We get the impression that our basic aims get achieved: to familiarize the students with the concept of participatory design and to introduce a set of methods that can be applied in this field. By giving them hands-on experience and improving their soft skills we want to empower the participants to use those methods confidently in the future. The actual achievement of these

goals, however, is harder to trace. But every year we are amazed how well students perform in this course. Oral grades generally exceed those granted in our other courses and usually are very good or excellent. Furthermore, preparing 'their' methods students show high motivation and their written reports are well structured, detailed and thoughtful. We think that this is due to the various forms of learning we practice: by listening, reading, asking, explaining, doing, discussing and writing.

Participatory Design and Soft Skills

Preparation and facilitation of the exercises by students has proved successful. Students gain good insights into the particular concepts and feel they could apply the methods in a 'real' project. But it also requires intensive teacher support: facilitation needs to be trained and reflected. By trying out a method among themselves and discussing the process and its results with the teacher, students can get a feeling for the method and the time the exercise will take in class. It is necessary that teachers give advice in advance and offer feedback after the group exercise so the students can learn and improve their skills.

Besides teaching PD methods, our course concept emphasizes the acquisition of general 'soft skills' on an individual level. PD means collaboration and communication in groups and demands highly developed interpersonal skills. In the exercises students structure and facilitate meetings and gain insights into the difficulties of managing group processes. Thereby they strengthen their communication skills and acquire problem solving strategies.

Mutual feedback after each exercise is essential. When the entire class meets again to talk about their experiences in group work, students are asked to comment on their own and their facilitator's skills and to think of ways to improve them. We generally ask the 'expert students' to prepare this phase in advance, but for some reason they rarely do. Probably we should ourselves moderate this last phase. This would also make it easier for the 'expert students' to take notes on the discussion which may help them to prepare their written report afterwards.

Course Evaluation

Students' feedback in the middle and at the end of the course confirms our positive assessment. Students are quite content with the concept: they like being involved in the exercises and rate the course as interesting and fun. The blocked three-hour format provides room for intensive work on the exercises, but at times it is rather strenuous for everybody. Especially during the first three meetings that mainly consist of frontal instruction, students feel overloaded with information. Next time we will introduce ethnographic approaches earlier, so hands-on activities can already be integrated into the first weeks of the course and students have more time to do field work. The rather solid first block of theoretical input could then be extended over more lectures.

We discovered that working on one sample project throughout the course not only facilitates the preparation of the various methods introduced, but is also motivating for the students. It permits to simulate working towards an end result (the prototype of a system) as a project team. But it is necessary to choose an application that allows various stages and strategies of interaction depending on users' decisions or preferences. If a 'solution' seems too obvious from the outset, the idea behind some participatory methods will not become clear to the students.

The simulation of users by students has always been discussed controversially: students often regret not to be able to work with 'real users', but they also admit that group facilitation might have become even harder and they might have been distracted from trying out and assessing the methods.

Moving into the field

Last year we taught the PD course to students who concurrently started their two-years project. The aim of that project was to develop a new eLibrary user interface that allows users to enhance their information competency while searching for relevant literature. Under the impression of the PD course the students decided to use some of the methods for their project and quite successfully did so. They invited library users and librarians as external experts for some sessions and felt inspired by the new insights they got. They reported that they were glad to have had the chance to try out the methods in class before and had felt rather self-confident when presenting them to the domain experts.

Informatik students are used to learning well-defined methods and process models. For PD, too, they are eager to learn clear-cut 'recipes' that tell them exactly when and how to use a method. Often they (mis-)perceive the succession and connection of the methods presented in the course as mandatory. In spite of our pointing out repeatedly that the methods were developed independent of each other and can each be applied independently, this message does not seem to get through to the students. Just as little do students like to hear that the applicability of some method may depend on the particular situation and even on the person who facilitates the method. While group size and time frame are accepted as relevant factors determining the choice of methods, the idea of appropriateness, personal style and preference or experience seems to interfere with their desire for clear directives.

CONCLUSION

Of course there are other ways to teach PD. Many colleagues address PD as a part of HCI or CSCW courses (e.g. [4][13]), often concentrating on the original idea and the understanding of the users' important role in the software development process or emphasizing one particular approach (e.g. ethnography [1]). Others offer project-oriented courses starting with an introduction to PD and to methods that might be used. Students have to work on a 'real world' project in some organisation or company and they have to apply participatory methods they choose themselves to explore work processes (see [2]). Concepts vary from making arrangements with one organisation to

host various projects (e.g. a school) to completely delegating the task of finding a project and a 'field' to the students.

We like the idea of asking the students to work with users doing 'real tasks' in their work environments. This way they experience the importance of getting to know work settings and user demands. They also find out that communication is not always easy between users and designers and even that there are a lot of obstacles to overcome until a project can actually start. The emphasis in such courses, we think, is set on getting the message of PD across. Our own course concept, as described above, aims first and foremost at teaching a variety of methods and enabling the participants to work in a participatory way and to select and apply PD methods in their own projects in the future.

Towards a Discussion of Methods & Curricula

There certainly is no one best way to familiarize students with the idea of participatory design and with methods they might possibly use for it. We consider it important to offer a mixture of lectures and hands-on activities in order to provide a theoretical background and to permit a live experience of the PD approach. Discussions and joint reflections are all-important, too, since the PD paradigm and ways of working are quite different from what Informatik students are taught elsewhere (at least in our study programs).

Students often feel ambivalent about the methods introduced: on the one hand they are fascinated by the approaches and the new ways of discovering a user's world. On the other hand they are skeptic about the informal nature of many methods and wonder 'whether they are used in practice at all'. How about some best practice examples? How well accepted are participatory methods in companies today? Do such methods really lead to better systems and save money in the end?

A fact that strikes them is that most standard references we ask them to read are from the early nineties (e.g. [9] [5]). Being born in the 1980s and being students of a young and fast moving discipline, they intuitively consider literature from that time quite outdated and probably irrelevant. They ask for follow-up studies and for more recent experiences with the methods we teach, but unfortunately not all of our methods are covered well in current literature.

Why do we seldom get to see papers that explicitly deal with a method in detail and discuss its various features and difficulties? Certainly many of our PD colleagues do apply and adapt PD methods in their research and development projects. We would like to ignite a debate on concrete participatory methods, their use and their effects. Let us discuss how to teach such methods to our students. This will help to establish the idea of participatory design in the Informatik and computer science curricula and empower students to use them in their professional life. Let us share our experiences - among each other and with the next generation.

ACKNOWLEDGMENTS

We would like to thank our Scandinavian colleagues for inspiring discussions at a Workshop on teaching PD in Kopenhagen, 2006, our students for their enthusiasm, and our anonymous reviewers for their critical remarks.

REFERENCES

1. Brown, B. et al. (2007). 'Seeing Ethnographically: Teaching ethnography as part of CSCW'. *ECSCW 2007*; pp. 411-430.
2. Bødker, K.; Kensing, F. & Simonsen, J. (2004), *Participatory IT Design: Designing for Business and Workplace Realities*, The MIT Press, Cambridge.
3. Bødker, S. & Iversen, O. S. (2002), 'Staging a professional participatory design practice: moving PD beyond the initial fascination of user involvement'. *NordiCHI '02*; pp. 11-18.
4. Gasen, J. B. (1996). `HCI education: past, present and future?' *SIGCHI Bull.* **28**(1); pp. 25-27.
5. Greenbaum, J. M. & Kyng, M. (1991), *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum, Hillsdale, New Jersey.
6. Kautz, K. (1996), 'User Participation and Participatory Design: Topics in Computing Education', *Human-Computer Interaction* **11**(3); pp. 267-284.
7. Muller, M. J. (2003). `Participatory design: the third space in HCI'. *In* A. Sears & J. A. Jacko (eds.), *The Human-Computer Interaction Handbook'*. Lawrence Erlbaum, Mahwah, New Jersey; pp. 1051-1068.
8. Muller, M. J.; Wildman, D. M. & White, E. A. (1993), 'Taxonomy of PD Practices: A Brief Practitioner's Guide', *Commun ACM* **36**(6); pp. 26-28.
9. Schuler, D. & Namioka, A. (eds.) (1993), *Participatory Design: Principles and Practices*, Lawrence Erlbaum Associates, Mahwah, New Jersey.
10. Wagner, E. L. & Piccoli, G. (2007), 'Moving beyond user participation to achieve successful IS design', *Commun. ACM* **50**(12); pp.51-55.
11. Wall, P. & Mosher, A. (1994), 'Representations of work: Bringing designers and users together'. *PDC'94*; pp. 87-98.
12. Weinberg, J. B. & Stephen, M. L. (2002). `Participatory design in a human-computer interaction course: teaching ethnography methods to computer scientists'. *SIGCSE '02;* pp. 237-241.

Hecht, K. Maike; Maaß, Susanne (2008): Teaching Participatory Design, In: Jesper Simonsen, Toni Robertson, David Hakken (Eds.): Participatory Design Conference 2008, Bloomington, Indiana, USA, 2008, S. 166-169.