

Participatory Design in a Commercial Context - a conceptual framework

Finn Kensing

The IT University of Copenhagen
Glentevej 67, 2400 Copenhagen NV, Denmark
+45 3816 8829
E-mail: kensing@it-c.dk

ABSTRACT

The paper addresses various challenges met by IT-design practitioners, who pursue Participatory Design (PD) agendas and approaches in projects in a commercial context. These challenges vary of course, but they are different from the challenges met in PD-projects that are mainly carried out for research purposes. A conceptual framework is proposed for managing relationships and designing IT-applications in order to meet the needs of specific organizations. The framework consists of six contexts and their interrelations, highlights factors of each context that need to be taken into account, and gives practical recommendations for how to make use of the framework. The purpose is to be more concrete and operational about 'context' than it is often the case in the literature. Thus the paper tears apart some of the conflation of denotations and connotations in the term 'context' which is often used as a catch-all, for what is difficult to specify. The main argument of the paper is that the proposed framework is simple enough to be used as a cognitive tool for IT-design practitioners and rich enough to help them focus on and move between issues that have proven crucial in many "real life" projects.

Keywords

Context, participatory design, IT-design practice.

INTRODUCTION

PD-research, including my own, has usually been carried out in cooperation with various organizational members, but seldom have the participatory practices survived after we left, and seldom were the IT-application designed considered important enough for the organization in question to continue with it. The research question behind the work documented here may in retrospect be formulated like the following. What would it take to get PD "out of the lab" and applied in "real life" projects? In order to address such a question, a different research strategy involving "other voices" than those usually heard in PD-papers had to be

applied. Here I concentrate on one part of the answer, a conceptual framework, which deals with 'context', and which together with other frameworks, is part of a coherent method for participatory design in a commercial context¹. First, let us see what is meant by 'context' in this paper and in the literature.

Contexts are like frames, within which actions taken by others or oneself get their meaning. Contexts guide our behavior through the unconscious application of background knowledge that shapes our thinking and actions. We move and link issues between the center and the periphery of our awareness to make sense of our current endeavor. Contexts may help us reflect on these endeavors and explicate reasons for our behavior, if they are consciously evoked and adjusted through an appreciation of the situation at hand. Contexts are dynamically and socially (re-)produced in action. They constitute a realm of concerns competing for our attention, and actions appearing meaningful in one context may be hard to understand in another context. Drawing attention to 'the context' means articulating the importance of concerns that were thus far considered at the periphery, while it may entail that other concerns are neglected.

Various notions of 'context' have been used in *research* on information technology and have provided valid and sound accounts of this term. Some of them have also contributed to 'context' as it is presented here. E.g. Ciborra and Lanzara (1994) introduce the notion of 'formative context', "that is, the set of institutional arrangements and cognitive imageries that inform the actors' practical and reasoning routines in organizations" (p. 61). Thus, they also include the phenomenon 'institutional arrangements' in their notion, whereas in this paper 'context' is restricted to denote a cognitive tool, though also dealing with this phenomenon.

There are two primary audiences for PD-researchers: "(1) workers and other organizational members who will benefit from the design project and (2) design professionals who

In *PDC 2000 Proceedings of the Participatory Design Conference*. T. Cherkasky, J. Greenbaum, P. Mambrey, J. K. Pors (Eds.) New York, NY, USA, 28 November - 1 December 2000. CPSR, P.O. Box 717, Palo Alto, CA 94302 cpsr@cpsr.org ISBN 0-9667818-1-3

¹The paper is part of the research program MUST, carried out together with Keld Bødker and Jesper Simonsen. We have been involved in more than 15 IT-design projects in Denmark and the US. Our aim was to develop a PD-method for the purpose of IT-design in an organizational context.

may adopt participatory agendas and approaches.” (Kensing and Blomberg, 1998). Here I concentrate on the latter group. Most PD researchers concentrate on this group on the basis of projects in which they have played the role of IT-designers themselves. It is also not always made clear if the participatory approach or the IT-applications designed survived after the researchers left after their work was done. This is a valid approach though - given that the circumstances are laid out. We learn a lot from this type of research both in terms of tools and techniques and in terms of an understanding of relations between technology and the quality of work life. However, the research behind this paper has been geared towards projects in which the viability of the IT-applications designed, and the process through which they were designed, both had a high priority for the organizations in question. Another important characteristic is that the role of the researchers has been reduced. Either we were not members of the project group, or we participated, but then only together with internal or external IT-design practitioners. In all cases mentioned user representatives and managers were part of the project organization.

The inspiration for the framework proposed here came originally from Brown and Duguid (1994). They argue that designers should pay more attention to the role context plays in users' work. They provide insightful analyses and illustrative examples of what happens when designers overlook the use context of their artifacts, and of how some designers have been able to take advantage of looking beyond the artifact to resolve their problems. Their contribution, though stimulating, is also rather general and somewhat decontextualized in terms of which type of designers they address. This point has also been made by Grudin (1994) and Winograd (1994) in their commentaries on the article. In addition, the article holds few recommendations for IT-designers, in terms of what it would mean to take context seriously at a practical level.

I strive here instead to remedy both concerns, focusing not only on one type of IT-designers but also on providing means in terms of unfolding 'context', factors to attend to, tools and techniques, and references to the literature for further studies. So here 'context' is introduced in new ways. First, it is proposed as a cognitive tool, with recommendations for *IT-design practitioners* who design for specific organization's needs. They are considered the actors most likely to take responsibility for seeing to it that the various contexts are brought to bear when appropriate. By this I do not mean to ignore the possibility that IT-designers may have their own agendas, and that the question of when what is appropriate is not a neutral matter. I will return to such issues later. Second, though the conceptual framework deals with issues, some of which have also been dealt with by other authors, I have found none that deal with them all. Finally, however simple the framework may appear, the paper provides empirical evidence that the framework is rich enough to help IT-design

practitioners focus on the contextual aspects that are most likely to influence their PD-project.

I am not suggesting that a conceptual framework is enough to meet the challenges encountered when applying PD-agendas and approaches to IT-projects important to the organization in question. We have dealt with other means in terms of methodological guidelines in e.g. Kensing et al (1998a) and how to disseminate a PD-method to IT-design practitioners in Kensing (1999) and Bødker et al (2000). Due to space constraints, I can only offer examples of tools and techniques that may help IT-designers be more aware of context. Dissemination activities are not dealt with here.

The structure of the paper is as follows. The next section gives a brief overview of various ways in which 'context' is used in the literature and explains how the conceptual framework presented in this paper covers and integrates these different notions of 'context'. The section on research approach and scope narrows down the type of IT-designers addressed in the paper, and explains the role they are expected to play. It further gives a brief account of how the framework has been developed. Then follows the main section of the paper that presents a new conceptual framework, which unfolds the various meanings and shades of 'context' as background knowledge for design. It further provides practical recommendations for IT-design practitioners, who pursue PD-agendas and approaches in their projects. Before the conclusion, I exemplify some of the events that have motivated the outlined framework.

'CONTEXT' AS USED IN THE LITERATURE

There seems to be no common definition of 'context' in the literature relevant to our area. Sometimes the term is used to cover aspects of the complexity in the technological component, other times in the interactions between agents. The purpose here is to systematize and operationalize the concept. Therefore I give some examples from the literature and relate them to the version of 'context' proposed here, which will be covered in more depth in the next section. Some authors propose methods or tools and techniques that support an awareness of users' work practices in activities like design, development, and implementation (see e.g. Blomberg et al (1996), Hughes et al (1993), Kensing et al (1998a), Beyer and Holtzblatt (1998)). This is referred to here as paying attention to *the work practice context*, which seems to be the dominant interpretation of 'context', as reflected by Brown and Duguid's discussion. An awareness of this context has a long history within PD and its roots (see e.g. Kyng and Mathiassen (1982), Greenbaum and Kyng (1991)). Even standard IS methods now include an awareness of work practices through the introduction of e.g. use cases (Jacobsen et al, 1992), and many researchers within HCI and CSCW have for a while now demonstrated ways of taking the work practice context seriously in design (see e.g. Blomberg et al (1993), Hughes et al (1993)).

The above is often referred to as *the use context* in the literature. Here however, it is used as a meta concept, that in addition to the work practice context also includes *the strategy context*. How to align IT-design projects with business strategies is usually ignored in the PD-literature, and treated solely from a management perspective in e.g. Lederer and Sethi (1991). However, when a PD-approach is applied in a project important to the organization in question, one of the challenges met by the IT-designers is to interpret such business strategies, which may or may not be in writing. IT-designers have to find out which of the strategies their current project contributes to and which falls outside the scope of the project. Sometimes they even have to maneuver in a highly political terrain, where the interests of users or other actors are raised and found in conflict with management's strategies.

Grudin (1991) argues for an appreciation of the context in which systems development takes place, and uses this as a starting point for a discussion of the appropriateness and applicability of e.g. participatory design strategies. He distinguishes between the points of time in a project when IT-designers and users meet, which leads to three types of projects: Contract development, product development, and in-house development. The point is well taken and has been used here to focus on contract development and in-house development, in which both the future users and the IT-designers are identified when a project starts. Grudin's distinction draws attention to one aspect in what here is referred to as *the project context*, and which is further divided into *the design context* and *the implementation context*. This division reflects two distinct time spans, which was found in most of the IT-projects studied. They were organized into a design part and an implementation part with a period of decision making in between. This was partly due to an increased use of standard systems and outsourcing of IT-services. What is more important to the focus of this paper though, is the division of labor that was associated with this type of project organization. More often than not, we found that the IT-professionals who took care of the first part, and who we therefore call the IT-designers, played either a minor role in the technical and organizational implementation of their designs - or none at all. Often the technical part was outsourced to a software house and the users had to take care of the organizational implementation themselves. Most (PD-) literature advocates that there should be no such division of labor between those designing and those implementing IT-applications. In principle, we agree with this, since a division leads to a loss of knowledge and all sorts of practical problems. But it seems that the IT-industry and user organizations have other reasons for this type of project organization. Thus, here I address how IT-designers may cope with this type of project organization and the division of labor often associated with it. This may imply that the framework is less - or not - applicable in other forms of project organization.

Standard IS literature is concerned with how to handle the technical parts of a design. For instance, object oriented methods (Jacobsen, 1992) provide guidelines for designing the overall systems architecture, the functionality, and the interfaces. In the framework presented in this paper, such issues are dealt with in terms of *the systems context*.

Finally, Blomberg et al. (1997) describe a project in which they developed a case-based prototype. They discuss the difficulties involved in trying to fit this prototype with the platform being developed by the product development divisions and with the user's technological infrastructure. Such issues are dealt with here as *the platform context*, which together with the systems context is part of *the technical context*.

SCOPE AND RESEARCH APPROACH

As indicated above I focus on IT-designers, who conduct PD-projects for a specific organization's needs for new IT-applications. Their role is to engage organizational members in developing and anchoring an understanding of the needs and opportunities of the organization in question, as well as in designing and anchoring one or more coherent visions for change (Kensing et al, 1998a). Such visions address IT-systems, organizational change, and the qualifications required by users. They participate in evaluating the visions seen from the perspectives of various groupings in the organization. Finally, they produce a plan for the technical and organizational implementation of the selected vision. Though narrowing down the target group in this way, we are still not dealing with a homogenous entity. This group is made up of people with long or short and diverse professional training and experience. However it is fair to look upon this group of IT-designers as a community of practice (Lave and Wenger, 1991). They do share - to a large extent - attitudes about user participation and a common understanding of what their job is all about, the role of technology, how projects are carried out, and which techniques to apply. Of course their practice has evolved over the years, and there have been disputes over for example appropriate techniques and roles in design. But this is also the case for e.g. midwives, tailors, and others in Lave and Wenger's (1991) examples.

The research approach applied in order to develop the conceptual framework has been a combination of action research (Greenwood and Levin, 1998) and a grounded theory approach (Strauss and Corbin, 1990). I have played all three of the below mentioned roles in PD-projects for the purpose of collecting information and experiences:

- IT-designer: In IT-projects with users and managers (Bødker and Kensing, 1994), (Simonsen and Kensing, 1997).
- IT-designer: In IT-projects with users, managers, and internal IT-design practitioners - sharing responsibilities with the latter (Kensing et al, 1998b).
- Coach: Performing participatory analyses of the work practices of IT-design practitioners, as a basis for dis-

seminating a PD -method (Kensing, 1999), (Bødker et al, 2000).

The grounded theory approach has been applied to observation and field notes, video recordings, transcripts of interviews, and other material collected through the above activities. The emerging categories were compared to relevant literature, which was also consulted for inspiration. The result is described and argued for in the next section. Then parallel to Stauss and Corbin (1990), I use one of the projects analyzed to illustrate some of the events that motivated the formulation of the framework.

CONTEXT - AN EMPIRICALLY GROUNDED, ACTION-ORIENTED CONCEPTUAL FRAMEWORK

The intention here is to propose a *conceptual framework*, not a theory or a definition. The framework takes form of six contexts, and for each of these I address some of the key factors that IT-designers need to be aware of and some of the tools and techniques, which support this. I am not able to provide all inclusive lists of factors that may influence any IT-design project or of tools and techniques. The point is rather that the framework assists IT-designers in producing workable local theories of how to navigate in an always evolving territory. The framework is *action-oriented* in the sense that it supports reflection and taking action in IT-design projects. Finally, the framework is *empirically grounded* since it addresses issues found important through a grounded theory approach to the analysis of IT-design projects in commercial settings.

Below I address each of the six contexts, one at a time. However in any given project their relations to each other are of course important to be aware of. This is illustrated by pointing (according to their number) towards other relevant contexts that need to be taken into account when each of them individually is at the center of our attention. So the framework supports separation of concerns as well as integration.

Project Context

Project context deals with the fact that IT-projects need the cooperation from various actors, who are assembled for this specific occasion and who may be internal or external to the organization in question. For reasons described above, IT-designers may have different roles and responsibilities in different periods of an IT-project, which is why I treat *the design context* and *the implementation context* separately below. They are each a temporary endeavor and involve a network of various actors: IT-designers, managers, users, programmers, external partners, and internal support groups. They are all accountable also to each of their permanent contexts. Newman (1994) refers to such complex institutional structures as "the organizational embedding of design practice", to which she ascribes several complications.

1. Design context: An IT-design project needs to be seen in the context of other organizational processes in which it is embedded.

The design context is addressed when factors like the fol-

lowing are dealt with: the ambitions of the project and the resources invested to meet them; project goals and their relations to other initiatives; interest groups and how they relate to the project goals; staffing; project organization and allocation of responsibilities; critical success factors; expectation management; and anchoring the visions.

IT-design is a collective activity, embedded in a larger network of actors. It is crucial for IT-designers to be aware of the contexts in which these actors are also accountable. IT-designers have to satisfy the legitimate needs seen from the aspect of these other actors. Therefore, IT-designers produce intermediate and final products in which factors like the above are dealt with. And this in ways that are relevant, understandable, and oriented towards the needs of actors, who contribute to, or will be affected by the overall change. While some of those products are results of and contribute directly to the ongoing design process, others are to be perceived as "products of accountability", to paraphrase Suchman (1994).

Some IT-design projects are performed as a relatively closed process in order to prevent new requirements from constantly coming up. Then intermediate design products are considered fixed and are handed off for others to implement with little or no further communication involved before the delivery. While this might be needed for some of the factors to ensure the project's driving force, we also need to open up the process for other actors to contribute and for relevant groups to respond to each other's intermediate products.

Kyng and Mathiassen (1997) address conditions under which projects are carried out. They do not spell out the differences, but simply state that American and European companies constitute different contexts, which again differ from that of university research. Their point is that these contexts shape the mutual (mis-)understandings of the actors involved in design.

This calls for mechanisms that allow for the different actors' understandings to be evoked, discussed, and prioritized. Therefore in addition to designing applications, the role of the IT-designers addressed here is to design a PD-project as a process. This involves finding ways through which the various actors can contribute with their competencies as directly as possible and where cooperation can be built upon continuity and upon a respect for contexts as perceived by others. PD is though not only concerned with the competence, but also with how the interests of the parties are taken into account. For this IT-designers need technical skills as well as skills to maneuver in a highly political terrain. So part of seeing project design as a process, is also finding practical answers to questions like the following: Whose interests need to be dealt with? How can conflicts be diminished? And when they surface, how should they be dealt with?

This is very different from a PD-project in which a group of researchers and users design IT-systems to meet the needs

of the latter. Political issues may also be relevant in such projects of course, but they are mainly portrayed in ways where harmony exists inside the project. Instead in the type of IT-projects dealt with in this paper, that is not to be expected, and as stated above, we see the IT-designers as being the actors most likely to take responsibility for the political matters being dealt with appropriately.

Practical recommendations: IT-designers need to deal with this context up front in terms of how the project will be organized and planned as a process. It is just as important however to be aware of the context in which the IT-design project takes place, e.g. in situations like routine check and when the project runs into problems or a crisis.

Project establishment (Kensing et al, 1998a) has proven to be helpful in laying the foundation for direct cooperation between actors, who are each responsible to their permanent organizational context. It addresses most of the factors mentioned above. The purpose is to negotiate and make clear the conditions of the project in question; to agree on its overall aims and its organization among project participants; and finally for the design team to make the first project plan.

An important aspect of the project plan is how quality control will be carried out. Quality control is here understood in a broad sense covering a vision's technical qualities (5) and (6) as well as its use qualities (3) and (4). In order to manage the dilemma between conducting a closed and an open design process, we need to produce intermediate products for other parties to relate to. These might be sketches, scenarios, prototypes, project plans, and drafts of final design reports. Other actors write programs, deliver hardware and basic software, write proposals for training, and carry out other implementation activities. These should be produced, evaluated, and negotiated in ways that open up a dialogue that allow for the competencies as well as the interests of the various actors to be heard. Such ways include for example IT-designers conducting inquiries into the contexts through which the actors generate meaning. Other techniques are reviews and hearings, where the project group and the steering committee present their current results and plans and gather the opinions of actors, who do not participate directly in the project.

Sometimes this can be done in a way that ensures an agreeable compromise from which they all can proceed. Sometimes this will not be possible. Therefore a proper forum for handling potential conflicts needs to be found. E.g. different interests may be represented in a steering committee, or conflicts may be brought to permanent organizational bodies. When conducting a real life PD-project we cannot choose sides - not for the weaker part anyway. We may however insist on producing materials that allow for the various parties to evaluate how their interests are effected by the visions proposed, and then leave it to them to find a solution that is viable in that organization. This means that it is part of our role to help establish and maintain a sustainable basis for the

overall change, which also includes an awareness of what the users are prepared to deal with within the scope of the actual project.

2. Implementation context: The visions of an IT-design project need to be seen in the context of its subsequent implementation.

It is important to recall that the conceptual framework presented here is intended for the IT-designers' use during an IT-design project. This means that the below factors do not include every aspect that has to be taken into account by those in charge of its subsequent implementation.

Relevant factors for the success of an IT-implementation project, which could be taken into account during an IT-design project, include: the resources that the organization is willing to spend to make the visions become a reality; the degree to which the organization as a whole is ready for the proposed changes, or what it would take for it to become ready; a plan for the implementation including an ongoing concern for relations between the visions and what is being implemented; the relationship (content, money, attention) between this project and other ongoing initiatives in the organization.

The point is to demonstrate the distance in terms of costs, efforts, and time between a vision and its implementation at a time where it is relatively cheap to make adjustments. This may even provide the evidence that certain visions will not make it in the organization in question and that further funding for these should be stopped. The latter is also considered as a valuable outcome of an IT-design project.

Näslund (1997) reflects on a project, in which he found that: "The actors all held contextual views on the application to be developed, but their views were different because they viewed the computer application in different contexts [...each of which] made it possible for the actors to apply their expertise [...but] the different views often created obstacles for communication between groups" (ibid., p. 187). For instance, he found that the developers, who were in charge of the technical implementation of a specified design, were eager to reduce the design space in order to ease coordination among several developers and to facilitate discarding new and innovative ideas which otherwise would prolong the project.

Practical recommendations: It is not considered among the core competencies of the IT-designers addressed here to be able to fully deal with this context and number (6) below. However, they need to attend to the implementation context during the IT-design project. This is especially important if they, or others who are able to account for the rationale of the overall design, are not expected to take part in its implementation. They may do so by including in the plan for the implementation project that they need to be consulted and participate in evaluations of the consequences, especially if proposed changes alter the content of or the relations between

components of their original design. The point is to ensure a coherent design, not to advocate always sticking to "the master plan," or claiming that IT-designers should have the same rights as some architects, who have gained certain privileges over their designs.

Use Context

Use context is about relating a coherent design to the work practices of the intended users and to the corporate strategies of the organization in question. As stated above, we consider a coherent design to be one that addresses and interrelates IT-systems, organizational change, and the qualifications required by users.

3. Work practice context: An IT-system, or a suite of systems, constituting the technical part of a coherent design, needs to be seen in the context of the *work practices* it is intended to become a part of.

For IT-systems to be really useful we need to understand the context in which our designs are going to be used. Important factors to attend to include: purposes and content of the business as such; users' goals in relation to their jobs; relations between users and with managers, customers, or clients; artifacts used for coordination and for the work proper; content and structure of direct and mediated communication; temporal aspects; sources of uncertainty; reoccurring breakdowns; relations between the proposed IT-systems, work organizations and users skills; and other aspects considered important by the users in question.

According to Brown and Duguid (1994), the overall challenge in design is related to its focus, which should not be the artifact. Those who still have this viewpoint might have difficulties in reading their article. Nor should the focus of design be on the "user", on the "customer", or on the "worker" - those subscribing to one of these standpoints seem to be their intended audience. They recommend that the focus of design be instead on the community of practice. They characterize a community of practice along the lines of Lave and Wenger (1991), as being "the social context of an artifact's use" and as "the level of the social world at which practice is common, coordinated, and reproduced, at which significance is created, and consequently, in which the border is socially recognized and generic conventions are developed and shared" (Brown and Duguid, 1994, p. 19 and 20). Their point is not unfamiliar to most current PD-researchers and practitioners, as also is pointed out by Bennett (1994) in his commentary to Brown and Duguid (1994).

Blomberg et al (1997) are among the key proponents of PD in the US. But they see the conditions set by U.S. companies as limiting the changes toward more work-oriented design practices, which they aim at. They find that the chief condition is the concern to cut labor costs both in design and in use. Thus they address a potentially problematic challenge for IT-designers that has to do with the relation between (3) and (4):

Striving to meet the needs that are found meaningful within the context of users' work practices, but that might not be in line with corporate strategies. And I may add that the opposite situation is just as problematic.

Even without such conflicts, PD is a new challenge for many IT-designers. They are not used to relating their designs that closely to users' work practice. E.g. they rely solely on interviews with users (sometimes though only with their managers) in their analysis of current work practices.

Some argue that focusing on current work practices will lead to conservative designs (Hammer and Champy, 1993). But as also pointed out in e.g. Brown and Duguid (1994) and Kyng and Mathiassen (1997), ignoring current work practices leads instead to flawed designs. The contributions to Kyng and Mathiassen (1997) are based on the assumption that not paying attention to the context in which the artifact is used is an important reason for many failures and breakdowns in contemporary systems development (*ibid.*, p.xi). And I may add that the main reason behind that assumption is that in most cases people have good reasons for performing the jobs in the manner that they do. What may seem odd to an outsider - like an IT-designer - often does so because it has taken many years of education and experience to learn to do the job. I consider it a virtue of PD-designers to demonstrate respect for 'organizational wisdom' and at the same time contribute with their own area of knowledge and organize mutual learning activities that may contest such wisdom. In addition innovative ideas often come from the deep understanding of intrinsic relations between factors mentioned above and that PD tools and techniques help to establish. Finally, it usually pays to be able to account for a design vision not only in terms of technological options but also in terms of current obstacles - not to mention being able to link the two.

Practical recommendations: I will be brief here, since this context is covered widely in the PD-literature. It is usually considered a core competence of PD-designers to be able to develop a deep understanding of users' work practice, and have that contribute to their designs. Ethnographers have especially been helpful in providing guidelines for understanding current work practices, see e.g. Blomberg et al (1993) and Hughes et al (1993). Tools and techniques have been proposed that relate to various combinations of the factors mentioned above, see e.g. Greenbaum and Kyng (1991). Such guidelines have been integrated into coherent PD-methods, where they are combined with interventions, like prototypes and scenarios of envisioned work practices, into overall iterative approaches, see Kensing et al (1998a) and Beyer and Holtzblatt, (1998). We are not ethnographers, but we have found that it has certainly made a difference to include ethnographic techniques in our design practice (Kensing, 1998), (Kensing et al, 1998b), (Simonsen and Kensing, 1997), and those techniques applicable specifically to IT-design practitioners (Kensing, 1999), (Bødker et al, 2000).

4. Strategy context: A coherent design needs to be seen in the context of *corporate strategies*, and these may or may not be in line with the *interests* of various individuals or groups of users in the actual project.

We do not expect IT-designers to conduct strategic analysis as part of the type of IT-design projects we address here, rather part of the job is to figure out which of these strategies the current project relates to and how. Some organizations do not have *written* corporate strategies, but they will play a role at some point or another when PD is applied in an important IT-project. Therefore IT-designers sometimes will have to assist management in making their strategies explicit, in order to be able to relate them to the current project, see e.g. Simonsen (1999), Lederer and Sethi (1991). We have seldom found that other groups of actors have interests formulated at this level, rather part of taking care of (3) is to assist in bringing these up.

So addressing this context involves factors like: management's general interests and their relations to other actors' goals and to the current project; how these may evolve during the course of the project; scope of the intended change; relations between the organization and (changes in) the environment; and the products and services offered by the organization.

Corporate strategies express the management's visions for change. Some of these visions are related to new or improved products and services, others to ways in which users' work is organized, the IT-support, etc. Thus we need to account for the degree to which the proposed design contributes to such visions. However, often managers and others see a new IT-project as an opportunity to promote their general ideas. Therefore, to ensure a clear mandate, we need to also make it clear which strategies or general ideas the current project is *not* intended to support.

Individuals or groups of users might question and take actions against corporate strategies because they are not in line with their own interests. Ignoring to take proper care of such organizational conflicts has led to that major parts of IT-systems are never put to use. Other times users had to invent work-arounds because the systems did not correspond with their work practice, which after all management did not have the insight or power to change.

Practical recommendations: How to deal with this context is one instance where our focus on IT-design practitioners is different from PD-projects where researchers team up with workers. The IT-designers addressed here need to interview managers and analyze strategic documents to obtain an understanding of management's strategies. They may apply e.g. a SSM-approach (Checkland and Holwell, 1998) to arrive at a workable 'system definition' or an Information Ecology-approach (Davenport, 1997) focusing on an understanding of relations to the environment. As already mentioned above, Simonsen (1999) and Lederer and Sethi (1991) pro-

vide thoughts of how to link ideas at the strategic level to an IT-project level.

Like it was the case for (1) above, this context has to be addressed up front, but it also needs to be kept in mind by the IT-designers during the course of the project. First because strategies may change, and second because part of management's criteria for evaluation of intermediate and final results will be based on their strategies.

In (1) above, we have already recommended how to deal with potential conflicts, which also covers those that might materialize between management's strategies and the interests of other actors.

Technical Context

Technical context refers to the fact that an IT-system interacts with other systems and with the platform.

5. Systems context: An IT-system, which is part of a suggested design, needs to be seen in the context of *existing and planned systems*, as well as in the context of *other systems that are part of the same design*.

This context is addressed when overall systems-architecture is dealt with, i.e. when we are concerned with factors like: criteria for deciding which parts of the overall design will be taken care of by which IT-systems; how they each by themselves - and in their relation to each other - constitute a whole; modalities of interaction; and when technical qualities like design for flexibility and maintainability are addressed.

It is the job of IT-designers to give form and function to an IT-system in such a way that it respects the form and function of other systems with which it will inter-operate, otherwise the other systems have to be re-designed. In other design disciplines, like in architecture, an arch serves a specific function, it connects to other building elements in special ways, and maybe its shape also produces an aesthetic effect. The architect might try to alter these other building elements, if they do not fit the arch he or she envisions. Then negotiations are called for if he or she is not responsible for these other building elements.

Likewise, IT-designers group elements of an overall design into various components, some of which are IT-systems, others are specifications of envisioned work practices. The rationale of each element and their relations are accounted for to check consistency and to ease development and maintenance. Changes might be called for during these subsequent processes: New user requirements turn up, some components are substituted for others due to economic or performance considerations, etc. Some changes, which at first might seem technically intrinsic, might have an effect on users' work practices or on the duration of the project. Therefore we also need to bring into the center of our attention factors related to the work practice context and the project contexts (see (3), (1), (2)), even when purely technical changes are decided upon, implemented, or reviewed.

Suchman (1997) points out that "professional design needs to be understood not as an end point but as a starting point, or a platform, for the ongoing processes of "lay" design or design-in-use that are both inevitable and necessary for an effective working environment".

While this is a fair claim, Naur (1985) points out that it will never be possible to fully account for an IT-system in a way understandable for people outside the team that built the system. Naur (1985) argues further, that when the last programmer has left the team, the newcomers might as well start all over again. The two positions constitute a challenge due to the project organization and the division of labor found in most projects (see above). The IT-designers might not be accessible when inevitable changes are called for. Who's in charge?

Practical recommendations: This context is the primary concern of current IT-design methods. E.g. object-oriented analysis and design (Jacobsen et al , 1992) suggests the use of encapsulation of related properties into components with clear-cut relations to other components. This technique also supports division of labor (1) and (2) - though Naur's point challenges this when the above mentioned division of labor is the case.

In addition other actors, representing interests or areas of knowledge not directly involved in the design, may be included through reviews or hearings. So a way to demonstrate an awareness of the systems context may be to bring in other voices and their respective perspectives and give them the opportunity to check consistency in design proposals.

6. Platform context: An IT-system, or a suite of systems as a whole, needs to be seen in the context of the *hardware and basic software*, on top of which they are to run.

King (1994) reminds us that technologies develop along paths carved by prior technological endeavors, some of which are hardware and basic software. They lay constraints on IT-design while at the same time they open up a solution space. Thus we need to be able to understand the platform context in order to take advantage of it in our designs. This includes being able to deal with factors like: speed, bandwidth, and storage capacity and the entailed constraints/openings in relation to the affordances of the IT-systems being designed; relations between the IT-systems and the organization's IT-strategy; and the capabilities of the platform's basic applications and development tools.

Due to the division of labor and associated specialization described above, the type of IT-designers addressed in this paper might not have the necessary skills to deal with all aspects of this context by themselves. They never or seldom get the chance to write programs that interact directly with the hardware and basic software. So they need to cooperate with people who have practical experience with the hardware and the basic software on which the designs will eventually

run. Thus IT-designers may have to rely on hard-core technicians as consultants, thus adding to the transaction costs of the project. Though for genuine communication to take place, IT-designers still need "a sound mastery of technical fundamentals", as also pointed out by Kapor (1991).

Hardware or basic software being developed or substituted concurrently with the design of new IT-systems is another factor that adds to the complexity as illustrated by Blomberg et al (1997). They reflect on a research project, which was carried out in an industrial research setting and that involved two product divisions and a potential user site. A major part of their discussion concentrates on the difficulties they encountered when striving to match up their case-based prototype with the platforms under development by two product development divisions and with the technological infrastructure of the potential user organization. The platform developed by the one product division was unstable and in flux, thus not suited for the researchers' prototyping approach. The development team from the other division turned out to be unable to fit the necessary customization into their deadlines, though this platform was more stable and had the potential of being integrated into the users' environment.

Practical recommendations: As mentioned above, I do not consider it to be part of the core competence of the IT-designers addressed here to be experts on this context. It is becoming still more complex as a new IT-system might require integration with old mainframe systems, client/server, and web-platforms. In most organizations, the three platforms are handled by different groups of people either inside or outside the organization. Therefore when platform issues need to be at the center of their awareness, IT-designers might turn their attention towards the project contexts described in (1) and (2) above.

AN EXAMPLE

In accordance with one of the criteria of Grounded Theory (Strauss and Corbin, 1990), I provide examples of some of the events that pointed to the formulation of the six categories and to some of their relations. The examples have been taken from one of the analyzed projects (see Kensing et al (1998b) for details). Again, the numbers in soft parentheses refer to the six contexts highlighted by the framework.

The aim of the project was to design computer support for the production of radio programs. The project was divided into an IT-design project, on the basis of which an overall design was decided upon, and an IT-implementation project. The proposed design comprised of a suite of 14 IT-systems, some of which were standard systems, while others were to be developed for the station's specific needs. The first customized system to be developed - the Event Calendar - should support the journalists' research of stories and ease coordination among editorial units. After the design report was accepted but before the implementation began, the deputy editor of the station decided that the Event Calendar should be "more ambitious". He wanted the journalists to report

on current broadcasting plans through the Event Calendar in order to support the editorial board's needs for internal coordination and for its vertical coordination with the editorial units. Both needs were included in the corporate strategy, but they were not to be addressed by this project. Thus, the participants were forced to be aware of a part of the corporate strategy (4), which the project was not intended to support and which they therefore considered at the periphery.

The deputy editor also wanted the system to support other journalists than initially planned for. Though it had been outside the design team's charter to support these other needs, the programmers, to whom the development and technical implementation of the Event Calendar was outsourced, incorporated these needs in their programs without further questioning.

Including the needs of an additional group of journalists turned out not to be a problem. On the other hand, to include the editorial board's needs required journalists to type in data, which gave them extra work without any benefits. Without requiring extra work of the journalists - these data would have been available from another of the systems proposed by the design team (5). However that system was postponed, since a market analysis showed that such a system would soon be available as a standard product. The manager did not want to wait.

Some of the journalists (quietly) resisted typing in the extra data, since they did not have the time for it (3). Therefore the manager had two staff members to collect the information and type it into the Event Calendar in order to fulfill the editorial board's coordination needs. However, one of the staff members reported that "I only use the Event Calendar when I have the time." (3) Thus, the system contained data that was only somewhat reliable (5).

The changes began as a strategic move (4) after the IT-design project based upon a concern for improved use-quality for new groups of users (3). The changes were thought of as only related to the systems context, adding extra functionality and moving a function from one system to another (5). But the system imposed negative effects on its primary users (3) and yielded unreliable data (5) for the editorial board.

A problem related to the interplay between the platform context (6) and the work practice context (3) was found during the final tests of the Event Calendar. Programmed in Visual Basic, it required more memory than the IT-designers had planned for. Thus, it was not possible to have the program running in the background while using other applications. In the end the Event Calendar was hardly ever used. Some journalists though found that the original functionality of the Event Calendar was helpful and they decided to go through the trouble of applying for extra memory and typing in the extra data.

The relations between the IT department and the user organization were challenged, due to an improper division of responsibilities (2). The project charter for the IT-implementation project did not include a decision of who should be in charge of the organizational implementation and how it should be carried out. During the negotiations involved in the acceptance of the design team's final report, it was decided that the organizational implementation should be left to the users. The IT-designers' warnings fell on deaf ears. They knew from past experience and from literature (see e.g. Bullen and Bennett (1990) and Orlikowski (1992) the amount of work and the skills it takes to make IT-systems work in an organization.

The management of resources among the IT-department's ongoing projects also turned out to be a problem. The IT-designers were constantly overbooked. For example, the project manager continued into the IT-implementation project but was also allocated to another big project during that time (2). Thus in practice, she had limited time and responsibility for delivery management during purchase, development, and implementation. Also, the person who knew the most about the client/server platform, did not have the time to consult the IT-department's various projects. He would have been able to calculate the needed amount of memory in order for the PC's to run the suite of proposed IT-systems. His colleagues however knew the size of his workload, so they seldom asked for his advice (1). Finally, the IT-manager was more eager to start new projects, to demonstrate the productivity of his department, than he was maintaining the necessary resources to carry them out (1) and (2).

CONCLUSION

The paper suggests a new conceptual framework that takes a practical approach to what it means to deal with 'context'. Since it has been developed and tested as part of a coherent PD-method, it is difficult to distinguish if the successful dissemination of the method (Kensing, 1999) and Bødker et al (2000) should be attributed to the framework or other elements of the method. However, the paper provides empirical evidence that an improved understanding of the contexts to which IT-design practitioners need to relate, is crucial for PD-projects to make it out of lab-like settings. Others have also addressed parts of the framework, but none have dealt with all of its aspects. The framework is a simple, yet rich cognitive tool for planning and conducting design endeavors. It helps us reflect on difficult situations that empirical studies show that we are likely to encounter in PD-projects in commercial settings. Combining and linking the elements of the framework allow us to build local theories that support and account for meaningful actions.

ACKNOWLEDGEMENTS

A less elaborated version of the framework was developed during the EU-funded DESARTE-project (Wagner et al, 1997). Thanks for valuable comments on earlier drafts of this paper go to Mark Ackerman, Niels Erik Andersen, Jeanette Blomberg, Keld Bødker, Sisse Finken, Jonathan Grudin, Julian Orr, Lone Hoffmann Pedersen, Kjeld Schmidt, Jesper Simonsen, Lucy Suchman, and Randy Trigg. Thanks also to Kristen Simonsen for improving my English.

REFERENCES

- Bennett, J.L. Moving from Concepts to Practice. In Brown and Duguid (1994).
- Beyer, H. and K. Holtzblatt: *Contextual Design*. Morgan Kaufmann, 1998.
- Blomberg, J., J. Giacomi, A. Mosher, and P. Swendon-Wall. Ethnographic field methods and their relations to design, in Schuler, D. and A. Namioka (Eds.). *Participatory Design: Perspectives on System Design*, Lawrence Erlbaum, Hillsdale, NJ, 1993, 123-154.
- Blomberg, J., L. Suchman, and R. Trigg. Reflections on a Work-Oriented Design Project. *Human-Computer Interaction 11*, 3, 1996, 237-265.
- Blomberg, J., L. Suchman, and R. Trigg. Back to Work: Renewing old agendas for change. In Kyng and Mathiassen (1997), 267-287.
- Brown, J. S. and P. Duguid. Borderline Issues: Social and Material Aspects of Design. *Human-Computer Interaction 9*, 1, 1994, 3-36.
- Bullen, C. V. and J. L. Bennett. Learning from User Experience with Groupware, *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, 1990, 291-302.
- Bødker, K. and F. Kensing. Design in an Organizational Context - an Experiment. *Scandinavian Journal of Information Systems 6*, 1. 1994, 47-68.
- Bødker, K., F. Kensing, and J. Simonsen. Changing Work-practices in Design. To appear in Ditrich, Y. et al (Eds) *Social Thinking - Social Practice* (working title). MIT Press, Cambridge, MA, 2000.
- Checkland, P. and S. Holwell. *Information, Systems and Information Systems - making sense of the field*. Wiley, 1998.
- Ciborra, C.U. and G.F. Lanzara. Formative Contexts and Information Technology: Understanding the dynamics of innovations in organizations. *Accounting, Management & Information Technology 4*, 2, 1994, 61-86.
- Davenport, T.H. *Information Ecology*. Oxford University Press, 1997.
- Greenbaum, J. and M. Kyng (eds.). *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Ass., Hillsdale, New Jersey, 1991.
- Greenwood, D.J. and M. Levin. *Introduction to Action Research - Social Research for Social Change*. Thousand Oaks, Sage, 1998.
- Grudin, J. Interactive Systems Bridging the Gaps between Developers and Users. *IEEE Computer*, April 1991, 59-69.
- Grudin, J. Communities of Design Practice. In Brown and Duguid (1994).
- Hammer, M. and J. Champy: *Reengineering the Corporation. A Manifesto for Business Revolution*. Nicholas Brealey Publishing, London, 1993.
- Hughes, J. A., D. Randall, and D. Shapiro. From Ethnographic record to System Design: Some Experiences From the Field. *Computer Supported Cooperative Work (CSCW): An International Journal 1*, 3, 1993, 123-141.
- Jacobson, I.M. et al: *Object-oriented Software Engineering*. Addison-Wesley, Wokingham, 1992.
- Kapor, M. A software design manifesto: Time for change. *Dr. Dobbs Journal*, 172, January 1991, 62-68.
- Kensing, F. Prompted Reflections: A Technique for Understanding Complex Work. *Interactions 1*, 1998, 7-15.
- Kensing, F. Method Design and Dissemination. In J. Pries-Heje et al (eds): *Proceedings of The Seventh European Conference on Information Systems*, Copenhagen, 1999.
- Kensing, F. and J. Blomberg. PD meets CSCW - Issues and Concerns. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*. 7, Nos. 3-4, 1998, 167-185.
- Kensing, F., J. Simonsen and K. Bødker. MUST - a Method for Participatory Design. *Human-Computer Interaction 13*, 2. 1998a, 167-198.
- Kensing F, J. Simonsen, and K. Bødker. Participatory Design at a Radio Station. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*. 7, Nos. 3-4, 1998b, 243-271.
- King, J.L. Life and Death in the Borderlands. In Brown and Duguid (1994).
- Kyng, M. and L. Mathiassen: Systems Development and Trade Union Activities, in Bjørn-Andersen, N. (ed.): *Information Society, for richer, for poorer*, North Holland, Amsterdam, 1982.
- Kyng, M. and L. Mathiassen. *Computers and Design in Context*. MIT Press, Cambridge, MA, 1997.
- Lave, J. and E. Wenger. *Situated Learning: Legitimate peripheral participation*. Cambridge University Press, 1991.
- Lederer, A.L. and V. Sethi: Critical Dimensions of Strategic Information Systems Planning. *Decision Sciences*, 22, 1991, 104-119.

- Naur, P. Programming as Theory Building. *Microprocessing and Micorprogramming 15*, 1985, 253-261.
- Newman, S. Interpretation, Negotiation, and Practice in System Design. In Brown and Duguid (1994).
- Näslund, T. Computers in Context - But in Which Context? In Kyng and Mathiassen (1997)).
- Orlikowski, W. J. Learning from Notes: Organizational Issues in Groupware Implementation, *Proceedings of the Conference on Computer-Supported Cooperative Work*, ACM, New York, 1992, 362-369.
- Simonsen, J., and F. Kensing. Using Ethnography in Contextual Design, *Communication of the ACM 40*, 7, 1997.
- Simonsen, J.: How do we take care of strategic alignment? Constructing a design approach. *Scandinavian Journal of Information Systems*, 11. 1999.
- Strauss, A. and J. Corbin. *Basics of Qualitative Research. Grounded Theory - Procedures and Techniques*. Sage, 1990.
- Suchman, L. Do Categories Have Politics? The language/action perspective reconsidered. *Computer Supported Cooperative Work (CSCW): An International Journal 2*, 3, 1994, 177-190.
- Suchman, L. Centers of Coordination: A case and some themes. In Resnick, L. B., Säljö, R., Pontecorvo, C., & Burge, B. (Eds.) *Discourse, Tools, and Reasoning: Essays on Situated Cognition*. Berlin: Springer-Verlag, 1997, 41-62.
- Wagner, I. D. Shapiro, G. De Michaelis, and F. Kensing: The Computer-Supported Design of Artefacts and Spaces. Esprit LTR project NO 21870. First phase final report, Vienna, February 1997.
- Winograd, T. Designing the Designer. In Brown and Duguid (1994).