User Centered Design – Problems and Possibilities

A summary of the 1998 PDC & CSCW workshop

Jan Gulliksen, Ann Lantz and Inger Boivie

Introduction

The approaches in User-Centered Design (UCD) vary from Participatory Design (PD) to model-based engineering. No matter the approach, UCD is not the simple, clear-cut way to successful systems development as is sometimes made out. To discuss these issues the authors arranged a one-day workshop at the Participatory Design Conference (PDC'98) in Seattle on November 14, 1998 entitled "User Centered Design – Problems and Possibilities" [1]. The purpose of this workshop was to discuss the problems encountered in UCD in practice and possible solutions, focusing on case studies in real systems development projects. Ten position papers were accepted and the workshop gathered 16 participants from 7 different countries. The position papers are available on the workshop web site [1].

This paper is a summary of the work performed prior to and during the workshop. The main areas discussed are used as headings below. The initial discussion concerned the concepts UCD and PD. These matters were discussed on our workshop web site [1] before and after the workshop. The workshop participants were asked to read the position paper in advance. The workshop did not include any paper presentations but a discussion based on each contribution and the assembled experience. This article starts with the discussion about the definitions that was the basis for the workshop.

Definitions

One of our observations in preparing this workshop is the confusion in the definitions of common concepts. What does user centered design and participatory design mean? And what does user mean? Depending on which discipline you represent, whether you are an academic or practitioner, whether your user population is well defined or not, these concepts are interpreted differently.

In order to be able to discuss the practical consequences of user centered design we needed to define these concepts.

We all agreed to use the international standard ISO/DIS 13407 (Human Centered Design Process for Interactive Systems) as a basis for our discussion on user centered design [2]. By User Centered Design we mean an approach to software and hardware design that identifies four different basic principles [2];

1. an appropriate allocation of function between user and system,
2. active involvement of users,
3. iterations of design solutions and
4. multidisciplinary design teams.

In addition to this it is important for a truly user-centered design approach to ground the design process on observations of the users' practices in the real world.

Similarly we regard Participatory Design as a specific mode of User Centered Design which implies the involvement of the users not only at the beginning and/or at the end of the process, but through all the design process. In a PD approach the users actually participate in and are in charge of the making of the design decisions.

One of the comments we received was that it is important to distinguish between bespoke (custom-built) and off-the-shelf products. PD suggests a bespoke product with the participants being the real users of the system being developed. UCD can apply to either kind of product - but in one case you might be working with the real users, while in others working with "representative" users.

What do we mean by "user"? For the purpose of the workshop and this report, we agreed that a user is a person who will use the system for performing tasks that are part of his/her job or leisure activities. People, who are only circumstantially affected by the system, are considered stakeholders, e.g. managers or support personnel.

User Centered Versus Participatory

It may seem that User Centered Design (UCD) and Participatory Design (PD) are very similar, almost equivalent terms, with PD being a subset of UCD.

However, what came to light during the workshop was that these are two overlapping sets, with an uncertain amount of overlap. Some cases were presented in the workshop which were user-centered, but which had no true sense of user participation (in the design process), whilst other projects were discussed where the user participation was in no true sense user-centered.

For example, participation by management and trade union representatives in design reviews (or even in a larger design effort) does not ensure that the designers centre their designs on the user and the users' needs.

It seemed to be generally agreed that reducing the size of the PD set that is not also user-centered was the most immediate challenge to UCD in practice.

PD, particularly in North America, is so much more than systems development. It is closely connected to the democratization process in the workplace, i.e. breaking down power structures and empowering the workers. UCD rarely involves such aspects. It is usually limited to ensuring the influence of specific users in a systems development process. It would probably be more or less impossible for most systems development projects in practice to address democracy matters and issues of power structures.

In Scandinavia PD is no longer as tightly connected to democratization or the influence of the unions as it was in the 70ties [3, 4]. This might be due to the fact that in Sweden the work environment legislation facilitates user involvement. The Swedish Work Environment Law states, among other things, that "the worker should be given the possibility to participate in the design of his /her own working conditions and in development work that concerns his/her work" [5].

Problem Areas

The basis of the discussion was the identification of several different problems with UCD. These problems were illustrated by the different case studies described in each position paper [1], and also a set of possible ways of overcoming these obstacles. The position papers covered the following aspects, amongst others:

- Communication difficulties or lack of communication between e.g. system developers and users, between management and users, between individuals in a team.
- Conflicting goals - do the goals of the different groups involved in the process conflict?
- Competence - what skills and expertise are required in a UCD project in terms of e.g. social competence, technical skills or expertise on the work activities?
- Attitudes - does UCD require certain attitudes in the organization and in individuals in order to be successful - attitudes about users and usability as well as the role of the designer?
- Project organization - what is the role of management in a project, what authority is required for making the necessary decisions?
- Work organization - in what way does IT development influence the organizational requirements?
- Work activity - is a user centered design approach appropriate for every type of work activity?
- Methods, techniques and tools - are the current methods and techniques applicable and efficient, are they easily available on a non-commercial basis? Are the tools suitable for user participation in the prototyping process?
- Requirements engineering - do UCD and requirements engineering conflict, does the demand for frozen requirements allow for the iteration called for by UCD?
- The following discussion focused mainly on four different problem areas:
- User participation,
- Project Management and Work,
- Organization,
- Communication.

User Participation

UCD should be integrated in all design processes but how this is done depends on the type of project and product.

Different approaches to UCD must be adopted if the user population is 1) known and accessible, 2) known and not accessible, or 3) unknown and therefore not accessible. Several methods exist for the first case but it is not obvious that the same methods can be used for the other two categories. Several of the questions in this session dealt with the matter of user participation when the users are not accessible or not known.

The following questions were raised:

- Where, when and how do we involve users and who should participate for the most efficient process?

- How do you work with users from the general public – who have non-professional interests in the project?
- How do we address user participation when we work in organizations with free-lancers and independent contractors?
- How do you work with UCD when the users are not available?
- How mature must the technology be, i.e. to what extent must a prototype work faultlessly, for showing it to users in e.g. distributed groups, and carrying out usability evaluations?
- How do you integrate UCD if you work with innovative IT? That is, when you work with technology that is not yet known to the public, e.g. in research and development. How do you then do UCD?

Whatever your approach, it is always better to talk to one or two users than not talking to any users at all. We all agreed that UCD should be integrated in all work. The below sections discuss how UCD could be integrated in different situations.

With or Without Users

It is always important to find out who the users are. The discussion included examples where user-centered design was performed without users, as well as examples where the project team met the users at several occasions throughout the process.

Several examples were described where the users were known but not accessible. The reasons for this could be that management prohibited contacts with the users of security reasons, e.g. military applications. Or that management considered their knowledge of the users' ways of working as better and more precise than the users' own versions. In one example management also feared that user participation would preserve old routines based on the current system. Other arguments against letting the developers get in contact with the users included heavy workload or simply because of tactical reasons.

UCD must be possible even when we cannot work with the users, even though practical user participation always should be preferred. When working without users we can focus on the users, for instance by working with scenarios, based on observations (without directly involving the users), created by the design group and make use of the prevalent psychological expertise about people. This way of working may also be useful when the users are not known and therefore not accessible.

If the system is intended for the general public, focus groups, in which representatives of the general public participate, can be used in the development process.

If the total user population is so small that we could fit all of them into the project other methods can be used. The learning process, of the users and the systems developers, that takes place in such a development project is very useful for comprehending the resulting system.

User Representatives

You can choose to work with representatives for the user group. A user representative represents a group of users or a specific category of users, e.g. disabled users, for a specific reason. One should always try to maximize the difference between the people you involve, i.e. try to cover as many different categories of users as possible. When developing for a specific organization all different types of work activities that a system is to support should be covered by the skills of the user representatives. Management and unions are not to be viewed as representatives for a user group, although they are important for the justification of a project.

Some of the representatives should be involved throughout the whole project so that they get to know the project and get committed to its purpose. However, it is well known that users after having participated in a project no longer are representatives of the typical users, e.g. in evaluations. They learn over time and get a lot more involved in the technology that is being developed. Therefore, you should involve other representatives for shorter periods for analysis and evaluations, just to overcome the risk of having the user representatives influence the project in a way that is not suitable for other users.

User Selection

Different strategies can be adopted when selecting user representatives. Selecting users on a random basis, of course, may provide you with some information on how the average user would behave. Often projects need to be able to arrive at consensus about specific solutions and in this situation it may be preferable to work with randomly selected users.

In other cases, however, you need to focus especially on the conflicting goals of the users. Your concern is not to develop systems for average users but to develop systems that support all user categories. Therefore you should cover the differences in user types. Try to maximize the differences by selecting users of, for instance different age, with different skills, various disabilities, and different computer experience.

Commitment

Having users that are committed to the development project is of course central. User commitment can be enhanced by means of, for instance, the users' participation within their normal work tasks, which give them the possibility of seeing directly how a new system could influence their work conditions. Voluntary participation of course increases the degree of commitment, as it is something the participants wish to do. Whether the users are being rewarded for their participation or not, in terms of increased salaries or benefits, travels, better positions, etc. is of course also central for user commitment.

When the users act on a non-profitable basis difficulties can occur. During the project it is important that the users really feel that they contribute to the project. For this to happen it is necessary to clearly show the participating users that all their suggestions and comments are addressed.

Working with Groups or Individuals
An example of working with representatives for a group of workers was given in which a homogenous group of about four representatives was formed in order to make them strong enough to work directly with researchers, designers and UCD facilitators.

If possible, you should work with both individual users and groups but at different occasions and with different activities in the process. Working with groups of people tends to be more conducive to creativity – people are less creative on their own. Having a group of users solving a particular problem is much more efficient than asking single users. Group work could also shift the balance of power from the development team to the user group.

Regardless of whether you work with individual users or user groups, users must be treated as equals regarding power and expertise. The goal should be to have at least be as many users as other participants in a project.

Humbleness and Respect
It is important to think about how to co-operate with users and that we have to take care of this contact. The keywords for success are humbleness and respect. The ways in which we interact with the users controls the result. Unfortunately attitudes both from the developer side and user side can be obstacles in this process.

It is what people think, say, feel and do that is important and what the context is.

When do We Meet?
In reality the users are often invited in the beginning and at the end of the development phase only. Sometimes, they are only involved in the scenario process and evaluations of a full-scale prototype. This is very unfortunate. In the phases where there is no user participation several decisions are made that affect the usability of the resulting system, that would not have been made if the users had been consulted. It is important to have user participation continuously throughout the process to preserve the usability aspects in the final product.

Working on the Field
Do we meet on neutral ground or do we let the users enter the design ground? Where do we meet?

It is important that not only the UCD facilitators but also the designers and developers should go out into the field and have direct contact with the users. Field observations might give the designers some impressions and ideas that they would not have been able to obtain otherwise. After all, users are clients that the designers and developers are supposed to deliver products to.

The designers must get to know their users but it is equally important that the users get a notion of the technological possibilities and limitations. Therefore, you should also invite the users to the designers' office: "Let the users leave their trace in the design office". This can be done, for instance, by means of modifications to a prototype or a mock-up, or by means of sketches of the system. This will be described in more detail in the section on Mutual Shared Understanding below.

Video Documentation
Video is a very useful medium for analysis and for visualizing current and future scenarios. Video is also a very efficient medium for showing developers how users actually use the products that they have designed. When a developer has seen a user perform the same error a couple of times there is no need for further convincing that the design must be changed.

In order to facilitate fruitful and informative interviews and to learn about the hierarchies of the workplace in advance, spending time at the workplace with a video camera could be very useful. The following example was described at the workshop:

"NN went down with a video camera and talked with some people that I previously had interviewed. I could tell from the video that these people that I had interviewed were the seniors and not typical user representatives. In this situation the participation had begun before we were involved, video taped the way people worked. "

In this example the video film that was shot in advance, on the floor could help the designers understand the power structures in the organization. Video can also be used for showing designers what is happening on the field with the developed systems or during the performance of work tasks, a short-cut to real observations on the field.

Finally, video can easily be used to visualize a future scenario or a prototype of a future system in its imagined environment.

Project Management and Work

The following questions were raised during this session:

- What is the role of the UCD facilitator in the process?
- How do you know when to stop your iterations?
- How do you involve the users in the actual design process?
- How do we speed up the UCD process?

The Role of the UCD Facilitator
Several roles are involved in the development work but we did only focus on the users, the designers/ developers and finally what we refer to as the UCD facilitator - who intermediates between the users and designers/developers. The role of a UCD facilitator can be as a go-between - i.e. somebody who actively steps in and resolves conflicts - or simply an observer. However, as a UCD facilitator you have shouldered the role of bringing people together, therefore you must not be partial or take somebody's side.

The role of the UCD facilitator is multi-faceted - UCD-facilitators can be matchmakers, interpreters or translators? UCD facilitators need to switch roles depending on whom they are

talking to. When communicating with users, the UCD-facilitator shoulders the role of the designer. When talking to the designer the UCD-facilitator is regarded to be a user representative. We have to be aware of the power imbalance and conflicts. It should not be taken for granted that the designers are interested in working in a user-centered manner. But, the UCD facilitator must involve the designers and create enthusiasm amongst them for the UCD process.

The role of the designer may not be clear to everyone in the project. Thus, we have to acknowledge the problems of the designers - they just want to get their job done. As of yet, we do not fully understand how the designer role changes when he/she starts to understand the users, the work/area and context.

Users and the Design Process

People and work change, therefore we will have to cover as much ground as possible and find out what the extremes are. It might be fruitful to look at old projects and where they failed in order to find out what one should avoid when adopting a true UCD approach.

Parallel design and iterative development are, of course, essential ingredients in UCD. And essential for involving the users in the actual design process. It is useful to show users mock-ups with increasing level of detail to promote a discussion. If the users come down with a totally different solution it is either bad design or a matter of bad expectation management.

Users evaluate design and redesign. The design material that we work with can tie us together. Mock-ups and user-created scenarios are useful means for involving the users in the design process. It is important to ask the users the right questions about mock-ups such as “Where would you like to put it?” and “Where would you like to plug it in?” - not “What would you like to use it for?”

Make sure that you, and others, do not value the ideas of the designers more than the ideas from the users.

User Terminology

Users are often not familiar with the abstract concepts and terms used in systems development and design. They can easily communicate in terms relating to their own work activity but may get confused or feel intimidated if asked to use computer terms or design terms. You could however help the users think in design terms that are related to their reality by using low-fidelity mock-ups.

The Cost of UCD

How do we speed up the UCD process? Should we work with quick and dirty UCD? Is UCD too expensive? Perhaps, these are not the questions we should ask - but in what ways can UCD contribute to the design process?

User centered design is often criticized for being expensive. Is this really true? We feel that with UCD we can start to think of development in another way, and offer another model. If we just add UCD to an existing model, then it becomes expensive. You have to decide on a UCD approach from the very beginning or it will just become one expense among others.

Knowing when to stop the iterations is often not really a problem, because the decision to stop iterating is in many cases not yours. It could be a decision made by factors outside of your immediate control, e.g. project managers, delivery dates or contracts.

Organization

UCD can be viewed at an individual level and on a group level, as presented above. But, practical problems also occur on an organizational level. The following questions were raised during this session:

- How do you manage integrated development, that is, simultaneous development of the organization, skill attainment and development of the work activity in conjunction with the IT-development?
- How do we create a cultural context for user participation?
- How do we take the power relations into consideration when developing IT?
- How do you avoid organizational politics in UCD?
- How do you manage user expectations?
- Whose knowledge is important when introducing new computer support? Co-construction of shared understandings?
- How does user participation affect the ways you think about design?

It is politically correct to say PD, it is a sales argument but many examples of problems were brought up during our discussion on UCD and the organizational level.

Conflict and Power

Almost any development project contains conflicts between different groups, e.g. conflicts between management and workers. Management may want to computerize the work activities but the workers might not want any change.

The following example illustrates the question about power between management and the user groups: In this particular situation, management felt that they had worked in a participatory design manner. They felt that they had already sufficiently analysed the work activities. Therefore, they did not want the designers to talk to the users or to the union people. Instead, management showed a video presentation of the new system to a group of users, and asked them to answer a brief questionnaire. Management thought that they had worked in a PD manner. The project manager agreed to this procedure and the users could do nothing to change it.

In another example the employees came up with ideas and even prototypes. Management did not care to start a new project and killed the new ideas.

Yet another aspect is that decisions are often made at the top levels of the hierarchy in an organization. People working on lower levels in the hierarchy have no say. Nevertheless,

when introducing a systems development project or a new system in an organization, the workers are those who are influenced the most. Ideally, a new system generates changes in the organization and power structures from which the workers can benefit. However, it is a well-known fact that people are extraordinarily flexible but organizations are very bad on changing.

IT affects Power structures - one of the participants brought up an example from the Swedish Health Services where an Intranet project aimed at breaking down the power structures. When the project ended, the structures returned, simply because doctors still got special services carried out for them by other staff members.

Intentionality

What is the intention of UCD? Is it technology push or user demand pull? Technology push means that the technology and its possibilities is in focus while demand pull means to start by analysing a real need or requirement from the user side - then you try to develop techniques to support it. Often it is the technology that pushes instead of the problems or needs that pull. In that case we will have to decide what our intentions are. Do we want to break down the power structures, or do we create something that reinforces them?

The discussion on power structures relates to the intention behind the systems introduction. If we want to change the power structures and flatten organizations with the purpose of democratization then PD is appropriate. If we introduce a new system well aware of the influence and needs to change the organization, but without it being the major goal, then UCD is appropriate.

We have a responsibility towards the users regarding cultural and social aspects.

Integrated Design

We have to be aware of the fact that the organization changes when you enter. When a systems development project is started the organizational change is also initiated.

The world is not unchangeable. However, in traditional software development methods we tend to regard systems development as a separate activity that is not affected by or affects anything else. This is rather natural, as most of the traditional systems engineering methods tend to narrow the view down to system construction and nothing else. UCD can help you brake out from this view. UCD can help you break out from this narrow view in that it focuses on aspects such as skill attainment and work organization in parallel with the actual systems development.

Cultural Context for UCD

The following case descriptions illustrate the problem of creating a cultural context for user participation.

A Case of Organizational Obstacles 1

The first case was a case including almost every organizational obstacle that we could ever imagine. The setting was a big organization with a very powerful hierarchy. The project was a research project on the impact of information technology and an office environment design project. The manager and the designers were willing to do PD, but they did not succeed because they had a certain perspective of their own roles and of the users. The users were not partners, but, at best, beta testers. The users were invited to workshops where they were presented ideas and prototypes. The users were supposed to state what types of modifications, they would like, but they were not able to propose other solutions and the modifications asked for were slight ones. Therefore, the users did not propose anything and did not seem to feel that they had to.

Although the designers collected data, their design vision were not very much influenced by the analysis of the data, but much more by their own perceptions and ideas. The designers took themselves as the best examples of users: “All users work in an office, and we do too, so we know the needs as well as (or perhaps even better than) the office workers involved in the design process.” There was a very strong feeling that the users did not know what they do and want. A designer stated that: “If you show them a red watch, they don't want it. But, if you then come back and show them the same watch, but green, they like it.” In fact, the designers did not create a situation in which they could enable the users to express themselves and to really explore alternative design possibilities. The designers did not really listen to the users.

Although, when a UCD facilitator was hired, she thought that she would be strongly involved in the design process and that she could use some of the nice techniques that she had learned in the project. But, because the setting was not there she was completely unable to use these techniques.

The problem is to create a cultural context that allows you to communicate. You need to tailor your language for the different audiences. But, if the management does not believe in these techniques and the “philosophical” background of PD, the efforts will fail. Either it will fail at the level of the design process, as in this case, or at the level of going from a prototype to an extended prototype or even a product. This can be exemplified in the case of the Air Traffic Control [6].

A Case of Organizational Obstacles 2

The second case was a governmental organization in which UCD had been promoted for several years. Being a governmental organization it does not pay its software developers very well. The result of this was that young software developers with less experience and skill were employed for a shorter period while acquiring experience after which they moved on to other, better paid jobs. The senior software developers stayed since their initial base was the work activity in itself. Having both knowledge of the work and software development they were indispensable and also very loyal to the organization. Moreover, the turnover of people in this organization is very high, software developers tend to work there for no more than a couple of years. The users, on the other hand, work within the organization for most of their lives. The organization contains about seven hierarchical levels and at all levels people have previously worked as

users at lower levels. Even if this was several years ago they still claim to be experts on the work on all managerial levels. The organization is constantly reorganizing. Thus, creating a cultural context for user centered design in which 25 years old male software developers are to cooperate with 60 years old female user representatives is almost an impossible mission.

Lacking user Contribution

We were also supported with an example of users that were not as contributive as you would want. The users tested a prototype but ignored the issues that the project wanted to test. The users did their own race why their tests did not supply the developers with very usable information.

Communication

Who should talk to whom and who needs to understand whom? These questions reflect the discussion on the topic communication.

Although some participants thought that a common view was impossible still the majority of the workshop participants talked about a common understanding and communication between the users, UCD facilitators and system designers.

Mutual Shared Understanding

Should the users have to learn to understand the designers? Not necessarily, but the designers must understand the user. The designer must show what the technology can be used for. Some argued that it is important for the users to understand the design constraints, others argued for the opposite. Designers should understand users, but, mainly, UCD is about helping users express their needs, ideas and expectations.

However, the users must understand that the design is a complex process involving trade-off. This may empower the users. Users cannot express their needs because of the complexity of the technology. Therefore its is our role as UCD facilitators to explain the technology to help users generate ideas about what they can use it for.

It is not clear from where you start the shared understanding. Is it confrontations and personal values or consensus and shared values that facilitate it? Basically, our shared understanding is based on the individual level - on attitudes, social skills, what am I supposed to do, and what can I expect from others?

Certain design models in architecture use themes to create a shared understanding of the important aspects in a building, such as transparency. Is there any way we could use the same technique in systems development - how would you make a systems designer think in terms of themes? What kind of themes could be used to describe, for instance, the important aspects of a user interface in a computer application?

Is it possible to participate as equals? In most projects the designers and computer people tend to take over, for instance, in workshops with users. Is it possible to participate as equals during the development process? One way of avoiding this is to start with task analysis, this puts the focus on the users and brings out the users' expertise.

Managing Expectations

Successful management of user expectations is important for keeping the users committed and contributive in the design process but also for ensuring usability in the end product. How do we bring in the users? Are the users willing to compromise?

IT people sometimes make the mistake of propagating the attitude that – "if you just give us a requirements specification, we can create a system that will suit your needs". This creates expectations that cannot be met.

In order to avoid unrealistic expectations, models, i.e. mock-ups or prototypes, can facilitate the communication between the designers and users. However, we have to make sure that we do not limit the conceptions and ideas of the users.

We value more the time of the designer than the users so we try to make the communication go from the users to the designers and not the other way around.

Group Discussions

During a one-hour session the workshop split up in three smaller groups. Each group chose which matters to discuss. The below sections describe the outcome of each group discussion separately.

Group 1

The first group discussed the following matters:

- How to introduce usability in an organization,
- User and designer characteristics,
- The role of prototypes,
- The UCD facilitator.

How to Introduce Usability in an Organization?

What do I do if I am the first usability person in a company? Should I build up a usability lab? In such case, where should I start? This was an issue that the group decided was important but that never was resolved.

User Characteristics

How do we work with the users in the design process? Should the users be involved in the aesthetic design process? People are not good at imagining things, but if provided with a prototype they immediately start doing things with it. If you ask people to tell what they would want to do in an abstract future they are very bad at imagining something else than what they have today. But if you give them prototypes to interact with and ask them leading questions they immediately start to do things with it. Concerning the aesthetics, users should not design. They do not have the ability, expe-

rience or even the interest to do so. But, given an appropriate context, the users could be made to perform magnificently with low-level mock-ups.

The Role of Prototypes

How do we make the user-centered approach live throughout the design? Very often the developers do not keep the main ideas of a designed prototype when starting the actual construction. An example was given of a design of a camera prototype. A big-sized functional digital camera was designed through a process of prototypes, mock-ups, videos, Director movies, etc. The client expected something that looked designed and they were very satisfied with the design? The main ideas of the prototype were preserved all through the production even though the industrial design changed, but the main usability aspects were preserved. One of the conclusions of this is that it is important to use different tools to produce mock-ups. One of the main goals with a prototype is to teach the developers what makes this product usable, and one way of communicating this knowledge to the developers is to design several prototypes with different tools.

Playing with Prototypes

Playing with the products is important, because when pretending to use a prototype you tend to discover several aspects of it that the user is not aware of and that are not obvious to the designer. However, physical products are much easier to treat in this way than software products. A hardware product is something that the user tends to pick up and immediately start interacting with. A software product takes a lot more imagination to deal with.

Market research, anthropology or customer/user demands should control the initial prototype, which should be very simple. For example if you are designing some sort of handheld device, you should make it as simple as possible, only as a black box. Then the user picks it up and immediately tries to figure out where to switch it on.

Interpretation of the user reactions is very important for success. It is a question of seeing what is happening, and going beyond listening to the users.

Designer Characteristics

Designers have a very special way of promoting their work. Good design is not necessarily usable design, it rather focuses on aesthetic appeal. This means that users want usable products but designers aim for design awards. This exclusive way of promoting each other's ideas obscures the general idea of creating usable products. The designers need to understand that users are clients that the designers are supposed to deliver usable products to. Perhaps you need to tame the designers from time to time. Bringing the designers to the users might give them some impressions that may change their minds about what good design is.

The context of work for the developers is also important. Having the designers meet the users in the user organization has proved to be efficient in changing the way the designers work. Solving the design problems by going next-door and asking the users directly is more efficient than trying to figure it out on your own.

The Role of the UCD Facilitator

We had quite an extensive discussion on multi-disciplinary, actors and roles. What does the UCD facilitators do? Is he/she a researcher or practitioner? In order to be a good UCD facilitator you need a multidisciplinary background. It is the need for people who know a little bit about a lot of different areas. You need knowledge on human-computer interaction, cognitive psychology, ethnographical methods and design. But you also need to know a bit about the software that is being used, you need to know its possibilities as well as its limitations in order to communicate with the developers. In addition, you need to know enough about the work domain to be able to understand the essence of the work tasks. Last, but not least important, you need to be socially competent, with good abilities to communicate with people with various backgrounds. It is important to be able to inspire confidence both with the users and the developers. Are we demanding too much of our UCD facilitator?

Evidently there is a need for more than software developers, e.g. interaction designers. Recently this has been recognized and such education are being defined at the different universities.

Shortening the Iterative Cycle

To be able to prevent the so-called "My-baby-syndrome" and to arrive at true iterative design it would be important to shorten the iterative life cycle time. Hewlett & Packard works with an iterative cycle time of 1-4 weeks, which has proved to be very successful. The iterative lifecycle consists in the beginning of lots of analysis, a little bit of design and almost no evaluation. As the cycles pass they include more and more of design and evaluation and less, but still some, analysis. Finally the evaluation and redesign takes over.

Group 2

The second group discussion started out with a discussion about iterative design. How do you plan and control such a process? How do you know when to stop?

The group also discussed different ways of involving the users.

Planning and Controlling Iterative Design

The group agreed that although the iterative process is more difficult to control, and although you do not know what the results will be when you start, the outcome is still better than the results of a non-iterative process. The normal procedure seemed to be a deadline within which as many iterations as feasible were carried out. Milestones for controlling the process were suggested.

Several ways of dividing the process into smaller steps were suggested:

- Start with a pre-study, in which you outline the contents of the next step which will result in a prototype.

- Split up the system function-wise, and construct deep prototypes.
- Carry out at least two iterations - in the first iteration you compile and test ideas from/with individual users, in the second iteration you test these collective ideas with a group of users.

Ways of Involving the Users

The group also discussed how to involve users. The ways of involving users differed, e.g.

- 5-10 workshops with 7-10 user representatives once a week early in the process.
- Workshops every second month or so with up to 15 throughout the whole development process.
- Meetings with a small group of committed users who have volunteered to participate. The meetings cover any urgent topic, e.g. Where can we put the computers? Can you share computers? The next step is to introduce the prototypes to a larger group of users.

Working with the same users throughout the process was recommended.

Half day or whole day workshops seemed to be the normal way of involving the users. In such workshops it is important to visualize the design ideas, e.g. photos, mock-ups, drawings, scenarios (written by the users). There are different advantages with different types of prototypes.

It is important that users are allowed to "leave traces" in some way - e.g. by means of modifications to mock-ups/prototypes. Video-record the workshops - the videos can be used to verify modifications and to evaluate the meetings. They can also be shown to the designers and to management.

Group 3

The third group discussion concerned power roles and co-construction of shared understanding.

Language can be one way of maintaining power. Each time we enter a new work site to start working with a new user group we have to build a common ground for communication, a so-called co-construction of a shared understanding. This you have to do every time you start a group process. By doing this you will get the basic knowledge about, e.g., the work site, the work performed and you also make a common agreement on the language and concepts used and their meaning.

Roles

All persons working in a systems development team have their roles. The UCD facilitator role is to mediate. The facilitator must also have a position as well as designers and users have. The cooperation between users and designers is dynamic and the dynamics must also exist between the UCD facilitator and the user. As facilitators we contribute to the design and for being more equal in the development team we should regard ourselves as being designers - calling our selves designers.

To avoid being trapped in traditional roles and just seeing the technology as the big interest, maybe we should focus on implementation rather than on design. This might let us focus more on the social side. Even as UCD facilitators we are a part of the design.

Users and Artifacts

Different people use artifacts in different ways and the term user is somewhat misleading since a user not only just use technology but is a person with a multiple range of other activities, interests and roles.

Designers try to design what people should do. We provide settings and develop tools and we should prescribe what should happen in this process, for instance in process automation and control room settings. Here, it is not a question of designing better control rooms, but to focus on what people do, trying to see the whole context.

We should try to design technology so that it can be used in various ways. Imagine a lorry setting. The driver sits in the lorry the whole day, driving. But, it is not the only thing he does. He might listen to the radio, talking with a friend in a cellular phone, eat or even shave his beard. If he has not reached his destination in the end of the day, he will sleep in the lorry. Our goal must be to construct other artifacts in a way that they could be used for multiple purposes. Such a goal is a new and maybe better goal to strive for.

A Purpose to Join in UCD

All involved parties have their purposes to join a cooperation process. It means work in the field that everybody wants to benefit from. You do not start to cooperate merely for its own sake.

To join you do not necessarily need an interest of the technology itself. To work with UCD or PD can have other goals. One of the participants in the group is driven by the goal to improve women's working conditions.

New technology has often not been as liberating for women as wanted. UCD can then become a tool for reaching greater goals than usable systems.

Practical Methods for Participatory Design

Comments by Anne-Laure Fayard

"I think it is important to stress the fact that for me Participatory Design means that users are involved from the beginning and during all the design process (and not only at the beginning and at the end). Users are not only observed and interviewed at the beginning or not only test the final prototypes, but they play an essential role all through the design process. Therefore, PD implies first observation, i.e. field studies using ethnographic methods, video and interviews, as well as the critical incident technique. During the field study, it is important to build a true relationship with the users, to succeed in having "empathy with them". Analysing the activity is the first point, then the design process in itself starts (although it is important, if possible, to keep going

back to the field and the data, during all the design process. It may often occur that the design process requires more analysis). The PD process consists in a series of workshops with researchers (HCI facilitators, ethnographers) and designers (and developers) and workshops with researchers, designers (and developers) and users. During the workshops with the users, you first present them what we call work scenarios which are created using the observations, videos and interviews. Scenarios should be at the same time very general and very specific. The users should evaluate these scenarios and tell if things really happen like that. These scenarios are the starting-point for brainstorming sessions where you imagine new solutions and ideas. You try to create new scenarios, design scenarios, which correspond to the first scenarios modified according to the ideas proposed in the brainstorming sessions. The ideas of the brainstorming are prototyped. It is essential that the first prototypes should be "quick and dirty", i.e. paper prototypes and video prototypes. One can also use the Wizard of Oz technique in order to give users an impression of what the system looks like. All these phases are iterative, until... and then you come back to the question: when should we stop the iterations, and I have no answer. I think it is important to have a system as flexible as possible, if this can stand as an answer..."

Conclusion

The workshop has illuminated several burning issues about UCD and PD in practice. The discussions covered the following topics:

- When and how to involve the user in a design and development process.
  The methods vary according to whether or not you have access to the users, whether they are the general public or a small, well-defined user group. No matter the circumstances, involving the users is always preferable, if not always possible.
- Practical experiences of prototyping and video recording in the analysis, design and evaluation process.
  Low-level prototypes are cheap and efficient means for having users generate new ideas about how to use new technology. It is however important to ask appropriate questions when the prototypes are evaluated. They can also be used for creating a shared understanding of the context in which the new system will be used.
  Video tapes are useful for capturing important aspects in the work context, such as pre-information about power structures, routine maneuvers and pieces of tacit knowledge, that otherwise may go unnoticed.
- Organizational obstacles to user centered design.
  These obstacles often include the unwillingness of management to involve the users. Management may feel that the users do not know what they want, or are unable to express it, or the goals of management may conflict with the wishes of the users.
  Other obstacles are attitudes within the development team, about users and usability. Or the designers' attitudes about good design, focusing on aesthetic appeal rather than usability.
- The role of the UCD facilitator in the development process.
  The UCD facilitator intermediates between the users and the designers. He/she often has to resolve conflicts and represents the users in the eyes of the designers and vice versa. A skilled UCD facilitator should have multi-disciplinary background, so that he/she can understand how humans interact with computers and other human beings, how computers and software works as well as the context of use.
- Communication problems that occur when people with varied skills and expertise communicate with one another. Respect for other people's expertise and skills is essential for bridging the communication gap.
  Although UCD is about helping the users express their needs, it is also important to acknowledge the need for users to understand the constraints of the technology and the complexity of the design process.

The workshop participants also contributed with several successful examples and a few horror stories about UCD and PD in practice.

The workshop participants concluded that there still is a strong need to market and promote UCD in practice today and we see as our mission to continue this discussion in the near future.

References

[1] *"User Centered Design – Problems and Possibilities"* Web-site for the PDC'98 -workshop http://www.nada.kth.se/cid/pdc98/workshop/

[2] International Organization of Standardization (1998) ISO/DIS 13407 (Draft international standard) *Human centred design process of interactive systems*.

[3] Ehn, P. (1988). *Work-Oriented Design of Computer Artifacts.* Stockholm: Arbetslivscentrum, Sweden.

[4 Bjerknes, G., Ehn, P. & Kyng, M. (1987). *Computers and Democracy.* ORT: Gower Publishing Company Ltd., England.

[5] *The Swedish Work Environment Law*. (Arbetsmiljölagen), § 2, part 1.

[6] Mackay, W.E., Fayard, A-L., Frobert, L. & Médini, L. (1998). *Reinventing the Familiar: Exploring an augmented reality design space for air traffic control.* In proceedings of the CHI '98, 1998 (Los Angeles, USA, April 18-23, 1998) ACM, New York, 1998, pp.558-565.

More Information

More information about the workshop, position papers from the attendants, as well as coming activities can be found on through the first author or by http://www.nada.kth.se/cid/pdc98/workshop/

Acknowledgments

The workshop participants are acknowledged for their input on the workshop and for their input on this report. The workshop participants were:

Ellen Balka (Simon Fraser University, Canada), Thomas Binder (Malmö University College, Sweden), Inger Boivie (Enator AB, Sweden), Eva Brandt (Technical University of Denmark), Jacob Buur (Danfoss A/S, Denmark), Anne-Laure Fayard, EDF-R&D Division, France), David Gilmore (IDEO Product Development, U.S.A), Jan Gulliksen (Uppsala University, Sweden), Anneli Hagdahl, (Linköping University, Sweden), Helge Kahler (University of Bonn, Germany), Helena Karasti,(University of Oulu, Finland), Ann Lantz (KTH, Sweden), Masood Masoodian (Odense University, Denmark), Markus Rittenbruch (University of Bonn, Germany), Steven Verjans (Odense Steel Shipyard Ltd., Denmark), Fredrik Winberg (KTH, Sweden)

About the Authors

Jan Gulliksen, Ph.D. in Systems Analysis performs research on user centered design in several applied projects (e.g. with the Swedish National Tax Board) at Uppsala University. Jan is also supervising the working group on user orientation at CID.

Ann Lantz, Ph.D. in Cognitive Psychology, working with user oriented system design and communities at CID and a project on knowledge exchange, communication and context in electronic networks.

Inger Boivie, MSc, works as a consultant with user-centered design and usability-related activities in a software engineering company. Inger also works part-time in the CID working group for user orientation.

Authors' Addresses

Jan Gulliksen
Dept. of HCI, Uppsala University
Lägerhyddsvägen 18
SE-752 37 Uppsala
Sweden
email: jan.gulliksen@hci.uu.se
Tel: +46-18.471 2849

Ann Lantz
CID, NADA, KTH
Lindstedtsvägen 5
SE- 100 44 Stockholm
Sweden
email: alz@nada.kth.se
Tel: +46-8-790 68 17

Inger Boivie
Enator AB
Kronoborgsgränd 1
SE-164 87 Kista
Sweden
email: inger.boivie@enator.se
Tel: +46-8-7036200