3

Contextual Task Analysis

CHAPTER CONTENTS

Purpose 67

Description 69

Scheduling and Integration with Other Tasks 73

Roles and Resources 74

Sample Technique—A Step-by-Step Procedure 74

Level of Effort . . 100

Alternative Techniques—A Review 101

Shortcuts 103

Web Notes 104

Sample Work Products and Templates 105

PURPOSE

As in the rest of the Usability Engineering Lifecycle tasks described in this book, the focus of Contextual Task Analysis is on projects in which a specific product has already been identified, defined, and scoped. Other task analysis techniques focus on major business reengineering efforts that include identifying opportunities for new products or on identifying basic features that should be included in new products. See, for example, the excellent book by Beyer and Holtzblatt (1998) and the valuable reference book edited by Wixon and Ramey (1996). If you do have the opportunity or need to do major business reengineering before the development of specific automated products, and you have used Beyer and Holtzblatt's Contextual Design technique to do this, I might point out that their technique leaves off just where actual user interface design begins in the Usability Engineering Lifecycle. So, you could pick up where they left off by using the tasks and techniques described in Chapters 8–17 of this book.

The focus of Contextual Task Analysis here is on projects in which a specific product has already been identified, defined, and scoped

Contextual Task Analysis as it is described here is most appropriate when you already have a set of functions and features identified and scoped out for automation; you primarily need to understand the identified current work in order to optimally support it with a new product such as a software application. On such a project, major business reengineering efforts such as those described by Beyer and Holtzblatt would simply be out of scope.

Here, the purpose of this task is to obtain a user-centered model of work as it is currently performed. That is, you want to understand how

users currently think about, talk about, and do their work in their actual work environment. Ultimately, the reason for this is that when you design the new product and its user interface, you want to find an optimal compromise or trade-off between three goals:

- Realizing the power and efficiency that automation makes possible
- Reengineering work processes to more effectively support identified business goals
- Minimizing retraining by having the new product tap as much as possible into users' existing task knowledge, and maximizing efficiency and effectiveness by accommodating human cognitive constraints and capabilities within the context of their actual tasks

Traditional software development methods focus mostly on the first goal. As a result, some of the second goal and most of the third are lost, because designers and developers never really understand the users' work and their current work models. A great deal of work "reengineering" occurs in application design, and only some of it serves the first two goals above. Much of it is unnecessary and results in training and usage burdens on the user. You cannot factor in the third goal—minimizing the training overhead and maximizing efficiency and effectiveness—unless you first have a clear picture of users' current work models: how they do their work in the realities of their everyday work environment, and how they think and talk about it.

To obtain a current user work model, you must perform a Contextual Task Analysis, which differs from a traditional systems analysis as described in Table 3.1.

The purpose of this task is to define usability requirements and to point towards ways of meeting those requirements

Traditional systems analysis usually results in the inclusion of all required data and low-level functions, and structures them in a robust implementation architecture. Without a truly user-centered approach, however, it often fails to organize and present that data and functionality in a manner that supports and optimizes the work performance of real users in their real work environment. This missing piece is the whole point of a Contextual Task Analysis. Thus, the purpose of this task is to supplement more traditional types of systems analyses in order to define usability requirements and point towards ways of meeting those requirements. Then, in later tasks, these requirements can be applied directly to making user interface design decisions.

Table 3.1 Traditional systems analysis vs. Contextual Task Analysis

	Traditional Systems Analysis	Contextual Task Analysis
Goal	Input to the design of: software processes and data structures	Input to the design of: the user interface
Output	Function models and data models	Work Environment Analysis, Task Analysis, Task Scenarios, and Current User Task Organization Model
Impacts	Implementation architecture	Reengineered task organization and task sequence models, Conceptual Model Design, Screen Design Standards, and Detailed User Interface Design
Focus	Technical information processing limitations, data characteristics, and implementation architecture considerations	Human information processing limitations, current work, and current user work models
Objects of Analysis	Data and functions	Users, users' work environment, and users' work goals

Contextual Task Analysis techniques tie in rather neatly with Object-Oriented Systems Engineering (OOSE—see Jacobson et al. 1992). Chapter 1 provided a brief introduction to OOSE concepts and terms that are relevant to this task. In this chapter, these concepts and terms are referred to, and the tie-ins are noted.

DESCRIPTION

In Contextual Task Analysis you perform the following basic steps:

- Gather background information about the work being automated.
- Collect and analyze data from observations of and interviews with users as they do real work in their actual work environment.
- Construct and validate a model of the users' current task organization.

A central step in Contextual Task Analysis is the second step above, referred to in this book as contextual observations/interviews. The idea is that you must observe and interview users in their real-life work context

Observe and interview users doing their actual work to understand and discover their work models. Only then can you present a user interface that supports their work models and tasks

to understand their work and discover their work models. Only then can you structure and present functionality in an application user interface in a way that taps into users' current work models and optimally supports their tasks.

An abstract modeling of users' tasks, which is the focus of more traditional types of systems analysis, does not typically take into consideration key aspects of actual work flow, the users, and their work environment. Another way of putting this is that traditional systems analysis models work in the abstract. They don't consider the basic capabilities and constraints of human information processing, the particular characteristics of the intended user population, the unique characteristics of the work environment, and how users themselves model, carry out, and talk about their tasks.

Actors and Use Cases in OOSE are abstract concepts capturing generalities across individual users doing similar tasks. They do not necessarily capture things like common user errors, breakdowns in the task process, bottlenecks, work-arounds devised by users, ways the task overtaxes human capabilities, and so on. In a Contextual Task Analysis, you observe instances of Actors (real people) and instances of Use Cases (real work, called Task Scenarios). From these direct observations, you will gain insight into both the optimal structuring of the work from the point of view of automation and the key aspects of the user's work environment and models that will have direct implications for designing a usable user interface.

Based on an analysis of your direct observations, construct a model that represents the users' point of view: how they think about, talk about, and do their work

Based on an analysis of your direct observations, you then construct a model that represents the work from the users' point of view: how they currently think about, talk about, and do their work. This model is not directly designed into the application or its interface. It feeds into only one of the goals referred to earlier—that of tapping into existing user knowledge, habits, and capabilities—and it is juggled with the other two goals of supporting general business goals and exploiting the power of automation. This juggling happens in a later task, called Work Reengineering, described in Chapter 7.

A detailed technique for carrying out a Contextual Task Analysis is given in the Sample Technique—A Step-by-Step Procedure section later in the chapter. This technique is adapted from the field of ethnography, a science used by social and cultural anthropologists to investigate, study, analyze, interpret, and describe unfamiliar cultures (see, e.g., Button and Dourish 1996). Several key principles come from this field and can be

applied directly to the problem of understanding the "culture" of a user group, their work environment, and work for the purpose of designing a new interactive product (see Ford and Wood 1996):

- Much behavior in the target culture is tacit and cannot be extracted by direct questioning; therefore, extended direct observation of the target culture is necessary.
- Investigators cannot have a highly structured approach to their study because they have no a priori knowledge or understanding of what is important in the target culture. Only a semistructured approach is possible because unlike other types of research, the investigation does not start with any hypotheses.
- It is important to observe how artifacts in the culture are used to accomplish real goals, rather than to impose some classification scheme on them based on the investigator's use of the same artifacts.
- Similarly, it is important not to distort the meaning of the target culture's language through a mechanical translation into the investigator's language, but rather to observe and understand how terms are actually used.
- In order to study the target culture, the investigator must learn to speak the language of that culture, rather than expect the culture to learn his or her language.
- It is important to have no implicit assumptions that might bias interpretation and to test explicit assumptions in a rigorous way.

Note that these principles fly directly in the face of most traditional approaches to systems analysis.

In ethnographic terms, if you wish to truly understand the culture of Borneo so that you can improve the lives of the natives in some way, you must go to Borneo, learn the language spoken there, and live in several villages, trying to partake in their everyday life and adopt their perspectives and philosophies. You cannot assume that natives of Borneo think and work like you do. And you cannot simply find a second-generation immigrant from Borneo, sit him down, and ask him a lot of questions about life in Borneo.

To truly understand the work of a set of users, go to their workplace, learn the user jargon, and observe and talk with a representative set of users of all key types

Similarly, if you wish to truly understand the work of a set of users so you can provide them with a useful tool (such as a software application), you must go to their workplace, learn the user jargon, and observe and talk with a representative set of users of all key types. You cannot

You cannot assume that users will think or work like you, and you cannot simply assimilate a former user into your project team

assume that users will think like you or work like you. And you cannot simply assimilate a former user into your project team (and into the development culture) and use him or her as a source of information about actual users. Work modeling techniques like "Use Cases" (see Jacobson et al. 1992) and prototypes (which users can readily understand) help communicate and validate your understanding of users' work. You need to avoid constructs like those used in data modeling, which force the users to learn a whole new language to communicate about their work with designers.

We can borrow and adapt methods from ethnography, but ultimately we will be altering the behavior we study in a Contextual Task Analysis

Note that the field of ethnography seeks only to define and describe actual behavior. We can borrow and adapt methods from the field of ethnography, but bear in mind that ultimately, through technology, we will be altering the very behavior we study in a Contextual Task Analysis (for more discussion on this point, see Button and Dourish 1996). How you get from an understanding of current work practice to the design of future work practice is addressed in Chapter 7 and the design chapters, Chapters 8, 11, and 15 (see also Chin, Rosson, and Carroll 1997).

Contextual Task Analysis techniques are relatively new and therefore not yet commonplace in product development organizations even when usability experts are present. However, variations of the technique described in the section Sample Technique—A Step-by-Step Procedure have been successfully used in a number of organizations, including Microsoft, U.S. West, Digital Equipment, IBM, Varian, Hewlett-Packard, ATL (a medical equipment vendor), Claris, WordPerfect, Lotus, BJC Health System (a health-care organization made up of fifteen hospitals), the University of Washington Medical Center (see Wixon and Ramey 1996, for many case studies), The National Science Foundation (Chin, Rosson, and Carroll 1997), the International Monetary Fund (Harper and Sellen 1995), and Tektronix (Lewis et al. 1996).

Several authors have reported that information gathered through Contextual Task Analysis techniques sometimes radically altered the direction the organization had been taking on a project (see Bauersfield and Halgren 1996, 192; Wixon et al. 1996, 71–85; Ramey et al. 1996, 8–13; Muller and Carr 1996, 29; Lewis et al. 1996, 67–8; and Page 1996, 205–208). Others (see Wixon and Ramey 1996) have reported successfully using Contextual Task Analysis techniques to

- Generate ideas for new products
- Identify key features to include in products

- Design the user interface for products that have already been identified and scoped
- Improve the usability of products already in production

In this chapter I focus on using these techniques to design the user interface for products that have already been identified and scoped.

SCHEDULING AND INTEGRATION WITH OTHER TASKS

The Contextual Task Analysis task fits into the overall Usability Engineering Lifecycle in the following ways:

- The User Profile task feeds directly into Contextual Task Analysis by identifying categories of users (Actors) whose tasks must be studied.
- Contextual Task Analysis feeds directly into the Usability Goal Setting task by helping to identify different primary goals for different task types (Use Cases) and by identifying bottlenecks and weaknesses in current work processes that can be improved through good user interface design.
- Contextual Task Analysis feeds directly into the Work Reengineering task. Current user work models are "reengineered" only as much as necessary to exploit the power of automation and contribute to explicit business goals. Current user knowledge and experience are exploited as much as possible to facilitate usability.
- Contextual Task Analysis will be documented in the product Style Guide.
- Ultimately, this task will have a direct impact on all design tasks, the selection of usability testing and evaluation issues, and the design of usability testing materials.

The Contextual Task Analysis task fits into the underlying software development methodology in the following ways:

- This task can *parallel, overlap,* or *follow* development of the Requirements Model in the Analysis phase of OOSE (or function and data modeling in the requirements phase of a traditional rapid prototyping methodology). It could either identify Use Cases for

the Requirements Model or take the definition of Use Cases from the Requirements Model as its starting point for constructing Task Scenarios.

- This task (along with all other Usability Engineering Lifecycle Requirements Analysis tasks) should precede development of the Analysis Model in the Analysis phase of OOSE (or the application architecture design in a traditional rapid prototyping methodology).

ROLES AND RESOURCES

The usability roles might participate in the Contextual Task Analysis task as follows:

Task leader: A Usability Engineer should direct this task, bringing to bear specialized skills in Contextual Task Analysis.

Other resources: The User Interface Designer and all project team members should participate in all aspects of this task. The work of contextual observations/interviews can be split among project team members. All other work, as described in the next section, can be delegated to team members under the direction of the Usability Engineer. Users participate by being the subjects of contextual observations/interviews and the Task Sorting exercise. They can also be active participants in the analysis process (see, e.g. Chin, Rosson, and Carroll 1997).

SAMPLE TECHNIQUE— A STEP-BY-STEP PROCEDURE

Currently in the field of Usability Engineering, there is no well-established, universally applied, practical, and highly structured technique for performing a Contextual Task Analysis that will drive the user interface design for a product that has already been identified and scoped. It is more art than science at this point, with each usability practitioner using his or her own informal approach. However, some structured techniques are beginning to emerge, and several authors are reporting their experience with them (see, e.g., Beyer and Holtzblatt 1998; Hackos and Redish 1998). Some techniques that have worked well in my extensive experience are structured and documented in the eleven steps described here. They can be roughly grouped and summarized as follows:

- Gather background information about the work being automated (steps 1–4).
- Collect and analyze data from contextual observations/interviews of real users doing real work in their actual work environment (steps 5–8).
- Construct a users' task organization model of the current work (steps 9–11).

The steps are depicted in the process model in Figure 3.1. In this model, steps are enclosed in boxes, and the output of steps is indicated in free text around the arrow leaving a step. Although steps 5–8 are described separately and linearly, in fact, I usually conduct them in an iterative fashion. The discussion in step 5 explains how all of these steps are interrelated.

As indicated in the process model, in some cases, the output of a step is a specific type of document. Examples of these documents are offered in the Sample Work Products and Templates section later in the chapter.

These steps are written from the point of view of a Usability Engineer who is introducing Usability Engineering techniques into a development process and culture that has not previously applied these techniques. The following steps outline how the Usability Engineer in this situation can best introduce the techniques and integrate them with other development activities that are standard practice.

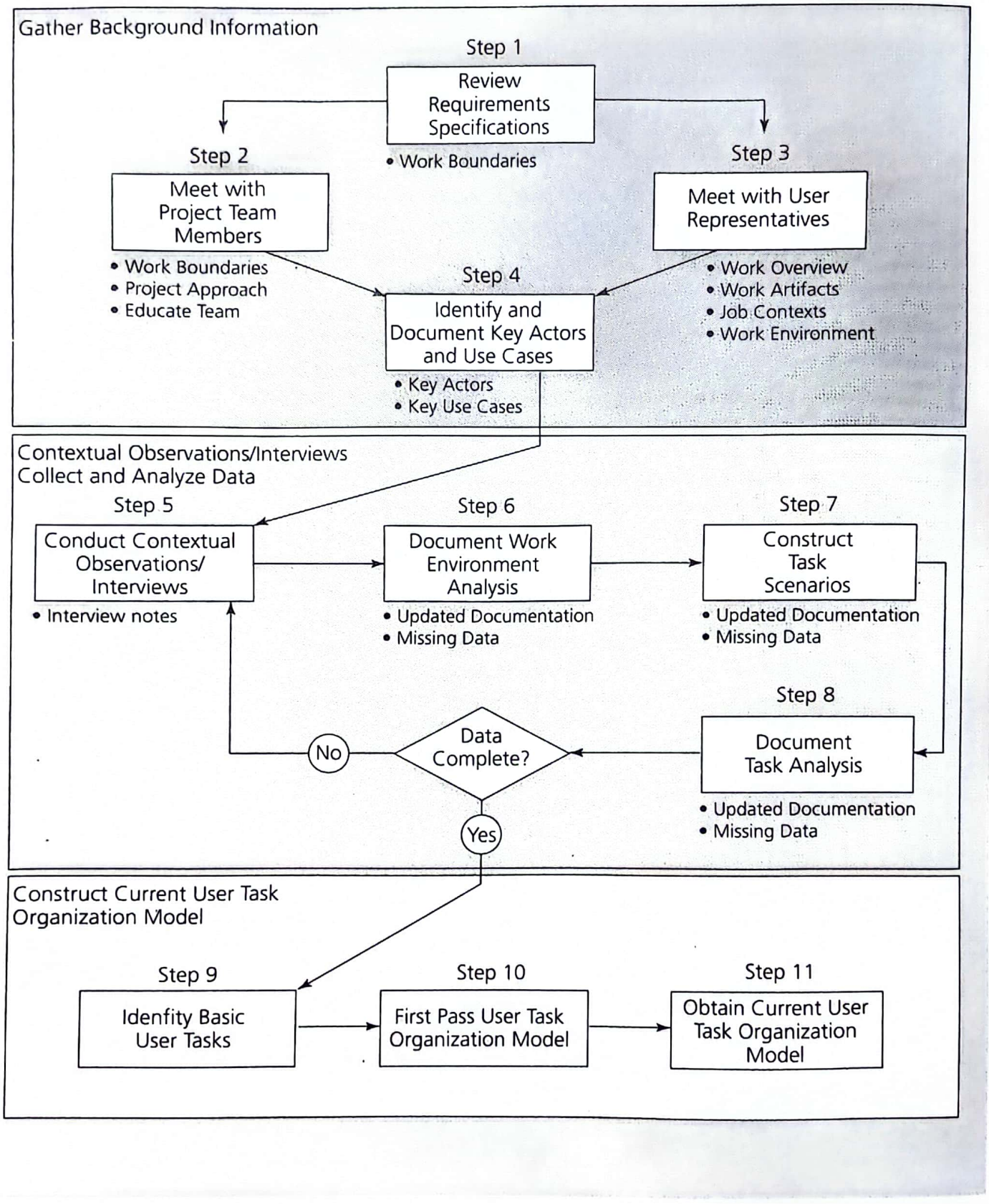
Gather Background Information about the Work Being Automated

❶ **Review requirements specifications (if available).** Sometimes usability experts are not brought in until a traditional systems analysis (or an object-oriented analysis) has been carried out and a requirements specification for the intended system has been completed (i.e., a traditional functional spec or, in OOSE, a Requirements Model). If this is the case, reading the requirements specification is a good starting point to get an overview of the user tasks that will be automated. If it is too early in the project and a requirements specification has not been completed, simply proceed to step 2 below.

Your goal here is to understand the boundaries around the work to be automated and to get a high-level picture of how the project team is planning to approach the automation of that work. This will provide context for step 5.

❷ **Meet with project team members.** Conduct interviews with project team members (e.g., managers, analysts, architects, developers) to get a high-level picture of the

Figure 3.1 Contextual Task Analysis—A process model



jobs/tasks being automated. Their perspective (and the requirements specification reviewed in step 1) will have a technical slant to it, and may reflect a systems-centered approach, but nevertheless is a good starting point to understand what the application needs to offer. At the same time, describe the approach laid out in the steps below to educate the team on the new techniques, win their confidence, and assure their cooperation.

Your goal here again is to understand the boundaries around the work to be automated and to get a high-level picture of how the project team is planning to approach the automation of that work. This will provide context for step 5. Another important goal here is to educate and get buy-in from project team members to assure their participation in all steps of the Contextual Task Analysis technique outlined below.

❸ **Meet with user representatives.** Conduct interviews with user representatives (e.g., user managers, user group leaders, subject matter experts or "SMEs")—called Indirect Users in OOSE parlance—to get a high-level picture of the jobs/tasks being automated. In some cases Indirect Users might even be outside the user organization. For example, actual customers would be considered Indirect Users of a customer support system used by customer support representatives. Similarly, travel agency customers would be Indirect Users of a travel agent's on-line reservation system. Even though they are not Direct Users or end users, Indirect Users' perspectives can be key in addressing the business goals of an on-line customer support or travel agent system.

By combining the big picture from a more user-oriented perspective with information gained from the User Profiles, you will sharpen your focus for your contextual observations/interviews (see step 5) with end users (Direct Users).

Spend some time with user representatives getting a list of (and copies of, if possible) the main *artifacts,* or *objects,* involved in the tasks being studied (see Wood 1996). Artifacts or objects might include paper forms, documents, and other types of work products, and tools such as software, special types of containers, and so on. Then look at how these artifacts or objects are used in the work tasks. Together, these objects give you a high-level overview of the "what" and "how" of the work tasks. This will provide important orientation and context for your contextual observations/interviews (see step 5).

Your goal here is to start to focus on *key* tasks (Use Cases), user types (Actors), and the artifacts or objects associated with them, and become alert to any special concerns and

issues that the user organization might have. Also, you can learn something about the physical and sociocultural work environment, which might not have become apparent from previous steps and which you can pursue further in your contextual observations/interviews (see step 5). In addition, you should focus on getting information about the whole job context of the users for whom the planned product will be just one tool supporting certain tasks. Without this, there is little hope of designing a tool that will fit with all the other tools, processes, and environment that represent the world in which the user lives and works (see Holtzblatt and Beyer 1996, 305; Beyer and Holtzblatt 1998). For one organization's approach to gathering this type of information, see Butler and Tahir (1996).

I want to emphasize that this step is not a substitute for contextual observations/interviews (see step 5), where you observe and interview real end users (Direct Users) in their actual work environment. The perspectives of SMEs and other regular partners from the user organization are helpful and important, but they are often very different from the perspectives of end users (see, e.g., Brown 1996, 161, for a contrast between a medical equipment vendor's regular "luminary" physician partners and actual end users of their products).

❹ **Identify and document key Actors and Use Cases.** Based on the User Profiles, the previous three steps, and on any OOSE requirements analysis that has been done, identify from the complete set of Actors (major user categories) and Use Cases (major types of user tasks) a subset of Actors and Use Cases on which to focus the Contextual Task Analysis.

A complex product may have a large number of Use Cases and Actors. It is often not practical or effective to study them all exhaustively at this point in the Usability Engineering Lifecycle; instead, select a small number of key Actors (Primary rather than Secondary Actors) and key Use Cases (e.g., those performed by Primary Actors and accounting for the majority of critical transactions) that do a good job of representing the whole scope of the product. The idea is that if you base a *user interface architecture* (i.e., reengineered work, Conceptual Model Design, and Screen Design Standards) on an analysis of and design for these key Actors and Use Cases, that architecture will probably extend well to the whole product. This is more efficient and probably more effective than an exhaustive analysis of Actors and Use Cases, and it is also consistent with the layered design philosophy that is the foundation of OOSE (see Jacobson et al. 1992, 16, 24).

Collect and Analyze Data from Contextual Observations/Interviews

❺ **Conduct contextual observations/interviews.** Because this step is so complex, it is further broken down into subsections. We begin with an overview.

Overview. Contextual observations/interviews are absolutely central to a Contextual Task Analysis. Here, you go out and observe and talk with actual intended end users (Direct Users) in their real work environment while they do real work (i.e., perform instances of Use Cases). Focus on those user tasks (Use Cases) that are relevant to the automation project, but also attend to the context of the user's overall job, which may be wider than the tasks being automated. Your main goals are to

- Identify the main work artifacts or objects.
- Collect Task Scenarios, or instances of Use Cases.
- Gain insight into the actual user and business goals underlying Use Cases.
- Gain insight into users' work models.
- Learn the users' language (their terminology and jargon).
- Gather statistics about Use Cases, for example, relative frequency, range and average time to complete, number and types of errors.
- Gain insight into problems, bottlenecks, errors, and other opportunities for improvement in the current work process (whether manual or automated).
- Gather data about the work environment.

Users' knowledge about their work is often tacit or no longer conscious (see Wood 1996, 38), so simply interviewing them outside their work context is not a reliable source of accurate information. By observing in the work context, you can see what the user actually does and then probe for details and underlying goals (see Butler and Tahir 1996, 260). Users are better able to reflect on and explain their own actions while they are performing them, as their own actions stimulate memory retrieval.

Also note that work practices documented in "policies and procedures" manuals often inaccurately reflect *actual* work practices, as users always modify these procedures in practice, finding work-arounds and adapting them to the real world (see Holtzblatt and Beyer 1996, 308). Users interviewed out of context will usually not report these work-arounds, either because they do them almost unconsciously or because they are reluctant to report how they deviate from official procedures in

their everyday work. Yet product designers often use these formal, "official" procedures as the main (or even the only) source of information about work practices.

A simple example of what can be missed by not doing contextual observations/interviews is found in Coble et al. (1996, 231). This report describes the use of contextual observations/interviews to study the functional requirements of physicians for a clinical workstation:

> Before the . . . session started, a physician explained the purpose and details of each section in his office chart . . . Later, when he was doing actual work in context, the person performing the . . . session noticed that the note he was looking at was written with red ink. She probed and the physician said it told him the previous encounter with this

WAR STORY

AN ARRESTING CONTEXTUAL TASK ANALYSIS

As an example of what can be missed by not doing a Contextual Task Analysis, I was performing such an analysis in a metropolitan police department. Police officers used a system of standardized paper forms to inventory confiscated property. I observed infrequent users of the forms struggling with them (there were many complex forms and many unwritten rules about how to fill them out), and I later observed how frequent users handled them. The frequent users tended to ignore the forms initially and take free-form notes describing the property they had to document in a format very different from that required on the forms. They then transcribed their own notes onto the forms as required. It was clear from this observation (and from follow-up interviews with the frequent users) that the forms were not designed in a way that supported the users' task, and a lot of what I learned about how frequent users worked around the forms went into designing better on-line forms. A traditional systems analysis would typically not involve studying users actually using paper forms in their actual work, but would have instead taken the paper forms themselves as a description of the work to be automated. Problems with the forms would have been missed entirely and perpetuated in the on-line version of the task.

There are always a great many things that users simply will not think to report during an off-site interview or "focus group." They only emerge during contextual observations/interviews of people doing real work in their workplace. Such things can have major implications for product user interface design.

> patient was a hospital visit. That fact told him he needed to review the hospital discharge summary and hospital laboratory results before entering the patient's exam room. *The physician was surprised that he had not mentioned that need before.* It was so ingrained in how he worked that he did not even process that highly relevant detail consciously anymore. (Italics mine)

Having identified a small subset of key Use Cases and Actors, you now observe and interview three to six users of each key user type or role, for several hours each, making sure that you observe instances of all key user tasks across all users. Different project team members can perform the contextual observations/interviews with different users and then consolidate their findings (see steps 6–10 below).

As a final note, if your product development project is intended to automate work that is not currently being done, it is still possible to use Contextual Task Analysis techniques to gather useful information that can help drive design.

In most cases, even when a particular job is not currently being performed, something highly related to that job *is* being done, and this can be the focus of a Contextual Task Analysis. If nothing at all related to the work your product is automating is currently being done, there is a good chance you are building a product that nobody needs! Good products are always driven by real user needs. Keep in mind also that once an initial release of a product is in production, you could perform another Contextual Task Analysis to discover how it is being used and where it breaks down, and use these insights to drive the design of later releases.

The next subsections of step 5 are meant to help you structure your contextual observations/interviews. Later steps tell how to analyze, document, and apply your findings. None of the steps as described here are as structured and clear-cut as I would like them to be. Contextual Task Analysis techniques are still evolving in the Usability Engineering field. Likewise, ethnographic research methods, which contextual observations/interviews are based on, are inherently only semistructured. But in my experience, even when designers and developers carry out contextual observations/interviews without a structured approach for collecting, analyzing, documenting, and applying this information, they gain amazing insights that help them design an application and user interface that best support real users doing real work in their real environment. So, after reading the rest of this step, even if you do not feel entirely clear about what information you should gather and how to go about getting it and applying it, *just do it!*

WAR STORY

DISCOVERING HOW USERS DO WORK BEFORE THEY DO IT

One of my clients was building a software application to help their users track work resources, such as supplies and equipment, not only to keep track of these resources and find them when needed, but also to justify budget plans for future years based on resource usage in the current year. As it happened, the user organization did not previously track work resources, and so a new job was being introduced at the same time as the software tool to facilitate the job. The client initially thought that since users were not currently doing the job (*tracking* the resources), there was nothing to do a Contextual Task Analysis on. However, as I explained to them, users were currently *using* the work resources; and so we could observe the way they found and used these resources, as well as how they named them, talked about them, and thought about them. These observations would undoubtedly give us important insights into how to present users with a way to record and track the resources in a software user interface.

I cannot overemphasize that there is nothing more valuable, important, and likely to yield key insights than spending some time actually living in the shoes of your users (see also Page 1996, 208). Once you start to do this, you may recognize additional structured approaches that work well in your organization.

Iterating through steps 5–8. Steps 5–8 should usually be performed in an iterative manner (see Figure 3.1). These steps are

Step 5: Conduct contextual observations/interviews

Step 6: Document the work environment analysis

Step 7: Construct Task Scenarios

Step 8: Document the task analysis

Some authors (Holtzblatt and Beyer 1996; Beyer and Holtzblatt 1998) advocate constructing detailed formal work models representing the types of information described in steps 6–8 *after each individual contextual observation/interview*, and then consolidating the individual models to capture commonalities after all observations/interviews

are complete. In the context of developing a product already identified and scoped (as opposed to a major business reengineering effort), I find stopping to analyze and formally model every contextual observation/interview individually too time consuming, unnecessarily thorough, and inefficient. I also find the formal models do not do a very good job of communicating important data. They are hard to interpret, and much of the richness of the data is lost.

On the other hand, it is a mistake to conduct too many contextual observations/interviews without stopping to digest and analyze what you have learned (see, e.g., Brown 1996, 174). You will lose the information that your notes might prompt you to remember if the experience is fresh in your mind, and you will simply find the interpretation, analysis, and documentation tasks overwhelming. Instead, I prefer to proceed through steps 5–8 by conducting observations/interviews in small batches, with interim analysis and documentation sessions between batches, and I prefer to document the data in written documents rather than as formal models. Usually, I conduct between three and six contextual observations/interviews for a given user type (Actor) over several days, then stop to analyze, organize, and document what I have learned in the format of the documents described in steps 6–8. Then I see where the holes are in my data and conduct the next batch of contextual observations/interviews, trying to focus on those holes. After each batch of contextual observations/interviews, I update my documents. In relatively simple projects with only one main type of user (Actor), sometimes only one or perhaps at most two iterations of steps 5–8 are necessary, with a total of three to six contextual observations/interviews. In more complex projects with many user types, more iterations are necessary, with a total of fifteen or more contextual observations/ interviews.

Finally, Page (1996, 211) offers some good advice regarding the whole process described in steps 1–11. He points out that you should not be a perfectionist about technique, or focus too much on the process itself. It is the results—that is, the ultimate impact on design—that are important. He also suggests that any organization trying out these techniques should schedule time periodically to reflect on the process and share their "learnings" with the rest of the organization.

Training observers/interviewers. Both developers and users tend to misunderstand the purpose and appropriate technique of contextual observations/interviews. As you want to have a cross-functional team perform these analyses (see Chapter 1), I highly recommend that all participating team members receive some training in

ethnographic and Contextual Task Analysis techniques before taking part in contextual observations/interviews. I have personally seen whole sessions derailed by developers who had not been sufficiently briefed about how to conduct contextual observations/interviews, and many authors in Wixon and Ramey (1996) report similar experiences. Everyone present, including the users, must understand the goals and techniques of the contextual observations/interviews and allow the most skillful interviewer present to direct and control the session.

Selecting users. Select between three and five users of each key user type (Actor). Initially, choose users known to represent "best practices," that is, users known to be task experts who do their tasks both accurately and efficiently. Users who have performed tasks frequently and for some time usually have worked out optimal ways of doing the tasks, have insights into the bottlenecks and weak points in the current system (whether manual or automated), and have devised effective and successful workarounds. In addition, they are better able to describe what they do in terms of underlying principles and abstractions (Wood 1996, 45). Later, it will be important to round out your contextual observations/ interviews to include less experienced, less frequent users, to see what their special problems and needs are; but it helps first to have, as context, a picture of the experts' approaches to tasks.

WAR STORY

WAITING FOR TASKS TO HAPPEN

On one project, I was doing a Contextual Task Analysis of property processing tasks in a large city police department. Most property is taken in as part of an arrest, and obviously arrests are not scheduled in advance. When an arrest with property is being processed, there is a great deal of time pressure to get things done and done correctly, and also a lot of distractions. Given this work environment, the best I could do was to hang around in police stations for whole days at a time, hoping I would get lucky and be there when certain tasks occurred. When things were slow, I would corner potential users and try spontaneously to interview them. Although this was inefficient and difficult to control, it worked, and I got the bulk of the information I needed. For another example of a challenging scheduling situation in a medical environment, see Brown (1996,166).

Scheduling contextual observations/interviews. Scheduling contextual observations/interviews is simple and straightforward in some situations and very difficult in others. When users do fairly independent tasks, and their work is scheduled or self-paced, it is relatively easy to arrange to be there during a certain type of task. In other cases, it's a real challenge to be there when tasks of interest are being done and have the chance to ask enough questions so you understand what is going on.

Starting contextual observations/interviews. Begin each contextual observation/interview by explaining why you are there and what you hope to gain. This is not only common courtesy but also can be crucial to the success of the observation/interview (see, e.g., Coble et al. 1996, 234; Brown 1996, 160). Give a brief description of

- Who you are (organizationally).
- The overall project these contextual observations/interviews will support.
- Your specific goals in this contextual observation/interview (e.g., to learn generally about the user's job; to observe specific tasks being performed; to get a general description of a certain type of task; to gain insight into current bottlenecks and problems; to learn about user jargon and mental models). These may vary across contextual observations/interviews, with early ones being very unstructured and later ones having a specific focus.
- What you expect of the user (e.g., just do your job and tell me about it; answer a set of specific questions; give me a grand tour of your work). Again, these will vary across contextual observations/interviews.
- How long the contextual observation/interview will last.
- How the user's input will help improve the products under development.

If you are conducting an international study using these techniques and you have brought an interpreter along, be sure the interpreter also has a clear understanding of your goals and plans so she or he will not "filter" out important data in translations of users' responses (see, e.g., Brown 1996, 165; Dray and Mrazek 1996, 148–149).

It's easy to be misperceived as Big Brother doing an inspection; this perception discourages users from being candid about their problems and the unofficial and informal work practices they have developed. So, it is especially important to break the ic and build rapport with users before beginning the contextual observation/interviev

One set of analysts (see Dray and Mrazek 1996, 150), who were interviewing people in their homes, found that bringing dinner and having some informal time before beginning work influenced the success of the contextual observations/interviews. Even when users understand that talking with you is part of their job, it is important to take the time to establish trust so that users are comfortable reporting accurately and candidly about their work.

Conducting observations/interviews. Probably the optimal number of interviewers per user is two. One interviewer can direct the contextual observation/interview and the other can take notes, or one can direct and take notes, and the other can operate a video camera (see the later section Data Capture Techniques). Any more than two observers/interviewers can have an inhibiting effect on users and also make it hard for one interviewer's train of thought and questioning to be coherently followed. (For other similar simple tips and rules of thumb about performing contextual observations/interviews, see Beyer and Holtzblatt 1996, 44–50.)

A good model to keep in mind, at least for the initial contextual observations/interviews, is *apprenticeship*. Remember that the user is the Master, or expert, and you are the Apprentice, or novice. As noted by Holtzblatt and Beyer (1996, 308–309), this model has the following implications:

- The contextual observation/interview cannot be highly structured beforehand because the Apprentice does not yet know enough about the job to know what is important to look for and ask about.
- A one-on-one contextual observation/interview is best (although observers and note takers can also be present) because each Apprentice needs to be able to learn in his or her own way and pace (see also Coble et al. 1996, 235).
- The Master is in control of the observation/interview but allows the Apprentice to probe to clarify his or her understanding.

In later contextual observations/interviews, you (the Apprentice) can introduce more structure and take more control, because you now have some basic understanding and are just trying to fill in gaps in your knowledge base.

As you conduct contextual observations/interviews, while bearing in mind the apprenticeship model suggested above, keep the contextual observation/interview focused so as not to waste either your or the user's time. Users have no idea what — of all the information they could volunteer—is and is not important or useful to you.

You must walk the fine line between focusing on the right kind of information and not losing opportunities to learn something unexpected. Remember that task analysis is like ethnography—until you get to know the culture somewhat, you don't really know what to look for and what will be important. On the other hand, you are not conducting an academic study, and you do not have unlimited time. So you must help the user to understand what sorts of things you are after. A great deal of tact is required when interrupting a user and redirecting the discussion. Don't be afraid to do so, as usually users will welcome the direction, but do so without being judgmental or critical (see Butler and Tahir 1996, 263).

When designers and developers do their first contextual observations/interviews, a common pitfall is to slip out of *information-gathering* mode and into *design* mode or user *education* mode (see Butler and Tahir 1996, 258; Coble et al. 1996, 235). Refrain from brainstorming technical solutions as the user describes his or her current process and its problems, and refrain from taking the time to educate the user in any depth about technological plans and projects (any more than a brief orientation to justify your contextual observation/interview). Remember, you are not designing yet, and you are not there to provide the user with information (see Butler and Tahir 1996, 262). You are there to gather information from the user. The user should always be doing most of the talking, taking direction from you only to change the area of focus.

Even when you are doing your best to focus on the right things, users will often derail you in similar ways. Sometimes they hope or expect that the session will provide training or solutions to their problems. Sometimes, rather than talk about their work as they do it today, they want to dictate technological solutions (see, e.g., Brown 1996, 160). Sometimes users bring along their co-workers, expecting a focus group format (see Page 1996, 200). You must work hard to keep things on track, and it is important to manage users' expectations about the purpose and format of the session both beforehand, when you schedule the interview, and at the beginning of the session. If problems still occur, gently remind the user of the purpose and necessary format of the session; and if possible, you can offer to get their personal agenda addressed at some other time in some other way (see, e.g., Butler and Tahir 1996, 258).

As you learn from your first contextual observations/interviews, work hard to adopt the users' language and jargon as quickly as possible, and use it in stating your questions (Wood 1996, 42). Do not translate users' language into technical, data modeling, or OOSE language and try to force the user to speak this language. If you do this, you will lose important information as the user tries to help you understand the

work by reformulating it instead of sharing with you what their actual work model is. For a good description of the format of a contextual observation/interview in a medical environment, see Coble et al. 1996, 233–235).

Types of data to capture. As context, be sure to first ask each user some basic questions about his or her background, such as

- How long have you worked at this location?
- How long have you worked in your current job/role?
- What relevant background/education do you have (e.g., academic background, additional course work, computer literacy, job-specific training)?

The best set of additional background questions can be identified from the User Profile, which has identified key user types (Actors). You will want to note not only what overall type each user observed/interviewed is, but also a couple of other characteristics known to vary widely for that user type. For example, some key questions to ask of physicians in the study by Coble et al. (1996, 237–238) included type of practice, subspecialty, location of practice, and typical number of patients in hospital at one time.

During the contextual observations/interviews, focus on obtaining Task Scenarios (see step 7 later in the chapter). A Task Scenario is an *instance* of a Use Case—a step-by-step description of an actual, detailed procedure that a user followed to accomplish some task. It captures the *sequence* of steps required to get a task done. For example, say the general Use Case is for a travel agent to make a set of reservations for a customer's trip. A real instance or Task Scenario might involve making airline, rental car, and hotel reservations for the Smith family (two adults, one child, and one infant) to go to Disneyland on July 23rd for a week, and trying to accommodate specific requests for such things as preferred airline and preferred hotel. The Task Scenario would identify the exact sequence of steps an agent followed to complete this specific task.

In addition to gathering Task Scenarios, ask users the following kinds of questions as you observe them at work:

- When . . . ? (to understand what triggers tasks)
- Where . . . ? (to determine the location of tasks in the workplace)
- How . . . ? (to understand steps in carrying out tasks)

- What . . . ? (to understand the objects or artifacts associated with tasks)
- Why . . . ? (to get at underlying goals)
- How often . . . ? (to identify frequent and infrequent activities)
- How long . . . ? (to get benchmark data on time to complete tasks)
- What do you call that? (to discover user jargon and terminology)
- What *errors* typically occur?
- How do you discover and correct these errors?
- What are the main *bottlenecks* in this task?
- What are the main *problems* you encounter when doing this task?
- What *work-arounds* have you found to get around the problems and bottlenecks?
- Are there any *exceptions* to normal procedures?
- What things would you most like changed?
- Do you have any specific ideas for improvement?

Data capture techniques. The main data capture technique in contextual observations/interviews is free-form note taking, but this can be supplemented by formatted data collection sheets (see this chapter's Sample Work Products and Templates section), copies of artifacts and work products, video- and audiotapes, and still photos.

Sometimes, in high-pressure, time-critical, or very difficult jobs (e.g., air traffic control, customer service, surgery, sales), it is not possible to interview users while they are doing their work. In such cases it is still very important to *observe* them in the context of their actual work, but you can also interview them outside of doing their tasks, based on notes taken during the observation. If possible, do this in their work environment, where they will be reminded of important aspects of their work and can show you work artifacts. If even this is not practical (e.g., when the task is doctors performing surgery, and the operating room is always in use), a good alternative is to videotape users as they work, and then review the videotape with them later and ask questions then (see, e.g., Ramey, Rowberg, and Robinson 1996).

Even when there are no obstacles to observing and interviewing users, you may want to use videotaping. It can be especially helpful when multiple observers/interviewers cannot get together in person to consolidate their findings. It can also be an effective way to train others to conduct contextual observations/interviews. A proficient observer/interviewer can videotape some of her or his work and then show it to others as an example of how to conduct a contextual observation/interview. Videotaping can also be particularly useful in observer/interviewers' early experience. Their

note taking may not be very effective early on, and the videotapes (or even audiotapes—see Coble et al. 1996, 234) will preserve valuable data. With more experience, observers/interviewers learn to take better notes and rely on recordings of sessions less.

I do not recommend using videotaping routinely to free yourself from taking notes during the contextual observations/interviews. Videotaping more than doubles the time to conduct an observation/interview—two to three hours to conduct it, then two to five hours to view the tape and take notes. Project teams have reported cases in which they videotaped every contextual observation/interview and then never had the time to review many of the tapes (see Rowley 1996, 141; Bauersfield and Halgren 1996, 188, 193). Unless you budget the time in advance, do not count on videotapes as your primary data collection technique. Take lots of detailed notes.

When you want to videotape, but it is unacceptable to users (sometimes it will be), audiotaping may be acceptable and will still provide a useful record from which to reconstruct data after a contextual observation/interview (see, e.g., Coble et al. 1996, 234). Also, still photos can be an effective way to capture details of the work environment. And it is often appropriate to ask users for copies of work artifacts, such as forms, documents, or database reports (see Dray and Mrazek 1996, 150–152).

When using videotape as a data collection technique, bear in mind the following guidelines from Mackay (1995) for ethical use of video. In general, consider the right to privacy of users, and keep their best interests in mind when making and using videotapes.

Before videotaping:

- Get informed consent.
- Make sure users always know when the camera is on.
- Explain the purpose of videotaping.
- Explain who will have access to the video.
- Explain possible settings for showing the video.
- Explain possible consequences of showing the video.
- Describe potential ways the video might be disguised.

After videotaping:

- Treat videotapes as confidential.
- Allow users to view videos and reconsider their permission.

- If the use of the videotape changes, seek permission again.

Editing videotapes:

- Avoid misrepresenting overall findings by biased use of video clips.
- Reveal any special effects.

Presenting videotapes:

- Protect the users' privacy.
- Do not present video clips that make users look foolish unless their identity is disguised.
- Do not rely on the power of a video to make a weak point.
- Summarize data fairly—do not misrepresent isolated incidences through video clips.

Distributing videotapes:

- Do not use videos for purposes for which they were not intended.

Document the tasks you observe in each contextual observation/interview in some common format (see the Sample Work Products and Templates section for an example). These tasks will form the foundation for construction of Task Scenarios in step 7. Also take lots of notes on whatever you observe about the physical work environment, the sociocultural environment, and the context of the task within the user's overall job (see step 6), whether tied directly to a particular task or not. These notes will feed into the development of the Work Environment Analysis document (see step 6) and the Task Analysis document (see step 8).

Generally I find that, in my earliest iterations and with very complex work situations, it is hard to know in advance the best way to capture the data, and so I just take free-form notes. In contextual observations/interviews of simpler work, and in later iterations when I have already learned something about the work and am ready to get more focused, using a tailored format something like the one offered in the Sample Work Products and Templates section is very useful.

Ending the observations/interviews. Be sure to express your appreciation for the user's time at the end of the contextual observation/interview. In some cases, users are paid for their time, and in others they are volunteers or are directed by their management to cooperate. In any case, be sure to thank them. If they are not compensated, it

can be a nice touch to send thank-you notes or give them tokens such as project mugs or T-shirts. Remember that you will probably want to do future contextual observations/interviews in the user organization, and users will talk among themselves about their experiences. Set the stage for future user receptivity to participating in Contextual Task Analyses (see, e.g., Coble et al. 1996, 234–235) by making it a positive experience and expressing your appreciation.

Greasing the wheels. To ease this whole process and help users take it in stride, it can be useful to set up a "Product Development Partnership" (Rowley 1996, 138). To do this, you recruit users on a volunteer basis before any particular project is begun and keep track of them in a database so that they are easily accessible and already understand the agenda when they are asked to participate in a contextual observation/interview. There are often organizational barriers to direct contact between developers and users, and these barriers must be broken down if true user-centered design is to be accomplished. In my experience, most users are delighted and more than willing to contribute to the design of the tools they will have to use. Sending out a general notice describing the kind of Usability Engineering work that is planned and asking for volunteers to occasionally put in time as a design partner will usually attract a large and enthusiastic response. Then, when they are contacted for scheduling contextual observations/interviews to support a particular project, users already have a general idea of what to expect, and they have already made a commitment to participate.

Wilson and colleagues (1997) point out some additional ways to facilitate user involvement in product design and development in general:

- **Motivate all stakeholders.** Users *and* user managers must be educated on and convinced of the benefit of their participation.
- **Ensure active management buy-in.** Simple agreement from user management to have users participate is not enough. Managers need to communicate to users that they can take time off from their daily work to participate in project activities when they are asked to do so.
- **Educate users about the whole design process.** Users who understand the whole user-centered design approach at a high level will be more effective participants in the process.
- **Facilitate later involvement through earlier involvement.** Users who participate early on in the project are more effective participants at later stages than

users who start later and don't understand the process or the current state of the project.

Public relations. A side effect of conducting contextual observations/interviews as part of product development is that it can be excellent PR. I have had many users express delight and appreciation for involving them in the design of their tools, and I generally find users eager to have input. Other authors report this as well (see Wixon and Ramey 1996). By and large users seem flabbergasted that anyone from their development organization is truly interested in the details of their work and is going to the trouble to learn from them. This is a sad comment on the history of relations between user and development organizations. Through Usability Engineering techniques, you have an opportunity not only to build better products, but also to build more cooperative, respectful, and productive relationships between users and developers.

It is very important, however, to manage users' expectations. Sometimes users expect to see an impact from their input immediately and directly (see, e.g., Brown 1996, 173), and they become frustrated and disillusioned when they do not. They need to understand the nature of the whole development process and how their input will feed into it.

6 **Document the work environment analysis.** This step and steps 7 and 8 are a joint effort by all observers/interviewers and other team members (including users—see Chin, Rosson, and Carroll 1997) involved in design. At this point in the overall process, it is a good idea to establish a design room (Holtzblatt and Beyer 1996, 322–323). It should be a room the team can use for the duration of the project, where team meetings can be conducted and information and design can be recorded and left in the working state.

As discussed in step 5, it is highly recommended that this step, along with steps 7 and 8, be performed iteratively. That is, an initial round of contextual observations/interviews should be performed and then immediately documented as described in steps 6, 7, and 8. Then, more contextual observations/interviews should be carried out, and after each round, the documents described in these three steps can be refined and updated. This approach avoids having to deal with an overwhelming amount of data at one time (see Rowley 1992, 135–136, 141; Coble et al. 1996, 232–233; Dray and Mrazek 1996, 152–153). It also allows observers/interviewers to digest, capture, and analyze information between rounds of contextual observations/interviews. It is easy

to get burned out by all the new information and simply be unable to process more new information effectively. Finally, with this approach, early contextual observations/interviews can be more open-ended, so as not to miss important issues, needs, and aspects of work; and later contextual observations/interviews can focus on clarifying vague data and filling in gaps that may not be apparent during one long course of observations/interviews.

In step 6, based on extracting work environment details from the most recent round of contextual observations/interviews, write (or update) a general description of the work environment, drawing specific implications for user interface design (see the Sample Work Products and Templates section for an example). Include descriptions of the following:

- The physical work environment (e.g., open/closed work areas, lighting, heat, noise level, distractions and interruptions)
- The sociocultural work environment (e.g., morale, motivation, interuser support and teamwork, past experience with and attitudes towards automation)
- The job contexts (e.g., frequency and importance of tasks within overall job, physical and sociocultural aspects of work environment unique to job)

Tools must be designed for the context in which they will be used, and this step ensures the context is fully understood. Imagine you are designing a screwdriver, and all you know is the size of the screw head. So, you design something like a traditional screwdriver, with the correct sized blade. But suppose it then turns out that the user needs to apply the screw from the inside of a narrow pipe in order to assemble some piece of equipment. Clearly, a traditional screwdriver will be useless in this context.

Similarly, suppose you know you want to design a software application for a set of users, but you have never gone to their actual work environment. So you assume a traditional officelike environment, and design assuming a traditional workstation. But suppose it turns out that in the actual work environment, users are constantly moving all around the environment to get different parts of an overall job done. Software for a traditional workstation simply will not work. You might need to design for a smaller and more portable device that can be carried around with the user, like the units carried by UPS delivery staff.

In one more example, suppose you have never visited the users' workplace, and you assume they all work in closed offices. So you design a system with voice input and output. But it turns out that users work in one big open area with desks located right

next to one another. The noise from all those people and workstations will render the system useless since most voice recognition systems simply don't work with acceptable accuracy in a noisy environment. The point is, there are many aspects of the actual work environment that will determine how well a tool will work, and so the environment itself must be studied and the tool tailored to it. That is the purpose of a work environment analysis.

One way to extract work environment data from the contextual observation/interview notes is by having the different observers/interviewers codevelop the document (as described in step 8). Another way is through a technique known as constructing an "affinity diagram" (see Holtzblatt and Beyer 1996, 318–319, and the Alternative Techniques—A Review section later in the chapter). These two techniques can also be used for developing the Task Analysis document in step 8.

7. **Construct Task Scenarios.** Based on a consolidation of all contextual observations/interviews, write (or update) two to five key, representative, high-frequency Task Scenarios, or *instances* of key Use Cases, representing real-life work tasks, for *each* major user type or Actor (see the Sample Work Products and Templates section for examples). These can be composites, constructed by merging parts of different observed scenarios in realistic ways. You need not exhaustively represent every major variation of a Use Case or cover every Use Case for each Actor; just select a small number that seem representative. The whole team of observers/interviewers—along with some key users (see Chin, Rosson, and Carroll 1997)—should develop these Task Scenarios together, from their combined experiences and notes.

   Write the Task Scenarios abstractly (but in user terms), with little or no reference to existing or future computer systems, as a list of steps. They will help capture important aspects of work flow (sequencing of steps within tasks) and will provide a representative focus and realistic context for design and testing later (see, e.g., Rosson and Carroll 1995). It can also be useful to have some videos of actual observed scenarios to take into the next step, as well as documented scenarios (see Chin, Rosson, and Carroll 1997).

8. **Document the Task Analysis.** Write (or update) a document containing Task Scenarios and other descriptions of insights gained from the contextual observations/interviews. Sections of this document might include

   - Purpose and Structure of this Document
   - Identification of Tasks and Users (this would refer to all, but also point to the

subset of key Use Cases and Actors that will be the initial focus of the design process)

- Task Scenarios (these provide realistic instances of Use Cases)
- Analysis of Current Tasks, for example:
  - Range of Task Complexity
  - Volume of Transactions across the Organization
  - Range of Frequency of Transactions by User Category
  - Level of Training
  - Error Rates
  - Level of Redundant Data Entry
  - Breakdowns in the Process
  - Bottlenecks in the Process
  - Degree of User Mobility during Tasks
  - Degree of Interrupts during Tasks

An example of content in such a document is given in the Sample Work Products and Templates section. Another good example of the type of content that might be found in this document for a medical system is given in Ramey, Rowberg, and Robinson (1996, 8–13).

When there are only a small number of observers/interviewers on a project, this document (and the one developed in step 6) can be codeveloped. One observer/interviewer goes through all of her or his notes from one batch of contextual observations/interviews and writes a first draft of the document, organized in a way that makes sense to the observer. Then other observers/interviewers, one at a time, add to that document from their batch of contextual observations/interviews, inserting their notes into the defined structure and modifying the structure if necessary. I have successfully and efficiently used this simple approach on projects with two observers/interviewers.

Construct the Current User Task Organization Model

Task Scenarios describe the sequence of steps users currently use to complete individual tasks. Across the final three steps in the sample technique, you will determine how users think of the relationship between tasks (the Current User Task Organization Model). Task Scenarios and the Current User Task Organization Model then lay the foundation for a later

task in the Usability Engineering Lifecycle: Work Reengineering. Here, you will identify all basic user tasks that will be automated in the new product and model the way users currently think about them and organize them. These are the steps:

- Identify the basic user tasks.
- Take a first pass at a Current User Task Organization Model of those tasks.
- Obtain a Current User Task Organization Model of those tasks directly from users.

Often, your model will take the form of a task hierarchy. However, a hierarchical structure is not the only way to represent a Current User Task Organization Model. It may simply not be appropriate for some types of work. But studies of expert knowledge suggest that such knowledge is generally organized hierarchically (Wood 1996, 37), and it is a structure that has proven appropriate over and over in my experience. I suggest starting with a hierarchical task structure and turning to something else only if it does not seem to capture the nature of the users' task organization models (see the section Alternative Techniques—A Review).

Bear in mind that here you are trying to determine the organization *across* tasks —the way users currently think about tasks and do them in their daily work—rather than the flow or sequence of steps *within* tasks. You have already obtained insights into within-task flow from the Task Scenarios you constructed from your contextual observations/interviews. Your Use Cases and Task Scenarios have given you a good start at identifying all basic user tasks, and you might have already gained some insights into a Current User Task Organization Model, but the steps in this section provide a specific technique for extracting a model directly from users.

9 **Identify basic user tasks.** In a meeting attended by all observers/interviewers and other project team members involved in design (see Holtzblatt and Beyer 1996, 310–311, 319–320), the group goes through all documented contextual observation/interview notes and Task Scenarios, with each team member serving as the expert on his or her own notes. Together the group constructs a single list of all the discrete work tasks that make up the users' overall jobs (see the Sample Work Products and Templates section for examples). These may correspond to Use Cases, or they may represent the building blocks of Use Cases. For example, basic user tasks in the job of tracking property in a metropolitan police department might include

- Enter incoming property from a new incident in property log.
- Document property from a new incident on proper forms.
- Get station commander's approval on property documentation.

- Turn over property to the property clerk.
- Prepare property transfer forms.
- Transfer property to a warehouse.
- Transfer property to a lab.
- Deposit cash in a bank.
- Return property to an owner.
- Update property log after property transfers.
- Destroy property.
- Auction property.
- Determine present location of property.

Similarly, basic user tasks in a customer support job at an insurance company might include

- Look up policy information.
- Change policy beneficiary.
- Change customer address.
- Take out a loan on a policy.
- Order office supplies.
- Provide customer information to sales rep.

If users do not participate directly in step 9, they should at least validate the final list of basic user tasks you generate.

10 **Take a first pass at Current User Task Organization Model.** I have found this technique to be useful for modeling users' task organizations (see Alternative Techniques—A Review for other techniques). Based on your contextual observations/interviews, organize the basic user tasks you identified in the previous step into a hierarchy reflecting a model of current user task organization, grouping things that seem to belong together both logically and in terms of work flow (see the Sample Work Products and Templates section for an example). In the next step, you will get input on this model directly from users; the purpose of this step is to set up some context in which the user input can be interpreted.

This is *not* an attempt to reengineer the work in the automated product. That comes in a later lifecycle task, Work Reengineering. It also says nothing about how functionality will be presented in the user interface to the system; you are not doing user interface design yet. Independent of any automation at all, this first pass is a reflection of your understanding of how users currently think about and do their work.

⓫ **Obtain a Current User Task Organization Model.** Now you want to empirically obtain a model that represents the way users organize the activities and artifacts of their work. One way to do this is to conduct a *task sorting exercise* with users to see how they organize low-level tasks into a hierarchy (you will most likely find important deviations from the first pass you generated in step 10).

To conduct a task sorting exercise, write each basic user task on an index card (in user terminology and using explanatory text if necessary). Recruit three to five users for each key user type (Actor) and schedule them for one to two hours each. For *each* user:

- Show them the housekeeping analogy (see the template for the task sorting exercise in the Sample Work Products and Templates section) as a way of explaining this exercise. (Read it over yourself now, to better understand the purpose of and how to conduct this exercise.)
- Ask them to group the basic user tasks in a way that makes the most sense, given how they think about and do their actual work. Encourage them to divorce themselves from artifacts of their current tools and focus on what is inherent and would be most natural, given their work.
- Ask them to take their groups and form subgroups if they haven't already done so and if this makes sense to them.
- Ask them to give labels to all groups and subgroups.
- Document the hierarchy from each user (see the Sample Work Products and Templates section for an example).

After getting input from a sample of users, compare the hierarchies and try to construct one hierarchy that captures all commonalities. Document it in the same format as the example in the Sample Work Products and Templates section. The final consolidated Current User Task Organization Model can be documented in the Task Analysis document that also contains the Task Scenarios.

Work Reengineering, a later task in the Usability Engineering Lifecycle described in Chapter 7, will be to take this Current User Task Organization Model as a base and "reengineer" it to be the task organization model upon which the application and its user interface will be based.