Handbook of Human-Computer Interaction
Second, completely revised edition
M. Helander, T.K. Landauer, P. Prabhu (eds.)

Chapter 11

Participatory Practices in the Software Lifecycle

Michael J. Muller
Boulder, Colorado
USA

Jean Hallewell Haslwanter
Tübingen
Germany

Tom Dayton
New Jersey
USA

11.1 Introduction

Participatory design has become increasingly important over the past several decades. Early work in Scandinavia (for review, see, e.g., Bjerknes and Bratteteig, 1995; Ehn and Kyng, 1987) has recently been complemented by work in other countries (e.g., Muller et al., 1991; Schuler and Namioka, 1993). There are many important contributions in areas of theory, research, practice, assessment, and analysis.

In this chapter, we focus on practices: Our goal is to help practitioners find methods, techniques, and procedures that they can use for participatory work. For this purpose, we have limited our scope to methods and techniques that are relatively well-defined as courses of action, suitable for adoption by practitioners without a great deal of additional research. Our concern is to help practitioners introduce (or to expand) their participatory practices in conventional software lifecycles. We hope that, in this way, we may become part of a growing community that is expanding the space for democratic principles and practices in the workplace. For these reasons, we have restricted our scope to approaches that are more than the use of a particular technology, or the creation of a particular artifact or representation. We believe that vital work is taking place in these areas, and we hope to contribute to these areas ourselves. Where that work included a component of guidance for new practitioners, we have attempted to include it.

11.1.1 The Plan of This Chapter

We begin with a brief introduction to what has come to be known as *participatory design*, including some of the questions that have arisen concerning the boundaries of participatory practice. We expand the discussion from the important—but limited and limiting—concept of design per se, toward a recognition that almost every phase of the software lifecycle may be improved through direct participation by users. This broad notion of the lifecycle serves as the context for the rest of the chapter.

We then describe a taxonomic space of participatory activities (see Table 1). In guiding the reader through that space, we note participatory practices that may be useful at different points within the software lifecycle. We also organize the practices in terms where they are likely to be used: in users' world, in the software professionals' world, or in a space intermediate between the two. Because many of the practices

Table I. Taxonomy of Participatory Methods, Organized by Phase of Lifecycle and by Site of Activity

	Problem Identification & Clarification	Requirements & Analysis	High-Level Design	Detailed Design	Evaluation	End-User Customization	Re-Design
Users' World	• Forum Theatre • Ethnography • Mock-Ups • Search Conference • Starting Conference	• Artifact Walkthrough • Blueprint Mapping • CARD • Ethnography • Forum Theatre • Mock-Ups • PICTIVE • PictureCARD	• Artifact Walkthrough • Lunchbox	• Artifact Walkthrough	• Forum Theatre • Interface Theatre	• Buttons	
Between Worlds	• Future Workshop • Graphical Facilitation • Layout, Organization, & Specification Games • Scenarios • Storytelling Workshop • Translators	• CISP • Collab. Design Workshop • Coop. Reqmts. Capture • CUTA • Future Workshop • Graphical Facilitation • Layout, Organization, & Specification Games • Participatory Ergonomics • Scenarios • Translators • Work Mapping	• ACE • CARD • Graphical Facilitation • Metaphors Game • Mock-Ups • Scenarios • Translators	• CISP • Collaborative Design Workshops • Critics • Graphical Facilitation • Icon Design Game • Scenarios	• CISP • Collab. Design Workshop • Cooperative Evaluation • Mock-Ups • Participatory Heuristic Evaluation • Participatory Ergonomics • Pluralistic Walkthrough • Scenarios • Storyboard Prototyping • Translators • Work Mapping	• Critics	• Critics • Priority Workshop
Software Professionals' World	• Workshop for O-O GUI Designing	• KOMPASS • Prototyping • TOD • Workshop for O-O GUI Designing	• HOOTD • KOMPASS • PrOTA • Prototyping • TOD • Workshop for O-O GUI Designing	• BrainDraw • HOOTD • PICTIVE • PrOTA • Prototyping • Video Prototyping	• CARD • PICTIVE • Prototyping • TOD • Workshop for O-O GUI Designing		
Methods that span *Multiple* Lifecycle Phases	• ACOST • CESD • Codevelopment • Conceptual Toolkit in CSCW Design		• Contextual Design • Contextual Inquiry • Diaries • ETHICS • FIRE		• Florence • Group Elicitation Method • Hiser Design Method • JAD • ORDIT	• SSADM • SSM • STEPS • UTOPIA	

may be used at multiple phases in the lifecycle, we postpone detailed descriptions of the practices until the Appendix.

Our taxonomy may be used by readers to select one or more participatory practices to incorporate into software lifecycle activities. We also describe some models of the software lifecycle that are themselves organized, at least in part, around the concept of participation. These participatory whole-lifecycle models may provide better *continuity of participation* than the less ambitious piece-part strategy of selecting one or two participatory practices for incorporation into a conventional lifecycle model. Practitioners may wish to consider the risks and benefits of whole-lifecycle approaches versus piece-part approaches.

We then briefly review the evidence for the success of participatory approaches, and conclude by reviewing problems that remain to be solved.

We especially encourage the reader to tell us about other practices that we have omitted, or about new applications of old practices of which we are unaware.

11.1.2 Participatory Design

There is no single definition of "participatory design" (PD) that satisfies all researchers and practitioners in this field. The core of the term as it is used in this chapter on human-computer interfaces, is that the ultimate users of the software make effective contributions that reflect their own perspectives and needs, somewhere in the design and development lifecycle of the software. By this we mean active participation—something more than being used as mere data sources by responding to questionnaires or being observed while using the software. User participation is no longer restricted to designing per se; it has proven valuable in activities throughout the entire software lifecycle. We will not offer a more precise definition, in deference to the diversity of principled positions that have been developed by our colleagues. For some sense of the disagreements about the definition and the nature of PD, see Hallewell Haslwanter and Hammond (1994), and the debates in *Communications of the ACM* (1993) and in *Scandinavian Journal of Information Systems* (1994).

In this section, we consider three convergent motivations for participatory approaches:

- **Democracy.** The first theme was stated clearly in the original Scandinavian formulation of participatory design. That work was conceived and undertaken in the context of a movement toward workplace democracy, and the development of workers'competence and power to influence decisions that would affect their work and their workplaces (e.g., Ehn and Kyng, 1987). This motivation remains strong today in both Scandinavian practice (e.g., Beck, 1996; Bjerknes and Bratteteig, 1995) and in some non-Scandinavian practice (e.g., Bernard, 1996; Blomberg, Suchman, and Trigg, 1995; Boy, in press; Floyd, Züllighoven, Budde, and Keil-Slawik, 1992; Greenbaum, 1996; Greenbaum and Sclove, 1996; Muller, 1996b; Segall and Snelling, 1996).

- **Efficiency, Expertise, and Quality.** A second theme has emerged from North American practice (e.g., Holtzblatt and Beyer, 1993; Noro and Imada, 1991; Wixon, Holtzblatt, and Knox, 1990). Effectiveness of software design and development is improved by including the users' expertise. Efficiency is improved by getting users to collaborate in design instead of merely to provide input for other designers or feedback on a completed design; and by involving users early in the design process, before much investment has been made in any design. Quality of the design and the resulting system is improved through better understanding of the users' work, and better combination of the diverse and necessary backgrounds brought by various participants (e.g., Braa, 1996). One way to restate this theme is in terms of *epistemological effectiveness*: That is, no single person or discipline has all the knowledge that is needed for system design. Direct participation by end-users is seen, in this context, as a means of enhancing the process of gathering (and perhaps interpreting) information for system design.

- **Commitment and Buy-In.** A third theme occurs in the area of *organizational development*. In this view, a system is more likely to be accepted by its "downstream" end-users if those users are involved in certain "upstream" formative activities (e.g., Macaulay, 1995).

There have been related developments in other areas. Participatory *action research* shares—and in some ways improves on—the democratic motivations of participatory *design* (e.g., Burkey, 1993; Reason and Rowan, 1981). Several participatory design practices have explicitly developed from practices in the fields of community planning (e.g., Jungk and Mullert, 1987; Nisonen, 1994) and social system and policy development (e.g., Ackoff, 1974; Boy, in press; Warfield, 1971). Developments in areas of labor theory (e.g.,

Braverman, 1984), activity theory (Bødker, 1990), feminist analysis in general (Albrecht and Brewer, 1990; Balka, 1991, 1993; Balka and Doucette, 1994; Benston and Balka, 1993; Greenbaum and Kyng, 1991; Linn, 1987; Suchman and Jordan, 1988), feminist constructivism in particular (Belenky, Clinchy, Goldberger, and Tarule, 1986), and critical theory (for a starting point, see Muller, 1995a, 1996b) have also contributed to participatory theory and practice. These relationships are not surprising. Indeed, participatory design emphasizes the combining of dissimilar knowledges. It would be surprising if theory and practice in this area did not communicate with theory and practice from dissimilar fields.

Individual theorists and practitioners disagree on the relative importance of the three themes described in this section. We have argued that the three themes are often convergent (Kuhn and Muller, 1993; Muller et al., 1991), and one of us has collected evidence showing that the economic and political motivations have sometimes been included in the same project (Hallewell Haslwanter, 1995). In brief, democratic work at the level of design has the potential to improve both knowledge gathering *and* "downstream" commitment and buy-in—both within the software development process and within the users' work. In this analysis, democracy, epistemology, and commercial effectiveness go hand-in-hand.

11.1.3 Participation and Its Discontents

The increasing popularity of participatory work has led some researchers and practitioners to be wary of the phrase "participatory design" itself, and of the ways in which that phrase has been interpreted by others. To varying extents, we share their discomforts. In a review chapter such as this, we prefer to phrase these discomforts as questions—for example:

- Some knowledge elicitation techniques emphasize "participation" by workers for the purpose of increasing the knowledge that will be used primarily by the analyst or by a knowledge engineer (e.g., in the development of expert systems). What is the meaning of "participation" in describing practices in which the users give up their work-oriented knowledge without receiving in return any effective decision-making role in design? When does "participation" become exploitation?

- Certain well-known usability-testing practices in human-computer interaction treat users as measurement indicators of the productivity associated with a product, without considering the users' needs for comfort, dignity, respect, or a quality work environment. What is the meaning of "participation" when someone other than the users chooses which attributes of the users' experience are relevant? When does "participation" become objectification?

- Some practitioners have developed highly effective methods for participation by potential customers, in determining the most attractive attributes of mass market products. What is the meaning of "participation" when people contribute to activities that are then used to develop more effective advertising campaigns? When does "participation" become manipulation?

- Certain organizational development approaches have reassured managers that problems may be solved at a lower level in the organization, while decisions about those problems are retained at a higher level in the organization. What is the meaning of "participation" without decision-making? When does "participation" become illusion?

We three authors have worked together to varying extents, and yet we have come to somewhat different answers to these difficult questions (for related sets of questions, see Mackay, 1995; Muller, 1995a, 1995b). In this chapter, we will not attempt to provide examples of our own or others' answers to these questions. We hope that readers will consider the ethical and political dimensions of their own practices as they select from the techniques that we describe in this chapter.

11.1.4 Against Methods

Some practitioners believe that the methods-oriented approach we have taken here is inappropriate (e.g., Blomberg et al., 1995; in some ways, Bødker, Christiansen, and Thüring, 1995; see also the general disinclination to provide precise methodological descriptions in the Scandinavian work, as evidenced in many contributions to Greenbaum and Kyng, 1991).

In this view, methods are problematic for two reasons. First, this view's engineering assumptions regarding a "method" are that a method is a straightforward, usually linear or sequential, series of well-understood steps that will lead to a predictable and relatively guaranteed outcome. These assumptions do not hold in the area of participatory methods, because any methodological description in our field provides

merely a scaffold or an infrastructure for a complex group process that is neither linear nor well-understood. Of course, the complex group process is essential, and is in fact the heart of participatory work. All of the details of methods that we present in this chapter are intended as support for this fundamental human-to-human communication.

A second objection to a methods-oriented approach is that some practitioners may be able to *name* a participatory method that they claim to have applied, but they may *apply* that method in a way that is contrary to the goals of participatory design. For example, two of us have been present when human factors analysts claimed to have done "participatory design with the developers—we used PICTIVE." Yet there were no users present in these allegedly participatory activities. This story is an example of an appropriation of the language of participatory design for a non-participatory activity. Some of our colleagues cite such problems as examples of the dangers of presenting participatory methods for use by people who may not agree with the democratic motivations of participatory design[1].

We have nonetheless decided to pursue a methods-oriented approach, because (1) We believe in the value of iterative, flexible, loosely guiding methodological approaches in general (Dayton, 1991; Karat and Dayton, 1995), including non-participatory ones; and (2) We believe that a clear "guide to participatory methods" can aid in the piecemeal, incremental, experimental introduction of participatory approaches in organizations that have previously relied on non-participatory software methods. Good practitioners of every background look for improved methods. We offer these participatory practices in the hope that practitioners will try at least *one* appropriate method in their work, and see the values of participation. They may then be able to use this chapter to find other methods that may apply to other aspects of their work. In this way, we hope that this chapter may help to bring greater and greater participation to work within a conventional software lifecycle. Recent analyses of the state of information engineering (Mulder, 1994) and HCI (Strong, 1994) agree on the need to introduce more participatory practices.

We agree with our colleagues about the dangers of misapplication of participatory methods in non-participatory work (see, e.g., the question that we listed in the preceding section). However, we are convinced that these dangers and problems will require searching inquiries, discussions, and reflections (see, e.g., Muller, 1995b). Making methods available to practitioners may in fact be a way of hastening these needed discussions.

We hope that these discussions will include a clarification of the concept of participation. During our preparations for this chapter, some practitioners pressured us to include practices that we considered to be borderline participatory at best. With mixed feelings, we have generally decided to err on the side of being too inclusive, rather than not inclusive enough. Perhaps the inclusion of borderline cases will help the participatory community to develop a clearer definition of its own concepts. We trust that future versions of this chapter, prepared by others or ourselves, will correct any mistakes that we have made.

There are alternatives to this piecemeal approach of adding one or more participatory practices to conventional lifecycle models. A later section of this paper briefly reviews what we have called *participatory whole-lifecycle models*. We encourage practitioners to consider a wholesale adoption of participatory practice through such lifecycle models. We believe that, even with a participatory lifecycle model, practitioners may need to search for additional methods. Thus, we hope that this chapter will be useful for practitioners who work within either conventional lifecycle models or participatory lifecycle models.

11.1.5 History of This Work

Work on this collection of participatory practices began in collaboration with Ellen White and Daniel Wildman. Conference participants at the CHI '92 conference were invited to write their own contributions to a "participatory poster of participatory practices" (Muller, Wildman, and White, 1992). The resulting, early view of participatory practices was published in Muller, Wildman, and White (1993).

One of us conducted a survey of participatory

[1]We believe that methods from participatory design may be useful in non-participatory domains. In fact, all of us have been involved in such activities (e.g., Muller, Hallewell Haslwanter, and Dayton, 1995; Nielsen et al., 1992). However, we recommend that practitioners use very clear language regarding such applications of methods. In Muller, Hallewell Haslwanter, and Dayton (1995), we distinguished between the *participatory analysis methods* that we were borrowing for the purposes of a non-participatory *collaborative analysis* that, for organizational reasons, excluded the end-users. In Nielsen et al. (1992), we borrowed a method from *participatory design* for the purpose of helping developers to understand concepts in graphical user interfaces. Our purpose here was education of developers, not codevelopment of systems with end-users. In both of these examples, we believe that the methods contributed to our work. In both of these examples, we were clear that the methods were being used in a non-participatory fashion, and that they therefore should be described in non-participatory terms.

methods and outcomes as part of her thesis research (Hallewell Haslwanter, 1995). In connection with that survey, the participatory poster was presented for further codevelopment at conferences in Australia, Europe, and North America in 1994 and 1995 (Hallewell Haslwanter, Muller, and Dayton, 1994; Muller, Hallewell Haslwanter, and Dayton, 1994, 1995).

This chapter is, then, a report of a continuing work-in-progress that is being carried out by members of the field of participatory design. The "authors" of this chapter serve only as the current facilitators of that process.

11.2 Taxonomies of Participatory Practices

This chapter presents the many participatory methods labeled by their places in several dimensions, such as the sizes of the groups the methods can handle, and the points in the software lifecycle where the methods are used. We hope that this organization will help readers understand the universe of participatory methods, and quickly find methods that are appropriate to the readers' particular situations.

11.2.1 Practices in the Lifecycle?

Early work in participatory design tended to treat the participatory activity as the overwhelming focus of the endeavor. It was often difficult—if not problematic—to situate the participatory work in the context of a software lifecycle or software development organization. We know of two responses to that set of problems:

- **PANDA.** The 1992 taxonomy of participatory practices was organized across a dimension of the relevant phases of the software lifecycle. Our original motivation for this organization was to help practitioners find useful practices. An unintentional outcome was a very clear demonstration that "participatory design" practices in fact are used in many more phases of the lifecycle than just in design. One of us discovered in her survey of practitioners that participatory activities were used most often in design, but also in the following phases: requirements, prototyping, system design, and system test. Her research also showed that over 50% of respondents used participatory activities in *all* lifecycle phases (Hallewell Haslwanter, 1995). One of us is exploring a broader treatment that can be rendered as a pronounceable term as PANDA, for Participatory ANalysis, Design, and Assessment (Muller, 1996b). This formulation highlights participatory activities that contribute to the *analysis* of work and to the *assessment* of computer systems and work systems, as well as to issues of *design*.

  However, as we will show, the PANDA description is in fact too narrow. There are participatory methods falling outside of the limited domain addressed by a literal interpretation of the PANDA formulation.

- **Tools for the Toolbox.** In 1992 and 1993, Kensing and Munk-Madsen (1993) developed an analysis of participatory methods that they termed "tools for the toolbox" (recently extended as the MUST conceptual organization for participatory work by Kensing, Simonsen, and Bødker, 1996). Their spirit was very much in keeping with our position as described above in section 11.1.4 ("Against Methods"): to provide practitioners with tools that could be applied in conventional software lifecycle models.

  Kensing and Munk-Madsen went beyond our own 1992 and 1993 work, however, by including in their analysis both participatory methods and formal methods. Their analysis considered both the relatively concrete and end-user-accessible methods of participatory design *and* a subset of the relatively formal and end-user-inaccessible methods that are used by software professionals for communication *within* their own profession. Their analysis suggested that there may be pairs of methods—one participatory method and one formal method—that complement one another. We have attempted to include this component of analysis in this chapter.

More generally, there has been concern to make usability methods more usable in the software lifecycle (Dayton, 1991; Dray, Dayton, Mrazek, Muckler, and Rafeld, 1993; Hefley et al., 1994; Hix, Hartson, and Nielsen, 1994; Kensing and Munk-Madsen, 1993; Olson and Moran, 1993; Strong, 1994). The critique of Olson and Moran is particularly apt. They suggested that each method be described in terms of problem to be solved, materials, process, and result. In this chapter, we have extended their proposed descriptors in directions that appear to us to be consistent with participatory design[2].

In the remainder of this section, we describe our

[2]For a different conceptual organization of a space of methods, see Braa (1996).

revised analytical space of participatory methods. Table 1 provides one organization of the methods. The Appendix provides more details on each method.

11.2.2 Describing Practices

Our analytical space is organized according to nine major attributes:

- **Abstract:** What is the practice supposed to do? Why might a practitioner choose this practice? In some ways, this attribute is related to the question of the phase or phases of the software lifecycle.

- **Object Model:** What materials are used in the practice?

- **Process Model:** What do people do to communicate with one another? How do they make decisions? What do people do with the materials in the Object Model?

- **Participation Model:** Who is involved in the work? Are people with specific roles needed for the practice to work well?

- **Results:** What tangible or intangible benefit is produced as a outcome of the work? How is this result used?

- **Phases of the Lifecycle:** In many cases, participatory methods were developed to meet a particular need within a particular phase of the software lifecycle. Where this is the case, we have indexed the method to that phase.

  However, some methods were developed for application to multiple lifecycle phases. In other cases, practitioners have discovered that a method originally developed for one particular phase can be applied with little modification to other phases. In these cases, we have indexed the method to multiple lifecycle phases. To avoid multiple descriptions, we have placed all of the detailed descriptions of method not in the sections on lifecycle phases, but rather in a single omnibus appendix to this chapter. In many cases, a single method description in the appendix is indexed from several lifecycle sections.

  The span of lifecycle activities in our analysis may be surprising to some North American practitioners. Following the Scandinavian approaches and also the persuasive analysis of Floyd (1987), we begin our lifecycle model with very early activities that are concerned with problem identification and problem clarification (e.g., Is there a problem to be solved? What is it? Would a computer system help to solve the problem?). Conventional North American lifecycle models often treat this phase as *external* to the lifecycle. Experiences from Scandinavia, from participatory action research, and from some of the pragmatically-oriented research and application, have shown the importance of end-user participation in this early phase of understanding and decision-making.

  We also extend the span of lifecycle activities somewhat later than some North American models, to include customization of computer systems in the field, and participatory redesign of existing systems.

- **Complementary Formal Methods:** Can the practice work in conjunction with a formal lifecycle method that might be known to software professionals?

- **Group Sizes:** Many participatory practices were designed for work with small groups. A few practices appear to require a pair of people with well-defined roles (e.g., user and developer). Other practices were designed for work with larger groups, or in a few cases with very large groups. We have attempted to indicate the group sizes for which each practice appears to work well.

- **References:** How may the reader find out more about this practice? Where possible, we list formal publications that are likely to be available in research libraries. When necessary, we resort to tutorial notes or position papers at conference workshops, recognizing that these resources will be more difficult for readers to obtain. In a few cases, there is no documentary record available: In these cases, we have provided contact information for the originator of the practice.

11.3 Participatory Whole-Lifecycle Models

Our focus in this chapter is on specific methods, techniques, and procedures for participatory work, which can be applied within existing, conventional software lifecycle approaches. We therefore can only briefly mention several promising approaches toward a *par-*

ticipatory model of the software lifecycle (specific references to each of the approaches named in this section may be found in the Appendix). The ETHICS, STEPS, and CESD models for lifecycle-spanning participatory approaches are being developed in the United Kingdom, Germany, and Denmark, respectively. For Denmark, see also the Conceptual Toolkit project. The US has seen a much more restricted experiment in the Codevelopment project at Xerox.

Several people have developed integrated approaches that provide participatory continuity *across* a significant subset of lifecycle phases. Early phases (i.e., from problem identification through design) have been spanned by the UTOPIA project in Denmark, by Contextual Inquiry and Contextual Design in the US, by the Hiser Design Method in Australia, and by Soft Systems Methodology and ORDIT in the United Kingdom. The FIRE project in Norway has begun to integrate the later phases of the lifecycle.

Many of these lifecycle-oriented approaches provide conceptual, organizational, and political frameworks within which specific participatory practices may be located. Thus, several of the practices that are described by themselves (e.g., workshops, prototyping) also serve as components of the more integrated, continuous participatory approaches.

11.4 Approaches from Other Domains

We briefly note several other domains that may be of value to readers who do not find in this chapter, methods to suit their purposes. Participatory action research has been concerned with developing co-investigatory relationships between researchers and the people whom they study (e.g., Burkey, 1993; Reason and Rowan, 1981). Feminism has explored ways of reducing the objectification and exploitation of the people who are studied in social science research (e.g., Reinharz, 1992; see also Belenky et al., 1986). Grounded theory may provide a means of developing theoretical contributions directly from the social systems that are studied, as contrasted with the more traditional approach of fitting observations into a theoretical structure that was developed independently of those observations (e.g., Glaser and Strauss, 1967; for methods, see especially Strauss and Corbin, 1990). Finally, activity theory has recently been argued to have potential value to the field of human-computer interaction (e.g., Nardi, in press) and specifically participatory design (Bødker, 1990).

11.5 Does Participation "Work?"

The readers of this chapter may reasonably wish to ask whether participation in software activities by end-users is worth the time and apparent trouble? In this section, we briefly address that question.

First, we note that different methods, practices, and procedures have different time-courses. Ethnography, for example, typically involves a long relationship. At the other extreme, some participatory practices can be begun and concluded within a single session. We wish to emphasize that participatory activities do not necessarily require more time than conventional practices: Some participatory activities, in fact, take much less time than corresponding or complementary formal software lifecycle methods. The strengths and weaknesses of these different approaches must, of course, be considered as practitioners and researchers choose practices that fit within their particular constraints.

In an earlier section, we introduced several convergent motivations for participation. Economic motivations included the incorporation of user expertise, and increases in the efficiency and quality of knowledge gathering. Organizational motivations included improved communications, and earlier understanding and commitment by "downstream" staff and organizations). Political motivations included the enhancement of workplace democracy, and the direct articulation of end-users' needs by the end-users or their organization. Several researchers have examined each of these areas of potential contribution. An early study of user "involvement" in lifecycle activities showed no reliable effects (Ives and Olson, 1984). However, the concept of "involvement" appears in retrospect to have been too broad. It included direct participation, but it was large determined as a subjective estimate of the users' sense of involvement, ascertained as a unidimensional numerical response to a survey item. Moreover, the survey respondent was usually a manager in the user organization, rather than an end-user. Thus, this early study has less to tell us about the effects of user *participation* than we might have thought. More recent studies have shown complex but reliable positive effects of user participation (not just "involvement") in lifecycle activities. Cotton, Vollrath, Froggatt, Lengnick-Hall, and Jennings (1988) studied multiple dimensions of participation, and multiple outcome measures. Their results showed that certain forms of user participation were beneficial for certain economic and organizational outcomes. Saarinen and

Saaksjarvi (1989) showed a similar, complex set of relationships. Their study helped to tease apart to separate but equally valuable contributions of end-users as participants and of analysts as participants. Thus, careful studies of end-user participation have shown both (a) the value of end-users' contributions to economic and organizational goals, and (b) the value of the contributions of analysts and other professionals to those same goals.

These three studies—and especially the later two—provide some support for our first two motivations: economic and organizational. We also note that some of the work on individual methods, techniques, and procedures has included assessments of their effectiveness. Case studies of participatory design are provided by Greenbaum and Madsen (1993), Hallewell Haslwanter (1995), Mumford (1993), Thoresen (1993), and URCOT (1993).

Outcomes regarding the third motivation—workplace democracy—are more difficult to measure. Good qualitative summaries of these complex outcomes in Scandinavia are provided by Ehn and Kyng (1987) and by Bjerknes and Bratteteig (1995). We know of no similar omnibus summaries for participatory practices in software activities outside of Scandinavia.

11.6 Some Outstanding Research Problems in Participation

In this section, we briefly provide our perspective on unsolved problems in the area of participatory work with end-users in software lifecycles.

11.6.1 Relationships Between Participatory Practices, Formal Software Methods, and Contractual Models

As we noted earlier, it has been difficult to integrate participatory activities into the software lifecycle. Kensing and Munk-Madsen (1993) made progress in this area by suggesting correlative participatory and formal methods. We have attempted to continue their approach in this chapter. Nonetheless, we believe that there are many more relationships between participatory and formal methods that remain to be discovered or created. Establishing these relationships and their syncretic benefits may require modification to both participatory and formal methods: We anticipate that working through these tradeoffs will be a complex, difficult, and rewarding undertaking.

A more significant problem arises with certain formal lifecycle methods that are gaining in popularity. We outline two problems in this section, to provide a sense of the challenges facing participatory theorists and practitioners:

- **Use cases.** Certain object-oriented methodologies encourage the construction of use cases as scenarios of user activities related to the software system (e.g., Jacobson, 1995, and more generally Jacobson, Christersson, Jonsson, and Övergaard, 1992). There are several problems with these approaches. First, the use case model is almost always written with the software system as the overwhelming focus of attention. The use case model is thus an example of a *product-oriented paradigm*, which gives too much priority to the software, and too little priority to the end-users' work or life processes (see Floyd, 1987, for an influential distinction between product-oriented and process-oriented paradigms in software engineering). At a deeper level, each use case is a definition of user actions by system designers: Its words carry a connotation of end-user focus and work analysis, but its substance is in fact centered on software features that may or may not be related to end-users'needs. In practice, we have seen development organizations utilize the use case model as a replacement for work with end-users. The problem, then, is to make effective user participation an integral part of the use case model and its related practices.

- **ISO 9001.** The ISO 9001 standard for quality assurance encourages an agreement or contract between a software development organization and a user organization. In the analysis of Braa and Øgrim (in press), the standard focuses entirely on technical quality, as contrasted with other potential quality areas such as use quality, aesthetic quality, symbolic quality, and organizational quality. The standard also omits any formal relationship between the development organization and an organization that might represent the needs of the end-users (i.e., as contrasted with the needs of their managers, as reflected in the user organization). Thus, the standard may be unethical, according to Braa and Øgrim. It may also explain recent informal reports from Scandinavia, of end-user constituencies being omitted from discussions of new or revised systems, despite the fact that they had been primary participants in earlier discussions with the same employers. As the ISO 9001 and 9003 standards become more

widely used, we may find a need to modify them so that they explicitly include end-user participation as part of the assurance of quality, and, perhaps more crucially, end-user organizations (e.g., unions or other forms of representation) as among the parties who are directly involved in contractual relationships with developers and with user organizations.

11.6.2 Non-Organized and Contingent Workers

Participatory activities are easiest to conduct when the end-users are organized and self-represented through an autonomous organization, such as a union (e.g., Greenbaum, 1993; Grudin, 1990; Muller, 1996a). Nonetheless, many practitioners and organizations are engaged in participatory activities with workers who do not have formal representation. There are different problems and obstacles for these participatory venues (Bruce and Kirup, 1987; Carmel, Grudin, Juhl, Erickson, and Robbins, 1994; Darwin, Fitter, Fryer, and Smith, 1987; Greenbaum, 1993; Grudin, 1990). The problems are perhaps most difficult when the workers go from one temporary job to another (e.g., Greenbaum, Snelling, Jolly, and Orr, 1994; Greenbaum and Orr, 1995). Effective participation by non-organized or contingent workers presents a number of daunting challenges for which we are still searching for solutions. These problems are magnified when we consider the constituencies for on-line or distance educational services (Muller, 1995a).

11.6.3 Finding Partners and Structuring Relationships

Participation involves risks—especially for less powerful participants and constituencies. In the introductory section, we outlined several ethical questions. From the perspective of the practitioner's perspective, these questions may seem to be related to fairness, or even efficiency. From the perspective of end-users, these questions may seem to be related to exploitation, or even betrayal. Decision-makers in end-user organizations, such as unions, must weigh the risks of participation to their constituencies and also to themselves—for instance, as elected leaders who will be held responsible for their commitments by their membership. There is some work in Scandinavia toward reducing these risks (e.g., Bjerknes and Bratteteig, 1995; Ehn and Kyng, 1987). There appears to be less work outside of Scandinavia that addresses this difficult area.

If the risk is considered worthwhile, then there arises the question of the form of relationship that is to develop. In one sense, the diverse participants form a team. In another sense, each participant remains—and *should* remain—a member of her or his group of origin. Feminists working in multi-cultural contexts have explored differences among concepts of *coalitions*, *alliances*, and other forms of inter-organizational relationship (e.g., Albrecht and Brewer, 1990). We are unaware of similarly detailed analyses in the software area.

11.6.4 Balancing Group and Individual Activities

Participatory work is by definition shared among group members. However, there are other work components that are best done by individuals, or by subgroups. Examples include a human factors worker applying a detailed style guide to a rough design, and a union stewards' group developing a work force position with respect to a job design decision.

In the more formal constructs of Clement and Wagner (1995), subgroups and individuals have needs both of aggregation (working together, sharing data and experiences) and disaggregation (working alone, keeping data and experiences private). Much of the research literature has concerned needs for, and technologies in support of, aggregation. Aside from work on privacy and censorship, there is little research in the software area into needs for, and technologies in support of, disaggregation. Clement and Wagner argued that this area requires more subtle concepts than sharing "versus" privacy, or expression "versus" censorship, and they have provided a starting point for researchers who want to pursue these questions.

11.7 Conclusion

In this chapter, we have attempted (a) to provide a set of guideposts to the considerable progress that has been made in practices in participatory activities, and (b) to outline some of the problems that remain to be solved. Because of the extraordinary diversity of the communities that work toward effective participation, we feel the need to review our efforts to work beyond our own limited perspectives. The materials that we have summarized in this chapter have been gathered on three continents over a period of as many years. Nonetheless, most of the venues at which we conducted our "participatory posters" were professional meetings for privileged, salaried members of the managerial and academic social classes (and students

who aspired to join their ranks). Our information-gathering has thus been biased, despite our best intentions. We look forward to the next iteration of this summary of participatory methods, techniques, and procedures, and we hope that the people who prepare that summary can do so in a more inclusive manner than we have been able to achieve.

11.8 Acknowledgments

We thank Heather Shepherd and Anne McClard for helping find references; Ellen White and Danny Wildman for co-authoring the early participatory posters used to collect and present these data; Judy Hammond and Peter Petocz; and the many people who have contributed information about methods in the several years of this ongoing project.

11.9 References

Ackoff, R. L. (1974). *Redesigning the future—A system approach to societal problems.* New York: Wiley.

Albrecht, L., and Brewer, R. M. (Eds.). (1990) *Bridges of power: Women's multi-cultural alliances.* Philadelphia: New Society.

Balka, E. (1991). *Womantalk goes on-line: The use of computer networks for feminist social change.* Doctoral thesis, Simon Fraser University, Burnaby, British Columbia, Canada.

Balka, E. (1993). Cappuccino, community, and technology: Technology in the everyday life of Margaret Benston. *Canadian Woman Studies Journal / Les Cahiers de la Femme, 13*(2), 62–65.

Balka, E., and Doucette, L. (1994). The accessibility of computers to organizations serving women in the province of Newfoundland: Preliminary study results. *Electronic Journal of Virtual Culture, 2*(3).

Beck, E.E. (1996). P for political? Some challenges to PD towards 2000. In *PDC'96 Proceedings of the Participatory Design Conference*, 117-125.

Belenky, M. F., Clinchy, B. M., Goldberger, N. R., and Tarule, N. M. (1986). *Women's ways of knowing: The development of self, voice, and mind.* New York: Basic Books.

Benston, M., and Balka, E. (1993). Participatory design by non-profit groups. *Canadian Woman Studies Journal / Les Cahiers de la Femme, 13*(2), 100–103.

Bernard, E., (1996). Power and design: Why unions and organizations matter. In *PDC'96 Proceedings of the Participatory Design Conference*, 207.

Bjerknes, G., and Bratteteig, T. (1995). User participation and democracy: A discussion of Scandinavian research on system development. *Scandinavian Journal of Information Systems, 7*(1), 73–98.

Blomberg, J., Suchman, L., and Trigg, R. (1995). Back to work: Renewing old agendas for cooperative design. In *Conference proceedings of Computers in Context: Joining Forces in Design* (pp. 1–9). Århus, Denmark: Aarhus University.

Boy, G. A. (in press). Group elicitation method. To appear in *interactions.*

Braverman, H. (1984). *Labor and monopoly capital.* New York: Monthly Review.

Bruce, M., and Kirup, G. (1987). An analysis of women's roles under the impacts of new technology in the home and office. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge.* Brookfield, VT, USA: Gower.

Braa, K. (1996). Influencing qualities of information systems — Future challenges for participatory design. In *PDC'96 Proceedings of the Participatory Design Conference*, 163-172.

Braa, K., and Øgrim, L. (in press). Critical view of the application of the ISO standard for quality assurance. *Information Systems Journal.*

Burkey, S. (1993). *People first: A guide to self-reliant participatory rural development.* London: Zed.

Bødker, S. (1990). *Through the interface: A human activity approach to user interface design.* Hillsdale, NJ, USA: Erlbaum.

Bødker, S., Christiansen, E., and Thüring, M. (1995). A conceptual toolbox for designing CSCW applications. In *COOP '95: Atelier international sur la conception des systèms coopératifs* [International workshop on the design of cooperative systems] (pp. 266–284). Sophia Antiipolis, France: INRIA.

Carmel, E., Grudin, J., Juhl, D., Erickson, T., and Robbins, J. (1994). Does participatory design have a role in software package development? *PDC '94: Proceedings of the Participatory Design Conference*, 33–35.

---

[3] Michael J. Muller, 1969 Joslyn Court, Boulder, Colorado 80304, USA; muller.chi@xerox.com
Jean Hallewell Haslwanter, Weilerburgstrasse 3, 72072 Tübingen, Germany; jean.hallewell@wago.de
Tom Dayton, 41C Franklin Greens South, Somerset, New Jersey 08873, USA; tdayton.chi@xerox.com

Clement, A., and Wagner, I. (1995). Fragmented exchange: Disarticulation and the need for regionalized communication spaces. In *Proceedings of the Fourth European Conference on Computer-Supported Cooperative Work* (pp. 33–49). Stockholm: Kluwer.

Communications of the ACM. (1993). *36*(10).

Cotton, J. L., Vollrath, D. A., Froggatt, K. L., Lengnick-Hall, M. L., and Jennings, K. R. (1988). Employee participation: Diverse forms and different outcomes. *Academy of Management Review, 13*(1), 8–22.

Darwin, J., Fitter, M., Fryer, D., and Smith, L. (1987). Developing information technology in the community with unwaged groups. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge.* Brookfield, VT, USA: Gower.

Dayton, T. (1991). Cultivated eclecticism as the normative approach to design. In J. Karat (Ed.), *Taking software design seriously: Practical techniques for human-computer interaction design* (pp. 21–44). Boston: Academic Press.

Dray, S., Dayton, T., Mrazek, D., Muckler, F. A., and Rafeld, M. (1993). Making human factors usable. *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, 863–866.

Ehn, P., and Kyng, M. (1987). The collective resource approach to systems design. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 17–57). Brookfield, VT, USA: Gower.

Floyd, C. (1987). Outline of a paradigm change in software engineering. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 191–210). Brookfield, VT, USA: Gower.

Floyd, C., Züllighoven, H., Budde, R., and Keil-Slawik, R. (1992). (Eds.). *Software development and reality construction.* Berlin: Springer Verlag.

Glaser, B., and Strauss, A. (1967). *The discovery of grounded theory.* Chicago: Aldine.

Greenbaum, J. (1996). Post modern times: Participation beyond the workplace. In *PDC'96 Proceedings of the Participatory Design Conference*, 65-72.

Greenbaum, J. (1993). A design of one's own: Towards participatory design in the United States. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 27–37). Hillsdale, NJ, USA: Erlbaum.

Greenbaum, J., and Kyng, M. (Eds.). (1991). *Design at work: Cooperative design of computer systems.* Hillsdale, NJ, USA: Erlbaum.

Greenbaum, J., and Madsen, K. H. (1993). Small changes: Starting a participatory design process by giving participants a voice. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 289–298). Hillsdale, NJ, USA: Erlbaum.

Greenbaum, J., and Orr, J. (1995). Participatory panel: Re-divided work—Implications for work and system design. In *Conference proceedings of Computers in Context: Joining Forces in Design* (p. ii). Århus, Denmark: Aarhus University.

Greenbaum, J., and Sclove, R. (1996). Beyond participatory design. In *PDC'96 Proceedings of the Participatory Design Conference*, 63.

Greenbaum, J., Snelling, L., Jolly, C., and Orr, J. (1994). The limits of PD? Contingent jobs and work reorganization. *PDC '94: Proceedings of the Participatory Design Conference,* 173–174.

Grudin, J. (1990) Obstacles to participatory design. *Proceedings of CHI '90*, 142–143.

Hallewell Haslwanter, J. D. (1995). *Participatory design methods in the context of human-computer interaction.* M. Sc. thesis, University of Technology, Sydney, Sydney, Australia.

Hallewell Haslwanter, J. D., and Hammond, J. (1994). Survey of cross-cultural differences in participatory design. *OZCHI '94 Proceedings*, 317–318.

Hallewell Haslwanter, J. D., Muller, M. J., and Dayton, T. (1994). Participatory design practices: A classification. *OZCHI '94 Proceedings*, 319–320.

Hefley, W. F., Buie, E. A., Lynch, G. F., Muller, M. J., Hoecker, D. G., Carter, J., and Roth, J. T. (1994). Integrating human factors with software engineering practices. *Proceedings of the Human Factors and Ergonomics Society 38th Annual Meeting,* 315–319.

Hix, D., Hartson, H. R., and Nielsen, J. (1994). A taxonomy for developing high impact formative usability evaluation methods. *SIGCHI Bulletin, 26*(4), 20–22.

Holtzblatt, K., and Beyer, H. (1993). Making customer-centered design work for teams. *Communications of the ACM, 36*(10), 92–103.

Ives, B., and Olson, M. (1984). User involvement and MIS success: A review of research. *Management Science, 30*(5), 586–603.

Jacobson, I. (1995). The use-case construct in object-oriented software engineering. In J. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 309–360). New York: Wiley.

Jacobson, I., Christersson, M., Jonsson, P., and Övergaard, G. (1992). *Object-oriented software engineering—A use-case driven approach.* Reading, MA, USA: Addison-Wesley.

Jungk, R., and Mullert, N. (1987). *Future workshops: How to create a desirable future.* London: Institute of Social Invention.

Karat, J., and Dayton, T. (1995). Practical education for improving software usability. *Proceedings of CHI '95,* 162–169.

Kensing, F., and Munk-Madsen, A. (1993). PD: Structure in the toolbox. *Communications of the ACM, 36*(6), 78–85.

Kensing, F., Simonsen, J., and Bødker, K. (1996). MUST — A method for participatory design. In *PDC'96 Proceedings of the Participatory Design Conference*, 129-140.

Kuhn, S., and Muller, M. J. (1993). Introduction to the issue on participatory design. *Communications of the ACM, 36*(6), 24–28.

Linn, P. (1987). Gender stereotypes, technology stereotypes. *Radical Science Journal, 19,* 127–131.

Macaulay, L. (1995). Cooperation in understanding user needs and requirements. *Computer Integrated Manufacturing Systems, 8*(2), 155–165.

Mackay, W. E. (1995). Ethics, lies and videotape. *Proceedings of CHI '95,* 138–145.

Mulder, M. C. (1994). *Educating the next generation of information specialists.* Washington, DC: National Science Foundation.

Muller, M. J. (1996a). Defining and designing the Internet: Participation by Internet stakeholder constituencies. *Social Science Computing Review* 14(1), 30-33.

Muller, M. J. (1995a). Ethnocritical heuristics for HCI work with users and other stakeholders. In *Conference proceedings of Computers in Context: Joining Forces in Design* (pp. 10–19). Århus, Denmark: Aarhus University.

Muller, M. J. (1995b). Ethnocritical questions for working with translations, interpretations, and their stakeholders. *Communications of the ACM* 38(9), 64-65.

Muller, M.J. (1996b). *Participatory activities with users and others in the software lifecycle.* Tutorial at CHI 96.

Muller, M. J., Blomberg, J. L., Carter, K. A., Dykstra, E. A., Greenbaum, J., and Halskov Madsen, K. (1991). Panel: Participatory design in Britain and North America: Responses to the "Scandinavian challenge." *Proceedings of CHI '91,* 389–392.

Muller, M. J., Hallewell Haslwanter, J. D., and Dayton, T. (1994). Updating a taxonomy of participatory practices: A participatory poster. *PDC '94: Proceedings of the Participatory Design Conference,* 123.

Muller, M. J., Hallewell Haslwanter, J. D., and Dayton, T. (1995). A participatory poster of participatory practices. In *Conference proceedings of Computers in Context: Joining Forces in Design* (p. i). Århus, Denmark: Aarhus University.

Muller, M. J., Wildman, D. M., and White, E. A. (1992). *Taxonomy of participatory design practices: A participatory poster.* Poster at CHI '92, Monterey, California, USA.

Muller, M. J., Wildman, D. M., and White, E. A. (1993). Taxonomy of PD practices: A brief practitioner's guide. *Communications of the ACM, 36*(6), 26–27.

Mumford, E. (1993). The participation of users in systems design: An account of the origin, evolution, and use of the ETHICS method. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 257–270). Hillsdale, NJ, USA: Erlbaum.

Nardi, B. A. (Ed.). (1996) *Context and consciousness: Activity theory and human computer interaction* (pp. 387–400). Cambridge, MA, USA: MIT Press.

Nielsen, J., Bush, R. M., Dayton, T., Mond, N. E., Muller, M. J., and Root, R. W. (1992). Teaching experienced developers to design graphical user interfaces. *Proceedings of CHI '92,* 557–564.

Nisonen, E. (1994). Women's safety audit guide—An action plan and a grass roots community development tool. *CPSR Newsletter, 12*(3), 7.

Noro, K., and Imada, A. S. (Eds.). (1991). *Participatory ergonomics.* London: Taylor and Francis.

Olson, J. S., and Moran, T. P. (1993). *Mapping the method muddle: Guidance in using methods for user interface design* (Technical Report No. 49). Ann Arbor, MI, USA: University of Michigan, Cognitive Science and Machine Intelligence Laboratory.

Reason, P., and Rowan, J. (Eds.). (1981). *Human inquiry: A sourcebook of new paradigm research.* Chichester: Wiley.

Reinharz, S. (with Davidman, L.). (1992). *Feminist methods in social research.* New York: Oxford University Press.

Saarinen, T., and Saaksjarvi, M. (1989). The missing concepts of user participation: An empirical assessment of user participation and information system success. In *Proceedings of the 12th IRIS (Information System Research in Scandinavia)* (pp. 533–553). Århus, Denmark: Aarhus University.

Scandinavian Journal of Information Systems. (1994). *6*(1).

Schuler, D., and Namioka, A. (Eds.). (1993). *Participatory design: Principles and practices.* Hillsdale, NJ, USA: Erlbaum.

Segall, P., and Snelling, L. (1996). Achieving worker participation in technological change: The case of the flashing cursor. In *PDC'96 Proceedings of the Participatory Design Conference*, 103-109.

Strauss, A., and Corbin, J. (1990). *Basics of qualitative research: Grounded theory procedures and techniques.* Newbury Park, CA, USA: Sage.

Strong, G. W. (with Gasen, J. B., Hewett, T., Hix, D., Morris, J., Muller, M. J., and Novick, D. G.). (1994). *New directions in human-computer interaction education, research, and practice.* Washington, DC: National Science Foundation.

Suchman, L., and Jordan, B. (1988). Computerization and women's knowledge. In *Women, work, and computerization: IFIP conference proceedings.* Amsterdam: Elsevier.

Thoresen, K. (1993). Principles in practice: Two cases of situated participatory design. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 271-287). Hillsdale, NJ, USA: Erlbaum.

URCOT. (1993). *Systems development in the ATO: Staff participation in the HURON pilot* (Project briefing). Melbourne, Australia: Union Research Centre on Organisational Technology Ltd. (URCOT).

Warfield, J. N. (1971). *Societal systems: Planning, policy, and complexity.* New York: Wiley.

Wixon, D., Holtzblatt, K., and Knox, S. (1990). Contextual design: An emergent view of system design. *Proceedings of CHI '90*, 329–336.

11.10 Appendix: Summaries of Participatory Practices

11.10.1 ACE (Amsterdam Conversation Environment)

Abstract: Low-tech simulation/enactment of e-mail communication patterns to support conversation among participants, by using paper airplanes as a vehicle for discussion.

Object Model: Notepad, pencils and paper airplanes to carry notes. Streamers stapled to the planes carry the semantics of the message openness: If the streamer is exposed it is traceable; if it is tucked in it is anonymous.

Process Model: Series of face-to-face meetings, where participants discuss the proposed system by: (1) Writing notes. (2) Sending notes. (3) Receiving notes. (4) Replying to notes.

Participation Model: Users, developers, testers.

Results: (1) Understanding of communication patterns. (2) Envisionment of a system to support these.

Phases of the Lifecycle: Requirements, analysis.

Complementary Formal Methods: Prototyping.

Group Sizes: Designed to support 21 people, can be used for groups greater than 2 people.

References:

Dykstra, E. A., and Carasik, R. P. (1991). Structure and support in cooperative environments: The Amsterdam Conversation Environment. *International Journal of Man-Machine Studies, 34*, 419–434.

11.10.2 ACOST Project

Abstract: Technique used in groupware environment to let people generate ideas and vote anonymously during requirements, prototyping and evaluation.

Object Model: Group decision support system environment.

Process Model: Discussion takes place through a groupware environment. Stages include: (1) Analysis of current situation in terms of problem and possible solutions. (2) Group requirements identification process, including identification of critical success factors, activities, processes, information requirements, data

elements, and so on. (3) Prototype including user comments. (4) Piloting of the prototype. (5) Evaluation. (6) Rollout.

Participation Model: Process facilitator, developers, users, key players and "technical chauffeurs" (some of these may participate anonymously).

Results: (1) Prioritized list of critical success factors. (2) Information requirements with data elements and functions. (3) Prioritized list of modifications needed for each iteration.

Phases of the Lifecycle: Problem identification, requirements, evaluation.

Complementary Formal Methods: None known.

Group Sizes: 8–15.

References:

Coleman, D. (Ed.). (1994). *Proceedings of GroupWare '94 Europe.* San Mateo, CA, USA: Morgan Kauffman.

De Vreede, G. J., and Sol, H. G. (1994). Combating organized crime with groupware: Facilitating user involvement in information system development. In D. Coleman (Ed.), *Proceedings of GroupWare '94 Europe.* San Mateo, CA, USA: Morgan Kauffman.

11.10.3 Artifact Walkthrough

Abstract: Users utilize artifacts from their work environment to reconstruct and review a specific example of their work process.

Object Model: Artifacts from the users' environment, such as documents and tools. A large wall display (whiteboard, blackboard, brown paper) on which the facilitator can write.

Process Model: Participants, as a group, retell a narrative of a specific example of a work process. A model of that work process is constructed "on the wall" by the facilitator. "Focus areas," such as time to complete steps and information needed, are called out as the model is constructed. The constructed model is cleaned up by the facilitators, and reviewed using additional scenarios by the users. Qualifications and extensions are noted at the time.

Participation Model: End-users (any stakeholder in the work process being described), human factors worker as facilitator, human factors worker as recorder, developers, managers.

Results: A process or work flow diagram with descriptions of scenarios of work and conclusions about that work (e.g., usability goals, general design objectives, constraints).

Phases of the Lifecycle: Requirements, analysis, high-level design, detailed design (provides a basis for prototype construction and testing).

Complementary Formal Methods: Task analysis, use case analysis, cognitive walkthrough, Joint Application Design, prototype testing, usability engineering.

Group Sizes: 6–8.

References:

Wixon, D. R., Pietras, C., Huntwork, P., and Muzzey, D. (in press). Changing the rules: A pragmatic approach to product development. In D. Wixon and J. Ramey (Eds.), *Field methods casebook for software design.* NY: Wiley.

Wixon, D. R., and Raven, M. B. (1994, April). *Contextual inquiry: Grounding your design in user work.* Tutorial at CHI '94 conference, Boston.

11.10.4 Blueprint Mapping

Abstract: Onto a large map of the workplace (the blueprint), participants attach pictures of work sites, work tools, and workers. The acts of placing these representations serve as a series of triggers for participatory analysis.

Object Model: Large map of the workplace, attached to a display surface. Photographs or drawings of work sites, tools, and people. Post-it[4] notes as annotations or as surrogates for illustrations that were not prepared ahead of time.

Process Model: Ultimately, the plan is for workplace participants to place the small representations onto the large map, and to discuss the work that is done at each site on the large map (the "current situation") as they perform the placements. Subsequently, the same or different participants review the current situation by talking about how the work is performed at the illustrated sites on the map. In practice, it may be useful for the facilitators to do some initial placing of the small illustrations.

Participation Model: People from the workplace who

[4] Post-it is a trademark of 3M Corporation.

carry out the daily work activities.

Results: Map with illustrations placed at "correct" sites. Understandings of how each group perceives the work to be organized. Recognition of diverse viewpoints. Discussions of both ordinary, every-day work flows and exceptions.

Phases of the Lifecycle: Analysis.

Complementary Formal Methods: Requirements analysis. Tracking of persons or artifacts through a task flow.

Group Sizes: Small, but not specified.

References:

Klær, A., and Madsen, K. H. (1995). Participatory analysis of flexibility. *Communications of the ACM, 38*(5), 53–60.

11.10.5 BrainDraw

Abstract: Graphical round-robin brainstorming[5] for rapidly populating a space of interface designs.

Object Model: Paper and pens. These may be arranged in a series of drawing stations (e.g., easels placed in a circle), or they may be assigned to seated team members.

Process Model: Each participant draws an initial design at that participant's initial drawing station. At the end of a pre-stated time interval, each participant moves left to the next drawing station, and continues the design found there. At the end of a pre-stated time interval, each participant again moves left to the next drawing station and again continues the design found there. The process continues until the participants are satisfied that they have worked with one another's' ideas.

Alternatively, the participants may stay still, the designs rotating from participant to participant.

Participation Model: Users, designers, artists.

Results: Many candidate designs. Each design has received contributions from many or all of the team members. Thus, each design is potentially a fusion of the participants' ideas. However, each design has a different starting point, so the fusions are not necessarily identical.

Phases of the Lifecycle: Detailed design.

Complementary Formal Methods: None known.

Group Sizes: 2–8, perhaps more.

References:

Dray, S. M. (1992, October). Understanding and supporting successful group work in software design: Lessons from IDS [Position paper]. In J. Karat and J. Bennett, (Chairs), *Understanding and supporting successful group work in software design.* Workshop at CSCW '92 conference, Toronto.

See also the entry on *Group Elicitation Method.*

11.10.6 Buttons Project

Abstract: For the support of shared end-user customizations, a software system supports end-user design (and automatic implementation) of customized functions. Each design takes the form of a button. End-users may share their customizations by sending buttons to one another.

Object Model: Experimental software system.

Process Model: End-users use a template-based approach to specify functionality in buttons. End-users mail buttons to one another. Recipients of buttons may modify them further. A social intervention in workplace culture may be required before certain end-user constituencies are willing to take the initiative to perform customizations.

Participation Model: End-users, by themselves.

Results: New functionality, shared among end-users. A software record of the innovations, in the form of executable customizations.

Phases of the Lifecycle: End-user customization.

Complementary Formal Methods: None known.

[5]There are cultural differences in the ways in which brainstorming is conducted. In our experience, the largest cultural different occurs between academia and industry, at least in the US. In corporate cultures, and perhaps in Scandinavia in general, brainstorming is understood to be the free contribution of ideas, without any critique or evaluation (those come in subsequent activities). In US academia, brainstorming is often understood to include critique by other participants, and/or evaluation of the contributed ideas by the facilitator. All of our references to brainstorming in this paper are intended to follow the model of free contribution without fear of critique or evaluation, and with the assumptions that the group will commonly own *all* of the ideas from the brainstorming activity, and will conduct a critical evaluation of those ideas *as a group* during a subsequent activity.

Group Sizes: Typically one end-user works alone, then shares the resulting customizations with any number of other end-users.

References:

MacLean, A., Carter, K., Lovstrand, L., and Moran, T. (1990). User-tailorable systems: Pressing the issues with buttons. *Proceedings of CHI '90*, 175–182.

11.10.7 CARD (Collaborative Analysis of Requirements and Design)

Abstract: Participants use a deck of cards to lay out or critique task flows. The cards represent work components, including computer-based functionality, non-computer events and objects, cognitions, motivations, goals, and people.

Object Model: Cards representing work components, often arranged in a class hierarchy of components. A background sheet of paper onto which the cards can be attached, and onto which flow arrows, choice points, decision criteria, and so on, can be written.

Process Model: (1) Introductions that include each participant's expertise, contribution to the shared work, organization and constituency, and what that constituency is expecting from the participant (i.e., personal and organizational stakes). (2) Mutual education through "mini-tutorials," if needed. (3) Working together to explore the task domain (in the form of analysis, design, or assessment). (4) Brief walkthrough of the group's achievements during the session, preferably recorded on videotape.

Participation Model: End-users and one or more of the following: human factors workers, software professionals, marketers, technical writers, trainers, perhaps clients or customers of the end-users, other stakeholders in the system.

Results: Representations of task flows, recorded in sequences of cards, plus annotations and (optionally) task flow arrows, branch points, decision criteria, and so on, recorded on the background sheet.

Phases of the Lifecycle: Analysis, design, evaluation.

Complementary Formal Methods: Object-oriented analysis and modeling.

Group Sizes: Up to 8.

References:

Muller, M. J., and Carr, R. (1996). Using the CARD and PICTIVE participatory design methods for collaborative analysis. In D. Wixon and J. Ramey (Eds.), *Field methods casebook for software design*. NY: Wiley.

Muller, M.J., Tudor, L.G., Wildman, D.M., White, E.A., Root, R.W., Dayton, T., Carr, R., Diekmann, B., and Dykstra-Erickson, E.A. (1995). Bifocal tools for scenarios and representations in participatory activities with users. In J. Carroll (Ed.), *Scenario-based design for human-computer interaction*. New York: Wiley.

Tudor, L. G., Muller, M. J., Dayton, T., and Root, R. W. (1993). A participatory design technique for high-level task analysis, critique, and redesign: The CARD method. *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, 295–299.

See also the entries on *CUTA*, *Layout Kit*, *Metaphors Game*, and *PictureCARD*.

11.10.8 CESD (Cooperative Experimental System Development)

Abstract: CESD provides a framework in which to conduct a participatory lifecycle (however, it does not provide a model as such).

Object Model: Prototypes and prototyping are the most salient attributes of the work. Other Scandinavian techniques, such as workshops and Mock-Ups , may also be used.

Process Model: CESD provides a process "in the large"—at the level of the software lifecycle. At this level of granularity, there are two major differences from conventional lifecycle approaches: End-users participate in most or all phases of the work, and implementation (or realization) activities begin much earlier in the lifecycle, and are much more open to end-user participation).

Participation Model: End-users, software professionals.

Results: Completed systems, deeply informed by end-user and union concerns and perspectives.

Phases of the Lifecycle: CESD is a lifecycle approach, with application to all phases of the software lifecycle.

Complementary Formal Methods: Other software lifecycle models.

Group Sizes: Variable by activity.

References:

Grønbæk, K., Kyng, M., and Mogensen, P. (1995). Cooperative experimental System development: Cooperative techniques beyond initial design and analysis. In *Conference proceedings of Computers in Context: Joining Forces in Design* (pp. 20–29). Århus, Denmark: Aarhus University.

See also:

Grønbæk, K., and Mogensen, P. (1994). Specific cooperative analysis and design in general hypermedia development. *PDC '94: Proceedings of the Participatory Design Conference*, 159–171.

See also the entries on *Mock-Ups* and *Prototyping*.

11.10.9 CISP (Cooperative Interactive Storyboard Prototyping)

Abstract: A small team of developer(s) and user(s) cooperatively generate and modify user interface designs, evaluate existing interfaces, and compare interface alternatives, sometimes using computerized tools.

Object Model: CISP software tool, or HyperCard[6].

Process Model: Iterations of the following steps: (1) Explore storyboard (the user does the task, the interface tool records the user's steps). (2) Evaluate storyboard (play back the storyboard's record of the user's activities, and discuss within the user-developer team). (3) Modify storyboard.

Participation Model: One or a few users and one or a few developers.

Results: Enhanced storyboard or prototype; recordings of users' interactions with the storyboard or prototype.

Phases of the Lifecycle: Detailed design, assessment/evaluation.

Complementary Formal Methods: Design, usability inspection.

Group Sizes: 2–4.

References:

Madsen, K. H., and Aiken, P. J. (1992). Cooperative Interactive Storyboard Prototyping: Designing friendlier VCRs. In S. J. Andriole (Ed.), *Rapid application prototyping: The storyboard approach to user requirements analysis* (2nd ed., pp. 261–233). Boston: QED.

Madsen, K. H., and Aiken, P. (1993). Experiences using cooperative interactive storyboard prototyping. *Communications of the ACM, 36*(6), 57–64.

See also the entry on *Storyboard Prototyping*.

11.10.10 Codevelopment

Abstract: End-users and software professionals form a long-term working relationship to share the responsibilities for specifying and implementing a custom software system for the end-users and their organization. End-users collaborate at nearly every phase of the project.

Object Model: General meeting facilitation materials. Mock-ups. Prototypes. A large Wall area that holds artifacts (mostly paper) of varying formality (from scribbled notes, drawing, and photographs to finished reports) arranged along the project's timeline.

Process Model: Complex. Some work is done at the end-users' work site. Some work is done at the software professionals' work site. Specific processes within work sessions are tailored to the needs of each session.

Participation Model: End-users and software professionals.

Results: Working system. Formal documents. Informal, graphical, contextualized record of the project and its rationale in the form of the Wall.

Phases of the Lifecycle: This is a model for a participatory lifecycle. It spans all phases of the lifecycle.

Complementary Formal Methods: Other software lifecycle models.

Group Sizes: Perhaps as many as 20.

References:

Anderson, W. L. (1994). The wall: An artifact of design, development, and history. *PDC '94: Proceedings of the Participatory Design Conference*, 117.

Anderson, W. L., and Crocca, W. T. (1993). Engineering practice and codevelopment of product prototypes. *Communications of the ACM, 36*(6), 49–56.

[6] HyperCard is a trademark of Apple Computer.

11.10.11 Collaborative Design Workshops

Abstract: Scenarios are combined with low-fidelity prototypes for a task-centered walkthrough.

Object Model: Low fidelity paper prototypes; videotaping equipment.

Process Model: Guided by a contextualized work scenario, two users discuss work practices, variations, and alternatives while manipulating the low-fidelity prototype. The users' work with the materials (but not their faces) plus additional group discussions are recorded on video.

Participation Model: "Key players" — for example, two users, a designer, and a domain expert; a developer, a documenter, and a requirements analyst may be included upon need.

Results: The evolving paper mock-up and informal notes, sometimes formalized as a "design memo" or an illustrated scenario.

Phases of the Lifecycle: Analysis and design.

Complementary Formal Methods: None known.

Group Sizes: Small — four people in the most representative case.

References:

Bloomer, S., Croft, R., and Wright, L. (in press). Collaborative design workshops: A case study. To appear in *interactions* magazine.

See also *Artifact Walkthrough*, *Hiser Method*, *PICTIVE*, *Pluralistic Walkthrough*, and *Scenarios*.

11.10.12 Conceptual Toolkit in CSCW Design

Abstract: The toolkit supports communication during design-by-doing, through the use of checklists and scenarios.

Object Model: Scenarios of work. Artifacts from work. Checklists to support analysis and communication.

Process Model: Participants discuss work situations described in scenarios, using role-specific checklists (e.g., a work-oriented checklist and a technical checklist) to support and extend communications across different perspectives. This process occurs within a broader participatory scheme, including workshops and prototyping.

Participation Model: End-users. Software professionals.

Results: Annotated scenarios that particularize problems to be solved.

Phases of the Lifecycle: Requirements, analysis, high-level design.

Complementary Formal Methods: Requirements analysis, task analysis.

Group Sizes: Small.

References:

Bødker, S., Christiansen, E., and Thüring, M. (1995). A conceptual toolbox for designing CSCW applications. In *COOP '95: Atelier international sur la conception des systèms coopératifs* [International workshop on the design of cooperative systems] (pp. 266–284). Sophia Antiipolis, France: INRIA.

See also the entries on *Prototyping* and *Scenarios*.

11.10.13 Contextual Design

Abstract: Contextual *design* uses contextual *inquiry* as its first step, to gather user data. Those data are analyzed in a teamwork approach by a team that appears to consist, for the most part, of members other than end-users, to produce a user interface design. Steps along the way focus on analyzing the work flow aspects of the user data.

Object Model: Room dedicated to the project, viewgraphs shown on its wall, and its walls and table covered with flip chart paper, diagrams, and Post-it notes; paper prototypes of the user interface.

Process Model: Users are interviewed with contextual inquiry, by the product designers, at the users' work sites. The resulting user data are utilized in the forms of the product designers' understanding and notes, to inform the designers and other relevant personnel (excluding users) as they go through several specific steps to analyze the user work flow and produce an appropriate user interface design. All these post-interview activities are done by the entire team, together, in the dedicated project room. Several teamwork methods are used, such as brainstorming and group memory in the form of paper notes on walls. The understandings may then be structured in a bottom-up process to put the information into conceptual groupings to form an affinity diagram.

Participation Model: Users, product designer, us-

ability engineer, developer, system engineer, product manager. Users participate in the initial contextual inquiry step, but may be involved as co-designers only in the limited sense of responding to prototypes that the design team has created (Holtzblatt and Beyer, 1993).

Results: User interface design, prototyped in paper.

Phases of the Lifecycle: Requirements, analysis, high-level design.

Complementary Formal Methods: Writing of formal requirements documents to put the design into a format more suitable for system engineers, developers, and testers.

Group Sizes: Up to 10.

References:

Holtzblatt, K., and Beyer, H. (1993). Making customer-centered design work for teams. *Communications of the ACM, 36*(10), 93–103.

Whiteside, J., Bennett, J., and Holtzblatt, K. (1988). Usability engineering: Our experience and evolution. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 791–817). New York: North-Holland.

Wixon, D., Holtzblatt, K., and Knox, S. (1990). Contextual design: An emergent view of system design. *Proceedings of CHI '90*, 329–336.

See also the entry on *Contextual Inquiry*.

11.10.14 Contextual Inquiry

Abstract: The ethnographically-based contextual inquiry gets data from users by having product designers observe and interview users in their workplace, as they do their work. This is used as the first step in the method called contextual *design*.

Object Model: Users use all the artifacts they normally use to do their work. Interviewers take private notes, though they may show and explain the notes to users. Video recordings of people at work may also be used.

Process Model: Interviewers observe users doing their real work in their real workplace. Interviewers have permission to interrupt users at any time to ask questions, and to converse about the work. Interviewers also ask users' opinions on the interviewers' design ideas, and may ask users for their ideas for improvements in their work and current interface. An affinity diagram may be used to organize the findings[7].

Participation Model: Users and interviewers, the latter preferably being the product's designers. "Designers" can be anyone on the cross-functional team, such as a usability engineer, developer, system engineer, or product manager.

Results: The product's designers get detailed understanding and notes of the users' work and needs, fully contextualized in the real work situation. The resulting user data can be input to a design process.

Phases of the Lifecycle: Problem identification, requirements, analysis, some high-level design.

Complementary Formal Methods: Preliminary survey to identify appropriate samples of users and work.

Group Sizes: Usually 2; perhaps as many as 4.

References:

Bennett, J., Holtzblatt, K., Jones, S., and Wixon, D. (1990, April). *Usability engineering: Using contextual inquiry*. Tutorial presented at CHI '90, Seattle, WA.

Holtzblatt, K., and Jones, S. (1993). Contextual inquiry: A participatory technique for system design. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 177–210). Hillsdale, NJ, USA: Erlbaum.

Wixon, D. R., and Comstock, E. M. (1994). Evolution of usability at Digital Equipment Corporation. In M. E. Wiklund (Ed.), *Usability in practice: How companies develop user-friendly products* (pp. 147–193). Boston: AP Professional.

See also the entry on *Contextual Design*.

11.10.15 Cooperative Evaluation

Abstract: An evaluation team is formed of one end-user and one developer. Together, they explore a prototype and develop a critique.

Object Model: Software system or prototype, using the *York Manual* to guide the analysis.

Process Model: Elaborate—provided in the *York Manual*.

Participation Model: One end-user and one developer.

---

[7]Holtzblatt and Beyer (1993) excluded from the label "Contextual Inquiry" all activities after the actual field interview. However, Holtzblatt and Jones (1993) discussed affinity diagramming and related analysis as part of contextual inquiry.

Results: Critique of prototype or system.

Phases of the Lifecycle: Assessment.

Complementary Formal Methods: Usability testing, discount usability evaluations.

Group Sizes: 2.

References:

Wright, P., and Monk, A. (1991). A cost- effective evaluation method for use by designers. *International Journal of Man-Machine Studies, 35*(6), 891–912.

11.10.16 Cooperative Requirements Capture

Abstract: Informed by a broad analysis of who are the stakeholders (or interested parties) in a computer system design problem, software professionals enter a six-step process to determine the requirements for that design. Three steps involve direct work with users as active participants. Four other steps provide context within the software professionals' lifecycle.

Object Model: None. The work involves a series of workshops with users and other stakeholders.

Process Model: Six steps, really seven. (0) The pre-process step is an analysis of the stakeholders, in terms of four broad categories that are described below under "participation model." (1) Identify the business problem that needs to be solved; no direct user participation occurs in this step. (2) Formulate the team from among the four classes of stakeholders described in the "participation model," below (no direct user participation occurs in this step). (3) Explore the user environment through a User Workshop. (4) Validate understanding of user environment, *perhaps* through consultations with the users, but *perhaps* through market research or the team's confidence in what it already knows. (5) Identify scope of proposed system, including usability goals, through a User Workshop. (6) Validate scope with stakeholders via user participation in one of various venues (e.g., focus group, survey, interview, mock-up).

Participation Model: Four stakeholder groups are considered in the selection of representatives: (1) Software professionals who are responsible for design and implementation. (2) Business and marketing analysts with a financial concern for the design and system. (3) Managerial and support staff responsible for introduction and maintenance of the system. (4) Users, who should be drawn from three categories: primary (frequent users), secondary (occasional users), and tertiary (people affected by the system or its purchase, but who are not direct users of it).

Results: Paper documents, including a User Document after Step 3, "Explore the user environment," and an Initial Requirements Document after Step 5, "Identify scope of proposed system."

Phases of the Lifecycle: Requirements, analysis.

Complementary Formal Methods: This approach includes formal methods—or in any event, methods for software professionals only—*within* its six steps.

Group Sizes: 6–8.

References:

Macaulay, L. (1995). Cooperation in understanding user needs and requirements. *Computer Integrated Manufacturing Systems, 8*(2), 155–165.

11.10.17 Critics to Support End-User Customization

Abstract: A software system provides means for customization by end-users. Through specialized software entities called "critics," the system inspects the end-users' customizations for unusual properties, such as apparent violations of design rules. When a critic encounters an unusual customization, it queries the end-users who originated those changes, and records their rationale for later use by software professionals. Subsequent redesign of the system is guided in part by end-users' customizations and by their recorded rationales for those customizations. Critics may also be activated during software professionals' design and redesign activities.

Object Model: Experimental software system.

Process Model: Here is one version, which is more a scenario than a formal process model: (1) Requirements are established through an unspecified participatory process. (2) Software professionals create initial user interface design. (3) Software critics inspect the initial design according to standard design heuristics, noting apparent violations of those heuristics. Software professionals may record their rationales for their design decisions, or may modify their designs. (4) End-users engage in an unspecified collaborative activity to select system attributes for customization, and use customization capabilities that are part of the system to accomplish these customizations. Software critics inspect the end-users' customizations according to standard design heuristics, noting apparent violations of

those heuristics. End-users may record their rationales for their customization decisions, or may modify their customizations. (5) During the next iteration of the lifecycle, software professionals consult the end-users' customizations and their recorded rationales as part of their redesign activities.

Participation Model: During development: End-users and software professionals. During usage: End-users. During redesign: Software professionals and end-users.

Results: End-user-initiated customizations to the working system. Recorded rationales for all unusual customizations and design features.

Phases of the Lifecycle: Design. Customization in the field. Redesign.

Complementary Formal Methods: None known.

Group Sizes: Unspecified.

References:

Malinowski, U., and Nakakoji, K. (1995). Using computational critics to facilitate long-term collaboration in user interface design. *Proceedings of CHI '95,* 385–392.

11.10.18 CUTA (Collaborative Users' Task Analysis)

Abstract: CUTA is a variation on the CARD method. Participants use cards to lay out a task analysis. The cards consist largely of photographs of actual members of the workplace conducting work tasks, plus photographs of objects and items in the workplace, plus more abstract "wild card" drawings that represent less specific instances of work. Each card also contains a task analysis data template.

Object Model: Cards. Each card contains a photograph of an actual worker (who is known to the participants) or a workplace object or item (which is familiar to the participants). More abstract cards may contain drawings of categories of events, such as temporal events, meetings, and a person thinking. Each card contains a template that requests the number of this card's component in the task sequence, the duration of the activity represented by this card, and the frequency with which the action is done (e.g., "once per participant").

Process Model: Participants lay out the cards, filling in the required template information. Repeating sequences or other non-linear flows may be color coded.

Participation Model: End-users and software professionals.

Results: Cards as documentation of task flows. Estimates of task durations and frequencies of task occurrences.

Phases of the Lifecycle: Analysis.

Complementary Formal Methods: More formal task analyses.

Group Sizes: Up to 6.

References:

Lafreniere, D. (1995). CUTA: A simple, practical, and low-cost approach to task analysis. To appear in *interactions.*

See also the entries on *CARD, Layout Kit, metaphors Game,* and *PictureCARD.*

11.10.19 Diaries

Abstract: Diaries are maintained by the design group as a record and point of access for users and others who wish to understand the design rationale.

Object Model: Various system documents and system versions. On-line diaries.

Process Model: Informal notes are kept of design decisions. They are made accessible in an on-line system.

Participation Model: Designers and users.

Results: Improved understanding. Decisions recorded in diaries.

Phases of the Lifecycle: Potentially, throughout design and implementation.

Complementary Formal Methods: Other design rationale techniques.

Group Sizes: Unclear.

References:

Braa, K. (1992). Influencing system quality by using decision diaries in prototyping projects. *PDC '92: Proceedings of the Participatory Design Conference,* 163–170.

Kautz, K. (1992). Communications support for participatory design projects. *PDC '92: Proceedings of the Participatory Design Conference,* 155–162.

11.10.20 ETHICS (Effective Technical and Human Implementation of Computer-based Systems)

Abstract: Participatory systems development method that balances social and technical aspects to ensure an optimized system. This is a participatory lifecycle approach.

Object Model: Paper, flip charts, pens.

Process Model: (1) Describe current work situation. (2) Clarify business mission. (3) Identify problems in present work organization that are reducing the efficiency. (4) Set human objectives for new system (e.g., job satisfaction and quality of working life). (5) Set business objectives for new system (e.g., efficiency). (6) Consider changes that are likely occur and that the system will have to accommodate. (7) Consider alternatives for the system in terms of social system (e.g., job design and work organization) and technical/administrative system (e.g., hardware, software, work procedures, information flow). (8) Set objectives for the new system by matching objectives for human and business, and considering alternatives, problem areas, and likely changes. (9) Redesign organizational structure. (10) Choose technical system. (11) Prototype organizational and technical systems. (12) Change job design to match new system. Each job should represent an enhancement of the previous work and take account of the challenge, responsibility, autonomy, and so on. (13) Implement the system in the workplace. (14) Evaluate the system in terms of the human and technical/business sides. For the human side, job satisfaction of the users should be compared to the level before the new system was introduced, in terms of the knowledge, psychological, efficiency, task structure, and ethical fit. For the business side, the number of problems that have been controlled or eliminated should be identified.

Participation Model: Facilitator; representatives from user group, development, and management.

Results: (1) New or modified computer system. (2) Modified work organization that fits computer system. (3) List of changes that are likely to occur during life of system, and which can be accommodated by the new or modified system. (4) List of variances to be used to help increase efficiency and minimize problems. (5) Questionnaire to assess job satisfaction before and after introduction of system.

Phases of the Lifecycle: Entire lifecycle: requirements, analysis, high-level design, and evaluation.

Complementary Formal Methods: Prototyping. More broadly, other software lifecycle models.

Group Sizes: Up to 40.

References:

Mumford, E. (1983). *Designing human systems for new technology: The ETHICS method.* Manchester, UK: Manchester Business School.

Mumford, E. (1991). Participation in systems design—What can it offer? In B. Shackel and S. J. Richardson (Eds.), *Human factors for informatics usability* (pp. 267–290). Cambridge: Cambridge U. Press.

Mumford, E. (1993). The participation of users in systems design: An account of the origin, evolution, and use of the ETHICS method. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 257–270). Hillsdale, NJ, USA: Erlbaum.

Mumford, E. and Weir, M. (1979): *Computer systems in work design—The ETHICS method: Effective technical and human implementation of computer systems.* New York: Wiley.

11.10.21 Ethnographic Practices

Abstract: We include ethnographic practices in this chapter on a very tentative basis. Ethnographic practices have been highly influential in participatory design. However, many practitioners have stated quite clearly that these are not simple techniques that can be picked up and used by anyone. Rather, ethnography requires extensive training in both specific practices, and more importantly, in the perspectives and disciplines that underlie those practices. In this sense, the word fragment "methodology" in "ethnomethodology" should be understood as being *different in kind* from more conventional engineering methods[8]. Thus, this item in the taxonomy is intended as an access point, to help potential practitioners find descriptions of ethnographic work in participatory design, and to help them begin to appreciate the complex requirements of this type of work.

[8]In general, participatory "methods" are different from engineering "methods" along a number of dimensions. Ethnomethodology is perhaps the furthest of all participatory practices from an engineering "method."

Object Model: The users' workplace. Representations of the end-users' work, including (but not limited to) video recordings, where possible, of the end-users' work.

Process Model: A variety of ethnographic practices that are difficult to summarize in the telegraphic style required for our chapter.

Participation Model: End-users. Ethnographers. Possibly software professionals, facilitated by ethnographers.

Results: A detailed description of the end-users' work.

Phases of the Lifecycle: Requirements, analysis.

Complementary Formal Methods: Requirements, analysis.

Group Sizes: Generally small. Varies by ethnographer.

References:

Blomberg, J., Giacomi, J., Mosher, A., and Swenton-Wall, P. (1993). Ethnographic field methods and their relation to design. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices.* Hillsdale, NJ, USA: Erlbaum.

Nardi, B. (1995). Some reflections on scenarios. In J. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development.* New York: Wiley.

Rose, A., Shneiderman, B., and Plaisant, C. (1995). An applied ethnographic method for redesigning user interfaces. In *Proceedings of DIS '95* (pp. 115–122). New York: ACM.

See also:

Blomberg, J., Suchman, L., and Trigg, R. (1994). Reflections on a work-oriented design project. *PDC '94: Proceedings of the Participatory Design Conference,* 99–109.

Blomberg, J., Suchman, L., and Trigg, R. (1995). Back to work: Renewing old agendas for cooperative design. In *Conference proceedings of Computers in Context: Joining Forces in Design* (pp. 1–9). Århus, Denmark: Aarhus University.

Hughes, J., King, V., Rodden, T., and Anderson, H. (1995). The role of ethnography in interactive systems design. *interactions, 2*(2), 56–65.

Hughes, J., Randall, D., and Shapiro, D. (1992). Faltering from ethnography to design. *Proceedings of CSCW '92,* 115–122.

McClard, A. P. (1995). Borderlands: Ethnography in the technical industries. *Proceedings of the American Anthropological Association 1995 Annual Meeting.*

Orr, J. and Crowfoot, N. C. (1992). Design by anecdote—The use of ethnography to guide the application of technology to practice. *PDC '92: Proceedings of the Participatory Design Conference,* 31–37.

Suchman, L. (Ed.). (1995). Representations of work [Special issue]. *Communications of the ACM, 38*(9).

Suchman, L., and Trigg, R. (1991). Understanding practice: Video as a medium for reflection and design. In J. Greenbaum and M. Kyng (Eds.), *Design at work: Cooperative design of computer systems.* Hillsdale, NJ, USA: Erlbaum.

Wall, P., and Mosher, A. (1994). Representations of work: Bringing designers and users together. *PDC '94: Proceedings of the Participatory Design Conference,* 87–98.

See also chapter 15 of this handbook.

11.10.22 FIRE (Functional Integration through Redesign)

Abstract: Organizational principles and techniques for continuous redesign of computer-based systems that are conceived to be integrated with work and other technologies.

Object Model: None.

Process Model: System versions are redesigned in a planned and organized way, including user participation at specified points in the process.

Participation Model: All users at all levels (including management) and all development staff.

Results: Redesign decisions and suggestions.

Phases of the Lifecycle: Problem identification, requirements, analysis, high-level design, redesign.

Complementary Formal Methods: SSM. Object-oriented analysis and design.

Group Sizes: About 20.

References: A series of reports has been issued by the FIRE project. Contact FIRE, Department of Informatics, University of Oslo, P.O. Box 1080 Blindern, N-0316 Oslo, Norway, +47 2 85 24 10 (voice), +47 2 85

24 01 (fax), fire@ifi.uio.no.

11.10.23 Florence Project

Abstract: This project concentrated on communication with the users during the development process to find out what sort of computer system, if any, the users need.

Object Model: Paper and pens to record observations; materials for electronic prototyping.

Process Model: (1) Observe users in their place of work to gain understanding of current work procedures. (2) Develop prototypes to try out ideas and clear up misconceptions.

Participation Model: Users and computer scientists.

Results: (1) Mutual learning: Computer scientists learn about the work of the users; users learn about computers in relation to their work. (2) Envisionment of the computer system, if any, that is needed.

Phases of the Lifecycle: Problem identification, requirements, high-level design.

Complementary Formal Methods: None known.

Group Sizes: Not specified.

References:

Bjerknes, G. and Bratteteig, T. (1987): Florence in Wonderland: System development with nurses. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 279–295). Brookfield, VT, USA: Gower.

11.10.24 Forum Theatre

Abstract: A troupe of actors (stakeholders) acts out a scenario with an undesirable outcome, as an informal theatrical production. The audience (other stakeholders) has the opportunity to change the script, after which the actors again act out the scenario. This entire process iterates until the outcome is more desirable.

Object Model: None. Forum theatre has its roots in street and guerrilla theatre. There are few props, and the script is rather conceptual.

Process Model: A group of active designers develops a description of a situation that, in their view, ought to be changed. They develop their description into a scenario. They perform this scenario for other stakeholders, and ask those stakeholders to find ways of modifying the events in the scenario so that the outcome is improved. The process is very spontaneous.

Participation Model: End-users and other stakeholders (loosely defined).

Results: Improved understanding. A sense of drama.

Phases of the Lifecycle: Problem identification and clarification, analysis, assessment.

Complementary Formal Methods: None known.

Group Sizes: Large.

References:

In the political domain:

Boal, A. (1992). *Games for actors and non-actors* (A. Jackson, Trans.). London: Routledge.

In the software domain, forum theatre was informally presented at an IRIS (Information systems Research symposium In Scandinavia) conference by Finn Kensing and Kim Halskov Madsen. Forum theatre was listed as a participatory practice in:

Kensing, F., and Munk-Madsen, A. (1993). PD: Structure in the toolbox. *Communications of the ACM, 36*(6), 78–85.

11.10.25 Future Workshop

Abstract: A three-part workshop addresses a critique of the present situation, a fantasy of an improved future situation, and the question of how to move from the critiqued situation to the fantasy situation.

Object Model: None. Some common meeting supports are useful (e.g., blackboards or whiteboards, Post-it notes).

Process Model:

- *Critique Phase:* Participants engage in structured brain-storming focused on the current work problems. Problems may be recorded on a wall-chart or other medium that is accessible to all participants. Participants then break into small groups. Each group works with a subset of the problem statements to develop a concise critique of a subset of issues in the current work situation. Specific facilitation techniques that may be useful in this phase include exploration of metaphors about the current work situation, and strict limits on the time for each comment by any participant.
- *Fantasy Phase:* Participants envision a future work

situation that is better than the present. Specific facilitation techniques that may be useful in this phase include inversion of the Critique Phase's negative statements into positive statements, drawing pictures of the envisioned future, multivoting[9] to select the most desirable future attributes, and (as in the Critique Phase) metaphors and strict limits on the time for each comment.

- *Implementation Phase:* Each small group presents a report on their envisioned future. The workshop conducts a plenary discussion to evaluate whether the envisioned future can be achieved under current circumstances. If not, what changes need to be made? How can those changes be planned?

Participation Model: End-users and other stakeholders, generally not software professionals. One or more facilitators.

Results: An implementation plan, including specific action items to be completed by specific persons.

Phases of the Lifecycle: Problem identification and clarification. Requirements. Perhaps other phases.

Complementary Formal Methods: None known.

Group Sizes: Medium to large.

References:

In German municipal planning, and for a detailed procedural account:

Jungk, R., and Mullert, N. (1987). *Future workshops: How to create a desirable future*. London: Institute of Social Invention.

In software design:

Kensing, F., and Madsen, K. H. (1991). Generating visions: Future workshops and metaphorical design. In J. Greenbaum and M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 155–168). Hillsdale, NJ, USA: Erlbaum.

See also:

Greenbaum, J., and Madsen, K. H. (1993). Small changes: Starting a participatory design process by giving participants a voice. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 289–298). Hillsdale, NJ, USA: Erlbaum.

9 In multivoting, each participant has a number of votes that can be cast in any arrangement—that is, each vote on a different item, or all votes on a single item, or any other assignment. The group casts their votes on a number of topics are typically candidates for future, more focused work. The topics with the most votes are selected.

11.10.26 Graphical Facilitation

Abstract: A facilitator aids a group in clarifying its analysis or design (or other shared purpose) by reflecting the group's words and ideas back to the group in the form of quickly-drawn graphical images.

Object Model: Flip charts, blackboards, whiteboards, colored markers, chalk.

Process Model: Facilitator listens, sketches, queries the group for clarification.

Participation Model: One or two facilitators work with a group of project stakeholders.

Results: Graphical images are the only physical artifact that is produced. The major "product" is the enhancement of shared understanding among the stakeholders, and their movement toward a shared purpose.

Phases of the Lifecycle: (1) Problem identification. (2) Requirements, analysis, high-level design.

Complementary Formal Methods: None known.

Group Sizes: Up to 40.

References:

Crane, D. (1990). *Graphic recording in systems design*. Workshop at PDC '90: Conference on Participatory Design, Seattle, WA, USA.

Sibbet, D., Drexler, A., et al. (1993). *Graphic guide to team performance: Principles/Practices*. San Francisco: Grove Consultants International.

11.10.27 Group Elicitation Method

Abstract: A six phase workshop provides decision support to a method for sharing ideas in a brainstorming format.

Object Model: Nothing specific.

Process Model: (1) Issue statement and formulation: State the problem clearly and recruit relevant domain experts (see Participation Model, below). (2) Viewpoints generation (round-robin brainstorming, or "brainwriting"): Participants write ideas on a sheet of paper, and then pass the sheet to the next participant. Upon receiving another participant's ideas, each participant may then agree, disagree, or state a new viewpoint. This phase continues until all participants have

seen all other participants' ideas once. (3) Reformulation into more elaborated concepts: The group reduces the large list of ideas to a smaller number of more central concepts. (4) Generation of relationships among the concepts: Each participant completes a cross-product table, stating that participant's perception of the relative importances (greater, equal, lesser) among all pairs of concepts. (5) A "consensus" is calculated from the relative importance pair ratings. (6) Critical analysis of the results: The group considers the calculated results, and modifies them as necessary.

Participation Model: End-users and perhaps designers. A facilitator.

Results: Lists of ideas. Lists of concepts (refinements and combinations of ideas). Importance ranking of concepts. Critical analysis of importance ranking.

Phases of the Lifecycle: Problem identification and clarification. Requirements.

Complementary Formal Methods: None known.

Group Sizes: Up to 7.

References:

Boy, G. A. (in press). The group elicitation method for participatory design and usability testing. To appear in *interactions*.

See also:

Boy, G. A. (1991). *Intelligent assistant system.* London: Academic Press.

Warfield, J. N. (1971). *Societal systems: Planning, policy, and complexity.* New York: Wiley.

See also the entry on *BrainDraw*.

11.10.28 Hiser Design Method

Abstract: User interface design method that incorporates aspects of HCI to develop usable systems within real-world constraints.

Object Model: Papers and pens, normal office materials for paper prototyping and videotaping materials. Materials for electronic prototypes may also be used.

Process Model: Iterations of: (1) Analysis, which takes place through contextual field studies and informal evaluation of current systems. (2) Design, which takes place through collaborative design sessions, paper prototyping sessions, and electronic prototyping. (3) Evaluation, which is done through collaborative design and testing, heuristic evaluation, and usability testing.

Participation Model: Interface designer, analyst or user representative, and two or more users from different groups.

Results: (1) User profile document that gives information about the users, their activities, and their work environment. (2) Scenarios to envision system usage. (3) Design document. (4) Prototypes of system. (5) Style guide or user interface specification. (6) Usability goals and test findings.

Phases of the Lifecycle: User requirements, user interface design.

Complementary Formal Methods: None known.

Group Sizes: 3–4.

References:

Bloomer, S. (1993): Real projects don't need user interface designers: Overcoming the barriers to HCI in the real world. *Proceedings of OZCHI '93,* 94–108.

See also *Collaborative Design Workshop, Contextual Inquiry, PICTIVE,* and *Scenarios.*

11.10.29 HOOTD (Hierarchical Object-Oriented Task Decomposition)

Abstract: Participants decompose a task description into the information objects acted upon and the actions taken on them, and assign groups of those objects to interface windows.

Object Model: Index cards.

Process Model: All participants, in parallel, write each task (noun and verb) on its own index card. Duplicate cards are discarded. Then all the participants work as a group to sort the cards into piles, using whatever criteria the participants think appropriate. That clustering scheme is recorded, the participants resort the cards according to any other criteria, and finally choose one of the sorting schemes as the best. Each pile in that scheme becomes one task domain, with one of the eventual graphical user interface windows serving that task domain by containing all the objects and actions in that pile of task cards.

Participation Model: Users, usability engineer, system engineer, developer, subject matter experts in the relevant business processes and information systems, developers of documentation and training, managers of all these people.

Results: Definitions of all the user interface windows and the information objects contained in them.
Phases of the Lifecycle: Analysis, high-level design.

Complementary Formal Methods: Task analysis and design to ensure participants are aware of all the relevant tasks.

Group Sizes: Up to 8.

References: Contact Robert W. Root, Bellcore, RRC 1B-127, 444 Hoes Lane, Piscataway, NJ 08854, USA, +1-908-699-7763, broot@ctt.bellcore.com.

11.10.30 Icon Design Game

Abstract: One participant (the sketcher) draws informal icons while other participants attempt to guess at the concept that the sketcher is trying to express. The drawings become *first drafts* (not finished artwork) for the development of icons. The game can be played cooperatively (with a single team) or competitively (with multiple teams).

Object Model: Sketching surface, pens.

Process Model: (1) The sketcher selects a concept to attempt to communicate to the team. (2) The sketcher draws pictures of the concept. (3) The team attempts to guess the concept that the sketcher is trying to express. (4) Optionally, an observer takes notes on drawings that appeared to be particularly effective or particularly confusing. (5) The best drawings are delivered to a graphic artist or other professional for further development into polished, professionally designed icons.

Participation Model: End-users and one or more of the following: human factors workers, software professionals, marketers, technical writers, trainers, perhaps clients or customers of the end-users, other stakeholders in the system.

Results: Rough sketches of icons, for further development.

Phases of the Lifecycle: Detailed design.

Complementary Formal Methods: None known.

Group Sizes: Up to 20 people per team, though smaller teams probably work better.

References:

Muller, M. J., Wildman, D. M., and White, E. A. (1994, April). *Participatory design through games and other group exercises.* Tutorial at CHI '94 conference, Boston.

11.10.31 Interface Theatre

Abstract: Interface Theatre is an experimental practice to support a design walkthrough and critique by a very large group of end-users and other stakeholders. Working on a stage with human-scale theatrical props, the design team acts out the appearance and dynamics of the user interface and its system. Actors are guided by "object-oriented scripts" that describes the functionality of interface components. The audience of end-users and other stakeholders critiques the appearance and actions, transforming the design. The troupe of actors re-enacts the interface until they and the audience are satisfied.

Object Model: A stage. "Costumes" for the actors, in the form of human-scale interface objects (e.g., a 0.7 1 m cardboard dialog box). Object-oriented scripts, which specify the behavior of each interface object (as portrayed by an actor) in terms of the messages that might be sent to that role, the methods that the character (interface component) may execute in response to the messages, and any visible side-effects. "Behind the scenes" representations of system components without a direct representation in the user interface (e.g., buffers, databases, communication ports).

Process Model: (0) Prior to the theatre, the design team develops the theatrical props, such as dialog boxes, cursor images, and other interface components. The design team also develops object-oriented scripts. They may, as well, write a scenario to begin the drama. (1) Guided by three process characters (the audience agent, the critic, and the spirit—see steps 2–5), the characters (portrayed by members of the design team) introduce themselves to the audience of end-users and other stakeholders. (2) The audience agent works with the audience (a communal "user") to tell the cursor character and perhaps the keyboard character how to input to the system. Following their scripts, the cursor and keyboard characters send messages to other roles. (3) The other roles respond with methods and their own messages. (4) The critic works with the audience to critique the design. (5) The spirit attempts to highlight design and work questions, and to keep track of everyone's needs. (6) The play is re-enacted, with changes, until everyone is satisfied.

Participation Model: Actors (design team) and audience (end-users and other stakeholders). Perhaps the

process characters (audience agent, critic, and spirit) should be external facilitators.

Results: Modified interface-component theatrical props. Modified object-oriented scripts. Improved understanding of developers' visions and users' visions.
Phases of the Lifecycle: Assessment.

Complementary Formal Methods: Object-oriented design, object-oriented programming.

Group Sizes: Very large.

References:

Muller, M. J., Wildman, D. M., and White, E. A. (1994, April). *Participatory design through games and other group exercises.* Tutorial at CHI '94 conference, Boston.

11.10.32 JAD (Joint Application Design, or Joint Application Development)

Abstract: Selected user representatives are involved with many other people in highly structured, disciplined sessions. A neutral, skilled leader/facilitator is important, even more than users. The goal usually is not political or philosophical, but is to speed the design of, and improve the quality of, information systems. Users at the meeting need not be representative of the user population, because they are invited for their expertise. There is no single JAD method, but a family of methods descended partly from work by Chuck Morris and Tony Crawford of IBM in 1977. Other names for members of this family of methods are Joint Application Requirements, Joint Requirements Planning Interactive JAD, Interactive Design, Group Design, Accelerated Design, Team Analysis, Facilitated Team Techniques, and Rapid Application Development (RAD).

Object Model: Flip chart paper on walls, overhead projector transparencies, magnet board, flow charts, text, and sometimes CASE tools.

Process Model: The leader/facilitator enforces a strict agenda and time limit, controls who speaks publicly and who writes on the public surfaces, and often has the sole writing privilege. Public memory is the writing on walls, or a wall projection of CASE tool displays being produced by the scribe or leader/facilitator. There are several kinds of activities, including brainstorming and issue resolution.

Participation Model: A neutral leader/facilitator who is trained specifically in JAD, and who often is considered to be the key to the entire process, even more so than the users; "users," who can be either real end-users or their managers; sometimes the executives who have the power to define the entire project and its resources; scribe; the information system project's staff: analysts, project managers, database personnel, and technical experts.

Results: An information system design.

Phases of the Lifecycle: Requirements, analysis, high-level design, perhaps detailed design.

Complementary Formal Methods: This *is* a rather formal method, because of the strict and strictly enforced agenda, and the required formal and standardized training of the leader/facilitator.

Group Sizes: Commonly 14, though many other sizes are possible.

Reference:

Carmel, E., Whitaker, R. D., and George, J. F. (1993). PD and Joint Application Design: A transatlantic comparison. *Communications of the ACM, 36*(6), 40–48.

11.10.33 KOMPASS

Abstract: Participatory method for function allocation, job design, and socio-technical design for the complementary development and evaluation of design options for highly automated work systems.

Object Model: No special materials.

Process Model: Series of six steps: (1) Define system objectives and requirements through discussions with users. (2) List the functions the work system is to perform. (3) Flag functions according to decision requirements, type of activity, process transparency, and automation potential. (4) Decide on allocation based on function characteristics: human only, machine only, or joint human and machine. (5) Develop allocation options for functions done jointly by humans and machines. (6) Evaluate options based on three different levels: human-machine system, individual work tasks, and work systems.

Participation Model: Current operators, future operators, managers, system designers.

Results: Development options for system.

Phases of the Lifecycle: Analysis, high-level design.

Complementary Formal Methods: None known.

Group Sizes: Unspecified.

References:

Grote, G. (1994). A participatory approach to the complementary design of highly automated work systems. In G. Bradley and H. W. Hendrick (Eds.), *Human factors in organizational design and management—IV*. Amsterdam: Elsevier.

11.10.34 Layout, Organization, and Specification Games

Abstract: Games in which users get the chance to see operations from other points of view, to determine desired changes in layout and organization.

Object Model: (1) Layout Game: Large sheet of paper with layout of building, wood cards with different tools and accessories. (2) Organization Game: Set of situation cards describing market opportunities, new technological possibilities, and economic or political changes. (3) Specification Game: Large piece of paper, pens.

Process Model: Series of three games: (1) **Layout Game:** Users place the cards with the tools and accessories in the rooms in the building. This allows for an understanding of the present state. The finished layout can then be used to identify problems and design new alternatives for individual workplaces and the overall business layout. (2) **Organization Game:** Users take turns choosing situation cards and reacting as managers. Users take one of three roles: the tycoon, the stingy manager, and the enlightened owner. After the game, the outcome is discussed, focusing on the relationship between quality, business ideas, and the design of technology and work. (3) **Specification Game:** Results from the Layout and Organization Games are structured and refined. Users discuss aspects of the product, technology, organization, and work that have been made in the past. Through discussion about quality, users develop their own demands in these areas.

Participation Model: Users and software professionals.

Results: (1) Layout of business with proposed changes. (2) Proposed changes in the product, technology, organization, and work, to help increase quality. (3) Increased understanding about work of users, managers, and software professionals.

Phases of the Lifecycle: Problem identification, requirements, analysis.

Complementary Formal Methods: Mock-ups, prototyping.

Group Sizes: 2–8.

References:

Ehn, P., and Sjögren, D. (1991). From system descriptions to scripts for action. In J. Greenbaum and M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 241–268). Hillsdale, NJ, USA: Erlbaum.

Klær, A., and Madsen, K. H. (1995). Participatory analysis of flexibility. *Communications of the ACM, 38*(5), 53–60.

See also the entries on *CARD*, *CUTA*, *Metaphors Game*, and *PictureCARD*.

11.10.35 Lunchbox Project

Abstract: The lunchbox project used drawings and collages to understand children's ideas about what they would like their lunchboxes to look like. Similar exercises were used, on a tutorial basis, to design bedside alarm clocks.

Object Model: (1) Drawing materials. (2) Collage images drawn from graphical images in magazines and suchlike are color-photocopied and treated with a temporary (removable) adhesive.

Process Model: (1) Draw the design that you would like. (2) Assemble and arrange images that suggest the desirable attributes of the design (i.e., attributes and connotations, rather than specific features and interface objects).

Participation Model: End-users or consumers, facilitators.

Results: Graphical drawings and images that are suggestive (rather than definitive) of what the design should be.

Phases of the Lifecycle: Requirements.

Complementary Formal Methods: None known.

Group Sizes: Perhaps as many as 20.

References:

Nutter, E. N., and Sanders, E. B. N. (1994). Participatory development of a consumer product. *PDC '94: Proceedings of the Participatory Design Conference,* 125–126.

Sanders, E. B. N. (1992). Participatory design research in the product development process. *PDC '92: Proceedings of the Participatory Design Conference,* 111–112.

Sanders, E. B.-N., and Nutter, E. H. (1994). Velcro[10]-modeling and projective expression: Participatory design methods for product development. *PDC '94: Proceedings of the Participatory Design Conference,* 143.

11.10.36 Metaphors Game

Abstract: This game helps to develop a systematic metaphorical model for a complex system domain, with the goal of providing a potential mental model to make it easier for end-users to understand the system. A team explores one or more metaphor domains and attempts to match their attributes to the current design problem, using a card game and a board game to structure their work.

Object Model:

- *Card Games:* Formatted template cards. Each card indicates the metaphoric domain (or system domain), at least one attribute of that domain, and some notes on how the attribute relates to other attributes.
- *Board Game:* The playing boards are spaces in which to organize and interrelate the cards from the preceding card games.

Process Model: (1) The team explores one or more metaphorical domains, writing down the attributes of each domain on cards, one card per attribute. (2) The team explores the system or work domain, writing down its attributes on cards, one card per attribute. (3) The team explores potential matches of metaphorical attributes with the system's attributes, aligning the card on a board.

Participation Model: End-users and one or more of the following: human factors workers, software professionals, marketers, technical writers, trainers, perhaps clients or customers of the end-users, other stakeholders in the system.

Results: Informal understanding of the extent of match of several metaphorical domains onto the system domain. Materials (cards aligned on boards) that can easily be transcribed into a table that pairs metaphorical attributes with system attributes.

Phases of the Lifecycle: Analysis. High-level design.

Complementary Formal Methods: Requirements analysis. Development of "mental models."

Group Sizes: Up to 8.

References:

Muller, M.J., Wildman, D.M., and White, E.A. (1994, April). *Participatory design through games and other group exercises.* Tutorial at CHI '94 conference, Boston.

See also the entries on *CARD*, *CUTA*, *Layout Kit,* and *PictureCARD.*

11.10.37 Mock-Ups

Abstract: Computer technology is symbolized, and perhaps simulated, using coarse-granularity mock-ups (e.g., cardboard boxes for workstations and printers) or fine-granularity mock-ups (e.g., detailed screen images on cardboard boxes, with smaller boxes simulating mice). Developers and users walk through contextualized work scenarios, referring to the mock-up technologies as appropriate, to explore the potential changes of new computer technologies.

Object Model: Plywood, paper, overhead projectors, slide projectors, boxes, pencils, and so on, to simulate tools used in work. Most importantly, the users' workplace.

Process Model: Iterative process: (1) Develop models of potential solutions, which can be used in simulations. These should start out very simple and get more realistic with successive iterations. In later iterations electronic prototypes may also be used. (2) Simulate work with the solution. Allow the user to do the work that is to be supported, step-by-step with the model. This helps illustrate what information is needed. At the same time the systems designer should point out possibilities and limitations of the proposed solution. (3) Change the model based on information gained through the simulation.

Participation Model: End-users, developers, designers, perhaps facilitators.

Results: (1) Models of possible systems to support the

[10] Velcro is a registered trademark of Velcro USA, Inc.

work. (2) Mutual learning.

Phases of the Lifecycle: Problem identification and clarification, requirements, analysis, high-level design, implementation, assessment, redesign.

Complementary Formal Methods: Prototyping.

Group Sizes: 2–40.

References:

Bjerknes, G., and Bratteteig, T. (1987) Florence in Wonderland: System development with nurses. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 279–295). Brookfield, VT, USA: Gower.

Bødker, S., Ehn, P., Kyng, M., Kammersgaard, J., and Sundblad, Y. (1987): A UTOPIAN Experience: On design of powerful computer-based tools for skilled graphic workers. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 251–278). Brookfield, VT, USA: Gower.

Ehn, P. (1988). *Work-oriented design of computer artifacts*. Falkoping, Sweden: Arbetslivcentrum/Almqvist and Wiksell International (2nd ed.: Hillsdale, NJ, USA: Erlbaum).

Ehn, P., and Kyng, M. (1991). Cardboard computers: Mocking-it-up or hands-on the future. In J. Greenbaum and M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 169–196). Hillsdale, NJ, USA: Erlbaum.

11.10.38 ORDIT (Organizational Requirements Definition for IT systems)

Abstract: Process and tools to support communication between problem owners and developers, to generate and evaluate alternative socio-technical options for the future.

Object Model: Materials for modeling; materials for prototyping.

Process Model: Support debate between interested parties during iterative process to capture emergent requirements: (1) Get input from a wide range of task and user analysis methods, to determine user and task requirements. (2) Model situation to help generate solution options. Useful techniques include scenarios, enterprise models, and requirements reference models. (3) Generate socio-technical solutions. (4) Evaluate solutions through prototyping. This generates discussion about requirements. (5) After iterations are complete, capture the requirements that have emerged through the process.

Participation Model: User organization; may also include external developers, consultants, end-users, and so on.

Results: (1) Increased understanding of organizational and technical constraints and opportunities. (2) Jointly agreed statement of user requirements, including organizational and non-functional requirements.

Phases of the Lifecycle: Problem identification, requirements.

Complementary Formal Methods: SSADM, IE, and so on.

Group Sizes: Varies by activity.

References:

Harker, S. (1993) Using case studies in the iterative development of a methodology to support user-designer collaboration. *Adjunct Proceedings of INTERCHI '93*, 57–58.

11.10.39 Organization Game

See the entry on *Layout, Organization, and Specification Games*.

11.10.40 Participatory Ergonomics

Abstract: Using conventional quality process methods, workers contribute to the solution of usability problems, usually on the shop floor.

Object Model: Actual work.

Process Model: Standard quality process methods (brainstorming, fishbone charts, etc.).

Participation Model: Users/workers by themselves, sometimes with a facilitator.

Results: Proposals to management for changes to work processes and conditions, documented in quality process formats.

Phases of the Lifecycle: Assessment, perhaps problem identification and clarification.

Complementary Formal Methods: Total quality management.

Group Sizes: Variable.

References:

Noro, K., and Imada, A. S. (Eds.). (1991) *Participatory ergonomics.* London: Taylor and Francis.

11.10.41 Participatory Heuristic Evaluation

Abstract: Inspectors use an extended set of heuristics—some product-oriented, some process-oriented—to assess potential problems of a design, prototype, or system, in terms of both usability and appropriateness to the end-users' work.

Object Model: The design, prototype, or system to be evaluated. The set of heuristics.

Process Model: (1) The inspectors are oriented to the task of heuristic evaluation, including an exploration of the 14 heuristics used in this practice. (2) Inspectors carry out a free exploration or a scenario-guided exploration of the design, prototype, or system; one person keeps a list of problems found.

Participation Model: End-users, human factors workers, development team, other stakeholders.

Results: A list of potential problems in terms of usability or work-appropriateness, from the combined perspectives of end-users, human factors workers, developers, and other stakeholders.

Phases of the Lifecycle: Assessment.

Complementary Formal Methods: Inspection methods, discount usability methods, especially heuristic evaluation.

Group Sizes: Up to 10, but better with smaller sets.

References:

Muller, M. J., McClard, A., Bell, B., Dooley, S., Meiskey, L., Meskill, J. A., Sparks, R., and Tellam, D. (1995). Validating an extension to participatory heuristic evaluation: Quality of work and quality of work life. *CHI '95 Conference Companion,* 115–116.

11.10.42 PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration)

Abstract: Using low-tech materials, participants prototype the appearance and—at a descriptive level—the dynamics of a system with a textual or graphical user interface. The technique is most useful for design, but can also be used for assessment of a simulated paper-prototyped system, or for analysis.

Object Model: Common office supplies (colored pens, scissors, Post-it notes, colored acetate for highlighting). Customized, pre-printed materials (e.g., interface components that conform to a particular style guide or development environment, icons from the work domain).

Process Model: (1) Introductions that include each participant's expertise, contribution to the shared work, organization and constituency, and what that constituency is expecting from the participant (i.e., personal and organizational stakes). (2) Mutual education through "mini-tutorials," if needed. (3) Working together to explore the task domain (in the form of analysis, design, or assessment). (4) Brief walkthrough of the group's achievements during the session, preferably recorded on videotape.

Participation Model: End-users and one or more of the following: human factors workers, software professionals, marketers, technical writers, trainers, perhaps clients or customers of the end-users, other stakeholders in the system.

Results: (1) Paper artifacts representing the appearance of the system. (2) Paper artifacts that can be used to reconstruct the group's ideas about the dynamics of the system. (3) Videotaped walkthrough showing the appearance, dynamics, and summary rationale.

Phases of the Lifecycle: Design and assessment. Analysis, to a lesser extent.

Complementary Formal Methods: Prototyping.

Group Sizes: Up to 8 people, if the furniture allows all of them to work on the same set of shared materials.

References:

Muller, M. J. (1991). PICTIVE—An exploration in participatory design. *Proceedings of CHI '91,* 225–231.

Muller, M. J. (1992). Retrospective on a year of participatory design using the PICTIVE technique. *Proceedings of CHI '92,* 455–462.

Muller, M.J., Tudor, L.G., Wildman, D.M., White, E.A., Root, R.W., Dayton, T., Carr, R., Diekmann, B., and Dykstra-Erickson, E.A. (1995). Bifocal tools for scenarios and representations in participatory activities with users. In J. Carroll (Ed.), *Scenario-based design for human-computer interaction.* New York: Wiley.

For related approaches (which did not adopt an explicitly participatory agenda), see:

Rettig, M. (1994). Practical programmer: Prototyping for tiny fingers. *Communications of the ACM, 37*(4), 21–27.

Virzi, R. (1989). What can you learn from a low-fidelity prototype? *Proceedings of the Human Factors Society 33rd Annual Meeting*, 224–228.

11.10.43 PictureCARD

Abstract: In situations in which end-users and software professionals do *not* share a common language, they communicate using highly pictorial cards to develop a representation of work.

Object Model: Cards using almost exclusively pictures (digital images reduced to line drawings) of objects and events in the users' world. Cards are grouped into six major categories: Person, Action, Season, Tool, Event, Location (PASTEL).

Process Model: Cards are arrayed in a linear sequence, beginning with the general PASTEL categories and then refining those categories into specific subclasses.

Participation Model: End-users and software professionals.

Results: Stories told by the users, initially expressed through the cards, and subsequently translated into text.

Phases of the Lifecycle: Analysis.

Complementary Formal Methods: Object-oriented analysis and design.

Group Sizes: Very small: the storyteller, the card-provider, and perhaps observers.

References:

Tschudy, M.W., Dykstra-Erickson, E.A., and Holloway, M.S. (1996). PictureCARD: A storytelling tool for task analysis. In *PDC'96 Proceedings of the Participatory Design Conference*, 183-191.

See also the entries on *CARD*, *CUTA*, *Layout Kit*, *Metaphors Game.*

11.10.44 Pluralistic Walkthrough

Abstract: End-users participate in an inspection team that evaluates a user interface or system design. The inspection sessions are designed to highlight and emphasize end-users' perspectives.

Object Model: The system or design to be inspected.

Process Model: The team inspects the system or design. End-users and their comments and perspectives are assigned the primary and most privileged position in the team's inspection agenda.

Participation Model: End-users and software professionals.

Results: Critique of the design or system.

Phases of the Lifecycle: Assessment.

Complementary Formal Methods: Software inspections, usability testing.

Group Sizes: Not specified, but manageably small for a team effort.

References:

Bias, R. (1994). Pluralistic usability walkthrough: Coordinated empathies. In J. Nielsen and R. L. Mack (Eds.), *Usability inspection methods.* New York: Wiley.

Bias, R. G. (1991). Walkthroughs: Efficient collaborative testing. *IEEE Software, 8*(5), 58–59.

11.10.45 Priority Workshop

Abstract: Users and developers collaborate on redesign of a system or systems, usually in a matrix of multiple stakeholder organizations. The practice follows a sequence of eight activities in a workshop format.

Object Model: None. The work involves a series of workshops with users and other stakeholders.

Process Model: Eight stages: (1) Introductory discussion on the aim of the workshop. (2) Users' presentations of system attributes characterized as positive, negative, and desirable. (3) Developers' presentation of plans and priorities concerning the system. (4) Exploration of alternatives through prototypes and mockups, conducted in small groups. (5) Plenary discussion of alternatives in light of users' presentations from Stage 2. (6) Summary of priorities and qualities, subjected to rank ordering through users' ratings of "+" or "-." (7) Discussions of organizational consequences (for the users) of the changes selected in Stage 7, including modes of further user participation. (8) Clos-

ing discussion and summation, including plans for further such workshops.

Participation Model: Users and developers; also, project leader and/or manager. There is a need for a moderator and for a recorder, who may need specialized skills. In any event, the moderator appears not to be a member of the stakeholder organizations.

Results: Decisions regarding features and capabilities to be included in the redesign. The decisions are informed by an understanding of the implications for the users' organizations.

Phases of the Lifecycle: Redesign, involving multiple, interrelated user organizations, and potentially multiple, interrelated software modules.

Complementary Formal Methods: Requirements, perhaps analysis.

Group Sizes: 10–20 (tentative estimate).

References:

Braa, K. (1995). Priority workshops: Springboard for user participation in redesign activities. In *Proceedings of the Conference on Organizational Computing Systems: COOCS '95*. New York: ACM.

11.10.46 PrOTA (PRocess Oriented Task Analysis)

Abstract: Takes a set of task steps arranged in a flow, and reorganizes them for sensibility of expression in a user interface. In this way, PrOTA is a bridge between high-level and detailed designs of a process, and so can be used in between (for example) the CARD and PICTIVE methods.

Object Model: Index cards and Post-it notes.

Process Model: Participants break the input task flow into (1) individual tasks, and (2) individual contexts (environments) in which those tasks are done. Each task step is written on an index card. A taxonomy of tasks is created by clustering those index cards into piles, each pile representing a common environment in which those tasks are done.

Participation Model: Users, usability engineers.

Results: Clusters of task steps, each cluster being an environment common to the task steps within it. These environments can be input to methods such as PICTIVE that design details of interfaces. If a set of physical equipment was being designed instead of a computer interface, each environment might become a room containing that equipment. For a GUI design, each environment might become a GUI window or menu.

Phases of the Lifecycle: Bridge between high-level design and detailed design.

Complementary Formal Methods: None known.

Group Sizes: 2–4.

References: Contact Susan Hornstein, Bellcore, Room PY1 1L-175, 6 Corporate Place, Piscataway, NJ 08854, USA, +1-908-214-9631, susanh@cc.bellcore.com.

11.10.47 Prototyping

Abstract: Prototyping has been used in many ways in participatory activities. This brief entry in this appendix provides a starting point for exploring the various approaches. Note, first, that low-tech prototyping is covered elsewhere (see BrainDraw, CARD, CUTA, Mock-Ups, PICTIVE). Note also the full methodological descriptions of Storyboard Prototyping and CISP (Cooperative Interactive Storyboard Prototyping). This entry on Prototyping, then, lists points of access for prototyping initiatives that are not as fully detailed as the practices described elsewhere in this appendix.

Object Model: Software system, usually programmed in a flexible environment that supports rapid changes.

Process Model: Varies from one prototyping approach to another, and from one phase of the lifecycle to another.

Participation Model: Users, developers, and perhaps other stakeholders. Sometimes a computer-literate human factors worker replaces, or works with, the developer.

Results: One or more of: (1) Improved software. (2) Improved requirements. (3) Documentation of users'needs.

Phases of the Lifecycle: Potentially all of the following: Requirements, analysis, high-level design, detailed design, implementation, assessment, customization, redesign.

Complementary Formal Methods: Software design and development. Sometimes object-oriented methods and technologies.

Group Sizes: Usually quite small.

References:

Bødker, S., and Grønbæk, K. (1991a). Cooperative prototyping: Users and designers in mutual activity. *International Journal of Man-Machine Studies, 34,* 453–478.

Bødker, S., and Grønbæk, K. (1991b). Design in action: From prototyping by demonstration to cooperative prototyping. In J. Greenbaum and M. Kyng (Eds.), *Design at work: Cooperative design of computer systems* (pp. 197–218). Hillsdale NJ, USA: Erlbaum.

Bødker, S., Grønbæk, K., and Kyng, M. (1993). Cooperative design: Techniques and experiences from the Scandinavian scene. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 157–175). Hillsdale NJ, USA: Erlbaum.

Budde, R., Kautz, K., Kuhlenkamp, K., and Züllighoven, H. (1992). *Prototyping: An approach to evolutionary system development.* Berlin: Springer Verlag.

Budde, R., Kuhlenkamp, K., Mathiassen, L., and Züllighoven, H. (Eds.). (1984). *Approaches to prototyping.* Berlin: Springer Verlag.

Floyd, C., Züllighoven, H., Budde, R., and Keil-Slawik, R. (Eds.). (1992). *Software development and reality construction.* Berlin: Springer Verlag.

11.10.48 Scenarios

Abstract: Descriptions of work move from the abstract and decontextualized toward the concrete and situated, through the usage of specific stories about specific workplace events. These stories, or scenarios, can work as triggers for other participatory activities.

Object Model: None.

Process Model: Observe, or if necessary construct, very specific, contextualized scenarios. Use these in discussions with users and others.

Participation Model: Users and software professionals; perhaps other stakeholders.

Results: Increased understanding.

Phases of the Lifecycle: Various.

Complementary Formal Methods: None known.

Group Sizes: Small.

References:

Bødker, S., Christiansen, E., and Thüring, M. (1995). A conceptual toolbox for designing CSCW applications. In *COOP '95: Atelier international sur la conception des systèms coopératifs* [International workshop on the design of cooperative systems] (pp. 266–284). Sophia Antiipolis, France: INRIA.

See also the entry on *Storytelling Workshop.* More generally, see chapter 17 of this handbook, and:

Carroll, J. (Ed.). (1995). *Scenario-based design: Envisioning work and technology in system development.* New York: Wiley.

11.10.49 Search Conference or Starting Conference

Abstract: Participants from multiple, interrelated organizations, at multiple levels of management and power, meet together to analyze current working relationships, future opportunities, and how to move from the current to the future. Participants at different levels of power are partially protected from risks due to exposing their ideas or perspectives to their own organizations.

Object Model: None.

Process Model: In general, the workshop is structured so that the high-risk discussions take place among people who are at the same organizational level. Medium-risk discussions take place among people who are at different organizational levels, but who are *not* in direct reporting relationships with one another. It is only in the low-risk discussions that participants work directly with their own direct supervisors.

Participation Model: Members of the organizations from (usually three) levels of labor and supervision. Facilitators.

Results: Improved understanding (1) among organizations, and (2) among levels in each organization. An action plan to transform the current situation into the desired future situation.

Phases of the Lifecycle: Problem identification and clarification.

Complementary Formal Methods: None known.

Group Sizes: Large—multiple organizations in conversation with one another. This is a conference-scale workshop.

References:

Palshaugen, O. (1986). *Method of designing a starting conference.* Oslo: Work Research Institute.

11.10.50 Specification Game

See the entry on *Layout, Organization, and Specification Games.*

11.10.51 SSADM (Structured Systems Analysis and Design Method)

Abstract: Method developed to help overcome problems that are encountered in the design (such as cost and schedule overruns), while maintaining quality. User involvement during analysis and design ensures the system is based on the "real world" and can meet changing requirements.

Object Model: Paper, flipcharts, pens.

Process Model: Series of six steps, with quality assurance reviews at the end of each step: (1) Analyze system operations and current problems. This is done using data flow diagrams and logical data structures. (2) Specify requirements using data flow diagrams, logical data structures, entity life histories, and logical dialog outlines. (3) Select technical options. (4) Design data. This is done using relation data analysis (third normal form) and composite logical data design. (5) Design processes using entity life histories, logical dialog outlines, and process outlines. (6) Design physical system.

Participation Model: Users and developers.

Results: (1) File/database design. (2) Program specifications. (3) Manual procedures. (4) Operating schedules. (5) System testing and implementation plans.

Phases of the Lifecycle: Analysis, design.

Complementary Formal Methods: Interviews with users, user-interface design methods.

Group Sizes: 2–8.

References:

Longworth, G. (1992). *A user's guide to SSADM* (Version 4). Oxford: N. C. C. Blackwell.

11.10.52 SSM (Soft Systems Methodology)

Abstract: A well-known general methodology for handling problem situations, where all stakeholders are included to gain multiple perspectives to derive feasible and desirable solutions.

Object Model: Paper, flipcharts, pens to create rich pictures and models

Process Model: (1) Analyze the cultural situation as a group to create rich pictures about the problem situation. This involves brainstorming about the planned intervention, as well as about the social and political systems that exist. (2) Develop conceptual models of relevant systems to make different perspectives explicit, and to show activities that are to be supported by the technical system. (3) Examine differences between the models created and the real world. (4) Identify changes that are both feasible and desirable; this may include a precise list of system objectives. (5) Take steps to improve the situation, in this case partially through a technical system.

Participation Model: All stakeholders, including users, managers, developers.

Results: (1) Models of systems relevant to the problem situation. (2) Plan of action to improve the situation.

Phases of the Lifecycle: Problem identification, requirements, analysis, evaluation.

Complementary Formal Methods: Data-flow models to represent supporting technical systems.

Group Sizes: Any size.

References:

Checkland, P. (1981a). *Systems thinking, systems practice.* New York: Wiley.

Checkland, P. (1981b). Towards a systems-based methodology for real-world problem solving. In Open Systems Group (Eds.), *Systems Behaviour* (3rd ed., pp. 288–314). London: Harper and Row.

Checkland, P., and Scholes, J. (1990). *Soft Systems Methodology in Action.* New York: Wiley.

Vidgen, R., Wood-Harper, T., and Wood, R. (1993). A soft systems approach to information systems quality. *Scandinavian Journal of Information Systems, 5,* 97–112.

11.10.53 STEPS (Software Technology for Evolutionary Participative System development)

Abstract: Framework for user-oriented cooperative development, which integrates technical and social

concerns to provide high quality products that can be adapted to changing needs.

Object Model: No special materials.

Process Model: Iterations of: (1) Establishment of project or revision. At this stage, a system concept and project strategy are developed. (2) Production of system. This stage includes cooperative system design, development of a system specification, software realization by developers, and embedment preparation by users. (3) Implementation of system version. (4) Application of system. This stage involves system use by users, and maintenance by developers.

Participation Model: Users and developers.

Results: (1) Design specification, including functional specification and changes required in the user organization. (2) System version, including hardware, software, documentation, and guidelines for work organization. (3) Mutual learning.

Phases of the Lifecycle: Lifetime of a project

Complementary Formal Methods: Other software lifecycle models.

Group Sizes: Variable by process stage.

References:

Floyd, C. (1993). STEPS—A methodical approach to PD. *Communications of the ACM, 36*(6), 83.

Floyd, C., Reisin, F.-M., and Schmidt, G. (1989). STEPS to software development with users. In C. Ghezzi and J. A. McDermid (Eds.), *ESEC '89: Lecture notes in computer science Nr. 387.* Berlin: Springer Verlag.

11.10.54 Storyboard Prototyping

Abstract: Users and others evaluate and use a prototype that exists only as a storyboard—a series of still images. This type of prototype is often faster and cheaper to create than prototypes created with traditional programming languages, so iterations of design and evaluation are faster. Some versions of the method, such as CISP (Cooperative Interactive Storyboard Prototyping), involve users in codeveloping the prototype instead of just evaluating a version so other people can go off on their own to create the next version.

Object Model: Drawings made by hand or with software, presented on paper or computer screen. The lowest-tech variety of the method uses hand drawings on sheets of paper, each drawing showing one state from a succession of states of the interface. A higher-tech variety uses software for drawing and for presenting the images in sequence. Perhaps the highest-tech variety (e.g., CISP) creates *interactive* software storyboards, in which the user's actions on the screen (e.g., using a mouse to click an on-screen button) control which image appears next.

Process Model: (1) For each scenario of use of the interface, develop a "storyboard": a series of cartoon images of the interface states as they would occur during the task's steps. These storyboards may or may not be codeveloped by the users, as in the CISP method. If users do not codevelop the initial storyboards, then a prior step is necessary: Gathering information on users' basic needs, including fundamental goals and objectives, functions to be performed, relevant data elements and relationships, and any problems to be solved by system. (2) Present the storyboards to the interested parties, including users and/or managers. Participants review the succession of images. Comments may be elicited by asking "what if" questions. In some cases, participants can control the succession of images by pointing at interface controls in the pictures, with the appropriate next image being chosen either by a human or by software. (3) Iterate between developing storyboards and presenting them, until all participants deem them satisfactory.

Participation Model: Users, usability engineers, prototypers, graphic artists, maybe developers, maybe managers and marketers.

Results: Iteratively evaluated and redesigned storyboard or prototype. Improved understanding of requirements.

Phases of the Lifecycle: High-level design, detailed design, implementation (in the sense of prototyping).

Complementary Formal Methods: Programming of fully functional software prototypes that behave as the storyboard appears.

Group Sizes: 2 to perhaps 20.

References:

Andriole, S. J. (1989). *Storyboard prototyping: A new approach to user requirements analysis.* Wellesley, MA, USA: QED.

Andriole, S. J. (Ed.). (1992). *Rapid application prototyping: The storyboard approach to user requirements analysis* (2nd ed.). Boston: QED.

See also the entry on *CISP*.

11.10.55 Storytelling Workshop

Abstract: Participants bring to a workshop two short oral stories about computer usage. The invitation to participate includes the request that one story be positive, and one story be negative, with respect to usage and outcome. Participants share their stories.

Object Model: None.

Process Model: Participants tell their stories, comment on one another's stories, and comment on commonalties and contrasts.

Participation Model: End-users, facilitator(s).

Results: (1) Increased cohesion among the end-users. (2) Recognition among the end-users that the difficulties that each of them has faced as an individual are not in fact unique.

Phases of the Lifecycle: Problem identification and clarification.

Complementary Formal Methods: None known.

Group Sizes: Medium (perhaps up to 40).

References:

Greenbaum, J., and Madsen, K. H. (1993). Small changes: Starting a participatory design process by giving participants a voice. In D. Schuler and A. Namioka (Eds.), *Participatory design: Principles and practices* (pp. 289–298). Hillsdale, NJ, USA: Erlbaum.

See also:

Erickson, T. (1995). Notes on design practice: Stories and prototypes as catalysts for communication. In J. Carroll (Ed.), *Scenario-based design: Envisioning work and technology in system development* (pp. 37–58). New York: Wiley.

Orr, J. E., and Crowfoot, N. C. (1992). Design by anecdote—The use of ethnography to guide the application of technology in practice. *PDC '92: Proceedings of the Participatory Design Conference,* 31–37.

11.10.56 TOD (Task Object Design)

Abstract: Participants design a complete set of units of information that are needed and desirable for a worker to do a task that has already been documented in a flow chart. Each object is represented by an index card. These task information objects serve as a stepping stone from the task flow to an object-oriented user interface design.

Object Model: Index cards, Post-it notes, felt-tipped pens

Process Model: Initially the task objects are just extracted from the previously documented task flow by writing each task step's noun on an index card, and each step's verbs on a Post-it note attached to that card. But participants then start designing the details of the objects, by listing (on more Post-it notes) each object's attributes and hierarchical relations to other objects. Participants also usability test the set of objects for its ease of use in doing the task flow. This is an iterative process in which objects are discarded or drastically redesigned, new objects are designed, and the task flow itself changed. All activities are done by all the participants, who are seated around a small round table, with the materials on the table.

Participation Model: Users, usability engineer, system engineer, developer, subject matter experts in the relevant business processes and information systems, developers of documentation and training, managers of all these people.

Results: A complete set of abstract information objects for doing the task flow. These "task objects" are to be used in other methods, for bridging that task flow to the designing of any object-oriented user interface for doing that task flow. For example, the task objects might be translated into GUI objects such as windows.

Phases of the Lifecycle: Analysis, high-level design, assessment.

Complementary Formal Methods: Formal requirements-document writing to put the objects with their attributes and relationships into a format more suitable than index cards, for system engineers, developers, testers, and project managers.

Group Sizes: 2 to 6.

References:

Dayton, T., Kramer, J., McFarland, A., and Heidelberg, M. (1996). Participatory GUI design from task models. *CHI '96 Conference Companion.*

McFarland, A., and Dayton, T. (1995). A participatory methodology for driving object-oriented GUI design from user needs. *Proceedings of OZCHI '95.*

See the entry on *Workshop for O-O GUI Designing*

from User Needs for a context in which TOD is used.

11.10.57 Translators

Abstract: End-users and developers find common ground through a mediator (the translator) who understands both the users' and the developers' domains.

Object Model: None.

Process Model: The translator translates between the users' way of doing work and thinking about work, and the developers' way of doing work and thinking about work.

Participation Model: End-users, developers, one translator.

Results: Increased mutual understanding. Development of improved translation techniques.

Phases of the Lifecycle: Problem identification, requirements, analysis, assessment.

Complementary Formal Methods: None known.

Group Sizes: 6–8.

References:

Williams, M. G. (1994). Enabling schoolteachers to participate in the design of educational software. *PDC '94: Proceedings of the Participatory Design Conference*, 153–158.

Williams, M. G., and Begg, V. (1993). Translation between software designers and users. *Communications of the ACM, 36*(6), 102–103.

11.10.58 UTOPIA Project—Training, Technology, and Products From the Quality of Work Perspective

Abstract: This project concentrated on the development of computer-based tools for skilled workers. The tools were designed to be skill-enhancing tools which would lead to high-quality products.

Object Model: Plywood, paper, and such, to build Mock-Ups of different tools.

Process Model: A focus on work processes rather than data flow analysis, through a series of steps: (1) Learn about the work process of the user. (2) Visit other work places doing similar work as a group. This helps to gather information about technology and work practices. (3) Develop plywood and paper mock-ups to simulate different tools. This enables developers learn about existing technology and to develop use models, while users learn about the technical possibilities. (4) Use games (see Layout, Organization, and Specification Games) to learn about the work organization. (5) Develop requirement specification. At the same time, build alternative models for work organization. (6) Organize training for users. (7) After development is complete, pilot the system at one location to see how it works.

Participation Model: Users, trade unions and developers.

Results: (1) Specification for system to be implemented. (2) Plans for new work organization. (3) Design methods appropriate for designing systems with users. (4) Mutual learning about each other's domain.

Phases of the Lifecycle: Problem identification, requirements, high-level design, evaluation.

Complementary Formal Methods: None known.

Group Sizes: Variable.

References:

Bødker, S., Ehn, P., Kyng, M., Kammersgaard, J. and Sundblad, Y. (1987). A UTOPIAN experience: On design of powerful computer-based tools for skilled graphic workers. In G. Bjerknes, P. Ehn, and M. Kyng (Eds.), *Computers and democracy: A Scandinavian challenge* (pp. 251–278). Brookfield, VT, USA: Gower.

See also the entry on *Mock-Ups*, and on *Layout, Organization, and Specification Games.*

11.10.59 Video Prototyping

Abstract: Develop a representation of the dynamics of the user interface using paper-and-pencil materials and stop-action (cartoon) animation techniques

Object Model: Video recorder; paper-and-pencil materials to sketch interface components.

Process Model: The group's ideas about how the interface should behave (the dynamics of the interface) are recorded by working through the interface events using the paper-and-pencil materials to show each component and each event involving that component. The video camera is run while the participants are moving the interface objects. When the paper computer "screen" changes (for example, when a menu is pulled down from the menu bar), the camera is stopped, the pull-down menu is placed on the paper "screen," and the camera is restarted. The cursor is

often represented as a hand-drawn arrow, which is moved across the paper "screen" on a sheet of transparent plastic or acetate. Sound effects, such as clicks and beeps, may be included. The resulting video record is an animated version of what the design would look like if it were programmed. Voice-over narration may be included.

Participation Model: An expert animator and a design team.

Results: Videotape of interface dynamics, suitable for showing others how the design is supposed to behave. Because the "interface components" in the videotape are made of paper, the videotape cannot be mistaken for a real, computerized artifact; it is a representation of the intended artifact.

Phases of the Lifecycle: Detailed design.

Complementary Formal Methods: GUI description languages.

Group Sizes: 2–8.

References:

Young, E. (1992, May). *Participatory video prototyping*. Poster at CHI '92 conference, Monterey, CA, USA.

11.10.60 Work Mapping

Abstract: Method for information analysis and modeling that can help to understand work practices important to the redesign of business processes and computer systems.

Object Model: Paper, pens, mock-ups of work environment, business systems, and computer systems.

Process Model: (1) Develop models of current work processes. (2) Enrich and test models by acting out with mock-ups. (3) Identify problems and areas for efficiency improvement in work by analyzing work models. Things that may considered at this phase include bottle necks, decisions points, computer support, information sources, business objectives, and feedback mechanisms. (4) Examine impact of changes through simulations with mock-up. (5) Update model and develop action plan.

Participation Model: Facilitators from development team; staff and management to represent different functions in work area.

Results: (1) Systematic models of work, documenting existing work processes. (2) Action plan to improve work organization and supporting technology. (3) Commitment to implement changes to work organization.

Phases of the Lifecycle: Analysis, evaluation.

Complementary Formal Methods: Prototyping; The HURON Way

Group Sizes: Up to 40

References:

URCOT. (1994). *Work mapping: Possible application in the Australia Taxation Office* (Working Paper No. 4). Melbourne, Australia: Union Research Centre on Organisation and Technology Ltd. (URCOT).

11.10.61 Workshop for O-O GUI Designing from User Needs

Abstract: Participants use index cards to analyze and design task flows, and paper prototyping to design a graphical user interface (GUI) that is good for users to do those task flows. The bridge between those two steps is the index-card based TOD (Task Object Design) method.

Object Model: Flip chart paper, index cards, Post-it notes, felt-tipped pens, Post-it transparent adhesive tape, scissors.

Process Model: There are three major steps, with iteration within and among steps. For more details of the CARD, TOD, and PICTIVE methods that are components of the Workshop, see their descriptions in this chapter. Usability testing is done frequently within and after each step. All activities are done by all the participants, who are seated around a small round table, with the materials on the table.

The three major steps are (1) A tailored version of CARD is used for producing documented user requirements. The output is a flow chart of a desirable but feasible, detailed but somewhat abstract, set of steps that a user will take when using the new user interface. (2) TOD is then used to design a set of task objects that are well usable for a worker executing the task flow from the first step. (3) The fundamental GUI design is then produced by translating the task objects into GUI objects, via a succession of small steps, a multi-platform GUI design guide, and a tailored version of PICTIVE.

Participation Model: Users, usability engineer, system engineer, developer, subject matter experts in the

relevant business processes and information systems, developers of documentation and training, managers of all these people.

Results: Paper prototype of an object-oriented, style-guide compliant, usability tested, graphical user interface, and documentation of the task flow in which the user will employ that interface.

Phases of the Lifecycle: Problem identification, analysis, requirements, high-level design, assessment.

Complementary Formal Methods: Formal requirements-document writing to put the objects with their attributes and relationships into a format more suitable than index cards and paper prototypes, for system engineers, developers, testers, and project managers.

Group Sizes: 2–6.

References:

Dayton, T., Muller, M. J., McFarland, A., Wildman, D. M., and White, E. A. (in press). Participatory analysis, design, and assessment. In J. Nielsen (Ed.), *Handbook of user interface design*. New York: Wiley.

McFarland, A., and Dayton, T. (1995). A participatory methodology for driving object-oriented GUI design from user needs. *Proceedings of OZCHI'95*.