Gathering Customer Data

Ask a developer if he designs from customer data, and he will surely say he does. Sue went to a users' group meeting and talked to people there; Joe showed a demo at an industry show; Mary makes a point of meeting with internal customers at least once a month. These are traditional methods of maintaining customer contact. What is driving the widespread desire in the industry to go beyond these methods, to enable designers to learn more about their customers and involve customers more fully in the design?

"Design" in our sense is the intentional structuring of a system so that the parts work together coherently to support the work of people. There is plenty of formal and informal evidence that getting the design right is a major difficulty in the industry. Informally, products ship late or not at all because people cannot agree on what to build; Information Technology (IT) groups feel that the departments they serve can never make up their minds about what they want. Formally, studies show that most problems in software systems can be traced back to problems in the requirements, and the later in development a problem is caught, the more it costs to fix. Studies also show that the more customer contact a project has, the more likely it is to be successful (Keil and Carmel 1995). The literature and experience on requirements engineering demonstrate that gathering good customer data is hard. The exact combination of approaches to use on a particular project calls for careful consideration.

Getting the design right for the work is the major challenge.

---

1 This has become folk wisdom in the industry; see, for example, Daley (1977) or Boehm (1976).
and design. Simply following the organization's usual methods for gathering data will generally not produce the data a design team needs. The methods used by commercial and IT organizations are different, and we will consider them separately to show how they fall short of providing a complete view of the customer.

**MARKETING DOESN'T PROVIDE DESIGN DATA**

Developers writing commercial software usually depend on a marketing department to provide guidance on what to build. Marketing is a discipline with a long history and extensive literature—certainly longer and more extensive than software development. People have worked out effective ways of understanding a market to sell products to it. Yet, when marketing comes to a design team to tell them what to build, there's a mismatch.

"Marketing never tells us any of the things we need to know," say product designers. But the people in marketing say, "We give them all kinds of data! They just refuse to use it." In fact, understanding a market is fundamentally different from understanding what to design into a system, and the data traditionally collected for marketing has limited usefulness for product design. Marketing needs to understand what people will buy and how people make buying decisions; designers need to understand what will help people do their work better while fitting into their lives and matching their culture. There is only a limited overlap between these questions.

Marketing has developed many different techniques for finding out what people will buy. Important factors in the answer include how much money the target market has, what hardware (or mix of hardware) and other infrastructure they are committed to, what they think their big problems are, and what technology is currently "hot." This way of thinking about a market leads to asking certain questions. Given a story about how hard it is to print a label (such as the example in Chapter 1), a marketing expert might ask: Are you in a home office, small office, or large office? What kind of computer and printer do you have? Are they from the same manufacturer? What word processor do you use? Is it worth buying a more expensive one?

The designer, on the other hand, might be asking a different question: What are the different dimensions of your problem? The system in the example at this level may not be relevant; not only may the user not be able to read the report, they would also need another's starting position, into the system. The customer can order a product that meets these requirements.

Because marketing is focused on useful techniques for understanding a market by describing it, these techniques tend to be less useful to designers. A development team might ask a designer what they need: a letter, a reporting function, or a catalog description function. The answer looks obvious, but how does your key system relate to that answer? How does it fit between the components of the system?

Marketing's view of what the market is, and what technology is "hot," can be very different from what the designer needs, and these few dimensions can be important.
Marketing doesn't provide design data

Designers ask: how should it be structured?

Traditional marketing techniques can't collect design data

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the processor are you using? How often do you do this task? How much is it worth to you to have the problem fixed?

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the processor are you using? How often do you do this task? How much is it worth to you to have the problem fixed?

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the processor are you using? How often do you do this task? How much is it worth to you to have the problem fixed?

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the processor are you using? How often do you do this task? How much is it worth to you to have the problem fixed?

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the processor are you using? How often do you do this task? How much is it worth to you to have the problem fixed?

The designer's basic question is different: how can I structure a system to make people's work more efficient? This question leads to asking about the structure of the work people do: What are the parts of a letter? How is a label different from an envelope? Does anyone understand the difference between "on" and "online"? Can you reach your printer and your keyboard at the same time? A system impacts work; designing a system requires understanding work at this level. From marketing's point of view, these questions are irrelevant; none of them affect who will buy a product. Marketing wants to be able to say, "There is a market here for a product addressing these concerns. Customers in this market are companies of this kind, and they would be willing to spend this much money." That's the designer's starting point. Given that starting point, designers need to dive into the work as the people in the market perform it. They need to discover the detailed structure of existing work to see how their product can enable a new, better way of working.

Because marketing and design have different goals, techniques useful to marketing tend not to be useful to designers. Marketing techniques tend to characterize and scope the market, rather than describing the structure of its work. As a result, marketing techniques tend to be quantitative. When you want to scope a market, it may be useful to ask, "How much money do you expect to spend on equipment next year?" and average the results across all respondents. Designers must build on more qualitative data. "What are the parts of a letter, and how are they used?" The answer to this question is a description of work practice, not any sort of number. Even if a question looks like it has a numeric answer ("How far is your printer from your keyboard"), appearances are deceiving. For a designer, the true answer isn't a number, it's "Too far to keep dashing back and forth between them."

Marketing techniques generally assume you know what the questions are. When characterizing a market, this assumption may be reasonable—there are a few dimensions that matter, and they tend to repeat from problem to problem. Accordingly, marketing techniques structure the interaction and control the
resulting data. For example, surveys and structured interviews both start with a list of questions that explicitly or implicitly drive the interaction and define what is important. But as soon as design starts, no one knows what the questions are. No one knows what will turn out to be important. "Installation is the #1 problem" reports a customer satisfaction survey (a marketing technique). But what is wrong with installation (a design question)? When do installations happen, and who does them? What information is available when they do them? Which of the many alternative fixes is best?

Even the customer doesn't necessarily know what the questions are:

Users of an X-ray machine kept asking for more and more exact speed controls on their X-ray machines, trying to run the image at exactly 1/4 second per frame. It was not until someone studied the work they were doing that they realized the users just needed a timer—they were trying to run the tape at an exact speed so they could measure elapsed time. The customers requested a technical fix to the existing system, but the real issue was in the structure of the work they were doing.

This is true in general with wish lists and other customer requests; the customer will focus on a narrow fix, but understanding the context of the work that drove the request will result in more insight and better solutions. The customer acts as though the question were, "What simple tweak or addition to the system as it is will overcome the problem I'm having?" The designer wants to know, "What new concepts or features would make the system radically more appropriate to the job at hand?" Answering this question requires an open-ended technique.

None of this is to say that designers don't need to worry about what people will buy. It's only within the context of a market with needs to be met and money to spend that design makes sense. But once marketing techniques have identified a market and shown that there is money to be made there, designers must look in depth at how people in the market work to determine what

---

**Qualitative and quantitative techniques build on each other**

For simplicity's sake, we'll refer to the activities that customers are engaged in as "work" everywhere. Of course, a consumer product might support general life tasks, and a game supports play; these same techniques have been successfully applied to both environments.
The rocky partnership between IT and its clients

The job of an IT department is to support the business practice of the organization so people can get their work done efficiently. They must understand the work people do and know how to work with them to make their procedures more efficient with technology. IT departments have the luxury of building for a captive customer base. They know who their customers are and can talk to them directly. Their customers know the system is being built for them; they have often specifically requested it. Close working relationships should be easy to create.

In truth, however, the relationship between IT departments and their customers is often antagonistic. "They can never give us what we want in a reasonable period of time. Everything takes two years and even then it's late," say customers. But the IT developers respond, "Of course it's late. They changed the requirements five times, and then when they saw the system they decided they wanted a whole new subsystem added." Instead of creating a trusting partnership with the customer departments, IT is perceived as constantly failing. The customers—the people actually running the business—end up feeling that they cannot rely on IT to build. Quantitative techniques using predefined questions can identify the market and show designers where it is interesting to explore. Understanding the work of the market requires a qualitative technique that explores the customers' work practice and makes new discoveries about how people work and what they need. The discoveries may then lead to new strategies for addressing the market and new market messages for selling to it. They will confirm whether the identified market will really have a significant impact on the work. Then, quantitative techniques may again be useful to show that the work practice to be supported is sufficiently widespread to make a good business. The two disciplines, marketing and design, build on each other with complementary goals and techniques, to result in a whole-product definition. (Hansen [1997] reports on the effectiveness of different mechanisms for gathering customer feedback in a start-up.)
get anything done in a reasonable time, and IT believes they have to
cover themselves to prove it wasn’t their fault when changing business
needs or desires cause requirements to change.

IMPROVING COMMUNICATION WITH THE
BUSINESS

A common approach to addressing these problems is to work through
a customer representative—someone in the customer organization
who knows the business and has the job of communicating requirements to the designers in IT. Sometimes the representative is a “primary customer”
who still devotes some percentage of his or her time to the real job; sometimes it’s a manager who used
to do the job but doesn’t any longer; sometimes it’s a “customer liaison” who used to do the job but is now working with IT
full-time; and sometimes, as in many government contracts, requirements are communicated by an agency that prevents any direct contact with the end customer at all. Even in the best case, the representative only does personally one of the many jobs in the customer organization. And many IT systems impact work across several
departments; customer representatives usually only represent one. Any “customer representative” has a serious challenge in truly representing all aspects of the customer organizations.

Many IT departments avoid these problems by stationing IT
developers with the customer organization. This certainly succeeds in
making IT more responsive to the customer, but brings a loss of control. The developers easily become focused on short-term problems and solutions—they tend to become the local fix-it man. The structure of the customer’s work and long-term possibilities for improvement are no more visible to IT developers
than to the customer, and without this perspective they, like the cus-
tomer, focus on the immediate and most visible issues. And they are stationed in a particular department, so cross-departmental issues are as invisible to them as to their customers. They are rewarded for producing quick fixes to pressing problems. The usual result is dozens of small applications, each solving a single problem, that do not work together to support the work coherently.
In today's world, the systems that are needed are large and complex. They tie together all aspects of a department's work; they support business processes that cross departments; they integrate a company's systems with those of its suppliers and customers. To address these challenges, both IT and their customers need to step back, out of the day-to-day routine of doing business, to see the implications and possibilities. Design starts with who the designers talk to and where they are situated. When designers sit with the customer, with no time for reflection, the result is narrow, extremely focused designs. As process reengineering becomes more important, being able to envision and support large-scale process changes becomes critical to IT's mission. (Lubars et al. [1993] surveys the definition and use of requirements in different organizations for both IT and commercial systems.)

The role of intuition in design

The methods that IT organizations use to interact with their customers tend to capitalize on unarticulated knowledge or intuition. If the designer's intuition can't be trusted to produce a useful system because designers aren't the people doing the work, get the customers more involved in the design. They may not be able to say exactly what they do or why something is important, but they can say what they do or don't like about a design. Another way to bring intuition to bear is to seat developers with the customers so that their intuition gets trained by proximity. Commercial companies do the same thing when they hire accountants to develop accounting software, or send engineers to work with a customer organization for a long time, or run a focus group to allow potential customers to react to product ideas. They are making unarticulated knowledge available to the design team.

But can people reveal truths about their own work in such a situation? The underlying assumption is that people will say what's important given the opportunity, but people simply don't pay that much conscious attention to how they perform jobs that they do well. Think about how difficult driving was when you were first learning. Getting the steering coordinated with the accelerator and the clutch (if there was one) was awkward and jerky. With increasing skill came increased smoothness and less attention to each detail, until at last the whole process became unconscious.
Now, to teach someone else to drive, the teacher has to recover everything she worked so hard to forget. And driving is a simple, obvious task. How are you to know what aspects of everyday work are important? (Sommerville et al. [1993] describes the importance of understanding unarticulated procedures in the somewhat more important domain of air traffic control.)

Many of the important aspects of work are invisible, not because they are hidden, but just because it doesn’t occur to anyone to pay attention to them. Intuition doesn’t help make these aspects explicit:

An entire project team hangs out in the hallway outside their offices every morning and chats over coffee and donuts. Does anyone on the team know this is a critical project coordination session?

A worker in accounting calls a friend in order processing to gossip and mentions that a rush order is on its way. Does his manager know this informal communication is the only thing keeping the company’s rush orders on time?

Intuition has other limitations in a design process. Intuition is entirely internal—it can’t be shared with other team members. It can only be used as the basis for an opinion. But if my intuition and your intuition tell us two different things, then what? Either we have to argue, with no basis for making a rational decision, or we have to appoint someone else tiebreaker. Intuition comes from personal experience. It’s not clear how to go from experience with one customer, or a small set of customers, and generalize it to a department or market. All these problems suggest that a design process needs to externalize the unarticulated knowledge behind intuition. Given an external representation of customer work, we can validate it, share it, and use it to justify design decisions.

**Contextual Inquiry reveals hidden work structure**

A commitment to making customer knowledge explicit and external isn’t useful without a way to get at all the detail of work experience for all the different types of customers. But as we noted above, many
Contextual Inquiry reveals hidden work structure

common ways of working with customers remove them from their work. Consider trying to teach someone to drive not in a car, but in a conference room. With no pedals, turn signal, or steering wheel, explain what’s involved in making a turn. Try to describe what the road might look like, when to slow down, when to put on the turn signal, when to turn the wheel and how fast. It would be tempting to borrow a pie plate for a wheel and blocks for pedals. But even then, it would be so much easier to take your student out on the road and demonstrate. Yet this is the situation that customers are in—trying to explain their work, in a conference room, to designers who don’t do their work. This is the situation of anyone filling out a survey or participating in a focus group. To reveal all aspects of work practice, when so much of it cannot be articulated even by those who do it, you have to see the work. (Goguen and Linde [1993] evaluates different techniques for the ability to reveal unarticulated needs.)

We designed our field interviewing method, Contextual Inquiry, to address these issues: how to get data about the structure of work practice, rather than a market characterization; how to make unarticulated knowledge about work explicit, so designers who do not do the work can understand it; and how to get at the low-level details of work that have become habitual and invisible. We needed a technique that would allow marketing, engineering, analysts, and customer representatives to work together and share insights. These problems suggested an open-ended, qualitative approach that brings us in contact with the customer’s real work. Contextual Inquiry is such a technique. (Goguen [1996] discusses how social techniques such as Contextual Inquiry fit into the requirements gathering process.)

Contextual techniques are designed to gather data from customers in the field, where people are working or living. Contextual Inquiry is a field data-gathering technique that studies a few carefully selected individuals in depth to arrive at a fuller understanding of the work practice across all customers. Through inquiry and interpretation, it reveals commonalities across a system’s customer base.

Contextual Inquiry is based on a set of principles that allow it to be molded to each situation that a project encounters: context, go to the customers’ workplace and watch them do their own work; partnership, talk to them about their work and engage them in uncovering
unarticulated aspects of work; interpretation, develop a shared understanding with the customer about the aspects of work that matter; and focus, direct the inquiry from a clear understanding of your own purpose. These principles guide the creation of a data-gathering technique to collect the best data possible given the constraints of the situation. We've used these principles to apply Contextual Inquiry in many different ways. However, most of the time, the simplest form is sufficient: the contextual interview.

A typical contextual interview lasts two to three hours. A member of the design team meets the customer at his or her place of work and, after a brief introduction, watches the customer do work of the sort the team is interested in. From time to time, the interviewer interrupts, and the two discuss some aspect of the work just performed. Sometimes the discussion stimulates the customer to pull out a paper, form, or note, and they spend time analyzing the artifact in detail. Using these artifacts to support the conversation, the interviewer finds out about events that took place over a longer period of time.

Afterwards the whole design team works with the interviewer to interpret the results of the interview for the design problem. Any one of the design team, representing any business function (marketing, analysts, development, usability) may have run the interview; during the interpretation everyone shares their insight and perspective. Together, they develop work models to characterize the structure of the work of this customer. (Work models are described in Part 2 and the interpretation session itself in Chapter 7.)

Between 10 and 20 interviews like this, with people who perform widely different roles and work in very different ways, are usually sufficient to define an area of work. People only come up with a few different ways of approaching a task. The work models reveal this structure, showing the underlying commonalities across a wide variety of apparently dissimilar users. In every case we have studied, we discover that the underlying structure of work practice is consistent enough that by the time 10 to 20 interviews have been conducted, we are discovering little that is new.

By grounding the design process in detailed, trustworthy customer data, Contextual Inquiry addresses the major problems of both IT and commercial organizations. Commercial organizations find that...
Contextual Inquiry provides a way for the design team to investigate specific work practice, once marketing has defined a potential product area. It gives marketing and engineering a common language for talking to the customer and sharing their knowledge. IT organizations find that Contextual Inquiry helps them build a new relationship with the customer. It brings them into contact with the customer's day-to-day work and allows them to understand it in a way neither they nor their customer could before. The conversation between customer and interviewer about the customer's work (rather than about the system design) creates a shared understanding and commitment between the groups.

In the remainder of this part, we discuss the structure of the interview itself. We describe each principle in detail and show how the principles drive the form of the interview. We then discuss the practical questions of interviewing in the context of a real project: who to talk to, how to set up the interviews, and how different types of projects need different applications of the techniques. In Part 2, we describe the other side of the interpretation session—work models and how to construct them.
Principles of Contextual Inquiry

The core premise of Contextual Inquiry is very simple: go where the customer works, observe the customer as he or she works, and talk to the customer about the work. Do that, and you can't help but gain a better understanding of your customer.

That is the core of the technique, but we find people are generally happy to have a little more guidance. What do interviewers do at the customer's site? How do they behave? What kind of relationship allows customers to teach designers the depth of knowledge about their work necessary to design well?

In Contextual Design, we always try to build on natural human ways of interacting. It is easier to act, not out of a long list of rules, but out of a simple, familiar model of relationship. A list of rules says, "Do all these things"—you have to concentrate so much on following the rules you can't relate to the customer. It's too much to remember. A relationship model says, "Be like this"—stay in the appropriate relationship, and you will naturally act appropriately (Goffman 1959).

Many different models of relationship are available to us. A formal model might be scientist/subject: I am going to study you, so be helpful and answer my questions; it doesn't really matter whether you understand why I'm asking. A less formal model might be parent/child: I'll tell you what to do, and you'll do it because you want my approval (or else you'll rebel to show your independence). Each of these models brings with it a different set of attitudes and behaviors. Everyone knows what it is like

Use existing relationship models to interact with the customer

Design processes work when they build on natural human behavior
when someone treats us like a child, and the resentment it generates. Ironically, the natural reaction is to behave like a child and fight back. Relationship models have two sides, and playing one side tends to pull the other person into playing the other side. Find a relationship model that is useful for gathering data, and as long as you play your role, you will pull the customer into playing theirs.

THE MASTER/APPRENTICE MODEL

The relationship between master craftsman and apprentice is an effective model for collecting data. Just as an apprentice learns a skill from a master, a design team wants to learn about its customers' work from its customers. Though the model is no longer common, it is still sufficiently familiar that people know how to act out of it. When they do, it creates the right behaviors on both sides of the relationship for learning about the customers' work. We find that people with no special background in ethnography learn how to conduct effective interviews much more quickly by acting like an apprentice than by memorizing a list of effective interviewing techniques. Building on this relationship model creates a strong basis for learning about work.

Craftsmen, like customers, are not natural teachers, and teaching is not their primary job. But they do not need to be; the master craftsman teaches while doing. A master does not teach by designing a course for apprentices to take. Nor does a master teach by going into a conference room and discussing his skill in the abstract. A master teaches by doing the work and talking about it while working. This makes imparting knowledge simple.

Teaching in the context of doing the work obviates any need for the craftsman to think in advance about the structure of the work he does. As he works, the structure implicit in the work becomes apparent because both master and apprentice are paying attention to it. It is easy for the master to pause and make an observation or for the apprentice to ask a question about something the master did. Observation interspersed with discussion requires little extra effort on the part of either master or apprentice.

Similarly, in Contextual Inquiry, team members go to the customers' workplace and observe while they are immersed in doing their work. Like craftsmen, customers do not have a natural motive to make their work happen, learning is easy

When you're watching the work happen, learning is easy
work. Like the driver of a car, customers don't think about how they are working. But they can talk about their work as it unfolds. They do not have to develop a way to present it or figure out what their motives are. All they have to do is explain what they are doing, as does this user of a desktop publishing product:

I'm entering edits from my marked-up copy here . . . I'm working in 200% magnification so I can really see how things line up. It doesn't matter that I can't see all the text in this magnification because I'm not checking for continuity or natural flow of words; I'll do that in another pass later . . .

Even if the master were a good teacher, apprenticeship in the context of ongoing work is the most effective way to learn. People aren't aware of everything they do. Each step of doing a task reminds them of the next step; each action taken reminds them of the last time they had to take such an action and what happened then. Some actions are the result of years of experience and have subtle motivations; other actions are habit, and there is no longer a good reason for them. The best time to unravel the vital from the irrelevant and explain the difference is while in the middle of doing the work.

This holds true for customers as well. They are not aware of everything they do or why they do it; they become aware in the doing.

Once we observed someone sorting his paper mail. He was able to tell us exactly why he saved, opened, or threw out each piece because he was in the process of making that decision.

Another time, a research scientist came to the end of a painstaking series of mechanical calculations, turned to us, and said, "I guess you're surprised that I'm doing this." He was surprised at how inefficient he was, once he stopped to think about it.

But it is not natural to stop your work to think about it; the apprenticeship relationship provides the opportunity to do so.

Talking about work while doing it allows a master craftsman to reveal all the details of a craft. As he works, he can describe exactly what he is doing and

---

1 Polanyi (1958) discusses what tacit knowledge people have available for discussion at different times.
why. When either master or apprentice observes a pattern or principle in action, he can point it out immediately.

Customers who describe what they are doing while doing it, or talk about a prior event while in their work, have the same kind of detail available to them. Every action they take and every object around them helps them talk about the details of their work.

One customer said he would not use a manual's index to find the solution to a problem: "It's never in the index." He could not say what led him to this conclusion, what he had looked up and failed to find. All his bad experiences were rolled up into one simple abstraction: it's not there. But when we watched him looking things up, we could see that he was using terms from his work domain, but the index listed parts of the system. We learned what the problem was and what we could do to fix it.

People sometimes don't even remember how to do their jobs themselves; instead, they depend on the environment and things in it to tell them what to do:

A customer was unable to describe how she made her monthly report. When asked to create it, she pulled out her last report and started filling in the parts. The old report was her reminder of how to produce the next one.

Talking about work while doing it protects the master craftsman and the customer from the human propensity to talk in generalizations that omit the detail designers need. When the work's right there, the details, even details people do not normally pay attention to, are available for study and inquiry.

The apprentice learns the strategies and techniques of a craft by observing multiple instances of a task and forming his own understanding of how to do it himself. This understanding incorporates the variations needed to do the task well under a variety of circumstances. The master craftsman can communicate techniques and strategies without articulating them. By watching instance after instance, the apprentice builds up a big picture of how to do the work.

In the same way, interviewers observing multiple events and multiple customers learn to see the common strategies underlying the work. Once they understand the basic strategies, they can start to imagine solving a problem. Identifying a basic pattern requires that the designer notice by observing complex events, and designing solutions making good sense.

Even a master working in the field and making notes of his observations, is sometimes unable to remember how he did something. An apprentice notices how things are done, and can sometimes even remember how to do it. As the master works in the field he looks over his notes, often to review a basic pattern and structure, or to understand what happened and why.

The master may not remember how he did something, or why. When either master or Apprentice observes a pattern or principle in action, he can point it out immediately.
imagine a system that would support those strategies. For example, a basic pattern in coding is work on the code, test it, and see the results. Identifying bugs to fix leads back to working on the code. But this pattern holds true not only for code, but for creating analysis and design models and automated tests as well. We uncovered this pattern by observing multiple people working on multiple systems of varying complexity. We could then structure the CASE system we were designing to facilitate movement through this cycle. (Part 3 discusses making common patterns and strategies explicit.)

Every event serves as the starting point for discussing similar events in the past. In this way apprentices learn from experience gained by a master before their apprenticeship started. A particular occurrence or task reminds the master of other interesting times this event or task happened. If the event is reasonably close in time, the story is concrete and detailed. It is the retelling of a particular event, told while the master is immersed in doing the same activity with all the triggers and reminders doing that activity provides.2

A design team typically has less time to spend with its customers than the years needed for an apprenticeship. But in the same way that an apprentice can learn from the master's experience, interviewers can learn about events that occurred in the past. Events that occur while the interviewer is present remind customers to talk about events that happened previously. The artifacts of work—papers, forms, notes, clipboards, and so forth—trigger conversations about how they were used, how they were created, and how their structure supported their use in a particular instance.

A customer describing how she learned a feature told us, "I looked it up in the documentation." But when we asked her to look it up again, she was able to show us: "I looked the function up in the index and scanned the section. I saw this icon in the margin that I recognized from the screen, so I read just this paragraph next to it. It told me all I needed to know."

The documentation provided the context she needed to recover a detailed story, and the detail revealed aspects that had been overlooked—that the icon was her visual cue to the relevant part of the page.

1 Orr (1986) describes such storytelling to transmit knowledge among modern-day system managers for similar reasons.
Contextual Inquiry seeks to provide rich detail about customers by taking team members into the field. Once there, apprenticeship suggests an attitude of inquiry and learning. It recognizes that the customer is the expert in their work and the interviewer is not. An interviewer taking on the role of apprentice automatically adopts the humility, inquisitiveness, and attention to detail needed to collect good data. The apprentice role discourages the interviewer from asking questions in the abstract and focuses them on ongoing work. And customers can shape the interviewer's understanding of how to support their work from the beginning, without having to prepare a formal description of how they work or what they need.

THE FOUR PRINCIPLES OF CONTEXTUAL INQUIRY

Apprenticeship is a good starting point, but it is only a starting point. Unlike apprentices, interviewers are not learning about work in order to do it; they are learning about it in order to support it with technology. Interviewers cannot afford to spend the time an apprentice would take to learn the work. Unlike an apprentice, members of the design team contribute their own special knowledge about technology and what it can do. Apprentices learn a single job, but different projects may require the team to study a widely varying work practice—from the surgeon in the operating theater, to the manager in a high-level meeting, to the secretary at a desk, to the family in front of the video game. Designers meet the needs of a whole market or department, so they must learn from many people—individuals doing the same kind of work and individuals doing very different tasks and taking on different roles in order to get the work done.

The basic apprenticeship model needs modifications to handle a design team's needs and situation. Four principles guide the adoption and adaptation of the technique: context, partnership, interpretation, and focus. Each principle defines an aspect of the interaction. Together, they allow interviewers to shape their work to particular needs, helping customers to identify how to use technology to support their work.

Contextual Inquiry tailors apprenticeship to the needs of design teams.

They are all relevant in fleshing out the rich, relevant details of customer work. The customer is the expert; the interviewers serve the needs of design teams by learning from many people and asking about the work, rather than about abstract questions of work. Contextual Inquiry gives over to the customer the most interesting and revealing details of the work. The customer is the expert who can talk about their work in a way that is relevant to the design team, rather than one who has to explain their work in more abstract terms. They also allow interviewers to ask questions that directly address the needs of the design team and project.

SUMMARY

Contextual Inquiry tailors apprenticeship to the needs of design teams.

...
they allow the basic apprenticeship model to be molded to the particular needs of a design problem. We will describe each principle and how to use it in turn.

**Context**

The principle of context tells us to go to the customer's workplace and see the work as it unfolds (Whiteside and Wixon 1988). This is the first and most basic requirement of Contextual Inquiry. Apprenticeship is a fine example of doing this; the apprentice is right there to see the work. All the richness of real life is there, able to jog the customer's memory and available for study and inquiry.

The customer made a phone call in the middle of doing a task. Is this relevant to the work? Was she calling on an informal network of experts to get help in the task? Someone stops by to get a signature on a form. What is the customer's role in this approval process? Do they talk about it before she signs? What are the issues?

Context tells us to get as close as possible to the ideal situation of being physically present. Staying in context enables us to gather ongoing experience rather than summary experience, and concrete data rather than abstract data. We'll describe each of these distinctions in turn.

**Summary vs. Ongoing Experience.** We are taught from an early age to summarize. If someone asks a friend about a movie she saw last week, she does not recount the entire plot. She gives overall impressions, one or two highlights, and the thing that most impressed or disgusted her. (Never ask a seven-year-old that question—they haven't yet learned to summarize and will tell you the entire plot of the movie in excruciating detail.) Ask people to tell you about their experience with a new system, and they will behave just the same way. They will give their overall impressions and mention one or two things that were especially good or bad. They will have a very hard time saying exactly why the good things were important, or why the bad things got in the way. That would require that they be able to talk about the details of their work, which is very hard to do.

We once asked a secretary how she started her day. Her answer was, “I guess I just come in and check my messages and get started.” She wasn’t able to go beyond this brief...
summary overview. It was the first thing in the morning and she had just arrived at the office, so we asked her to go ahead and do as she would any other morning. She unhesitatingly started her morning routine, telling us about it as she went:

"First I hang up my coat, then I start my computer. Actually, even before that I'll see if my boss has left something on my chair. If he has, that's first priority. While the computer's coming up, I check the answering machine for urgent messages. There aren't any. Then I look to see if there's a fax that has to be handled right away. Nope, none today. If there were, I'd take it right in and put it on the desk of whoever was responsible. Then I go in the back room and start coffee. Now I'll check the counters on the copier and postage meter. I'm only doing that because today's the first of the month. . . ."

This person's morning routine has a definite structure: first she checks all her communication mechanisms to see if there is an immediate action that needs to be taken, then she starts the regular maintenance tasks of the office. But this structure is invisible to her. It would not even occur to most people as a topic of conversation.

The job of the interviewer is to recognize work structure. Discovery of work structure arises out of this level of detail about mundane work actions. Summary experience glosses over and hides this detail. Being present while the work is ongoing makes the detail available.

**Avoid summary data by watching the work unfold**

**Abstract vs. concrete data.** Humans love to abstract. It's much easier to lump a dozen similar events together than to get all the details of one specific instance really right. Because an abstraction groups similar events, it glosses over all the detail that makes an event unique. And since a system is built for many users, it already needs to abstract across all their experience. If designers start from abstractions, not real experience, and then abstract again to go across all customers, there is little chance the system will actually be useful to real people. Even in the workplace, customers easily slide into talking about their work in the abstract. But there are signals that indicate the customer needs to be brought back to real life.

If the customer is leaning back and looking at the ceiling, he is almost always talking in the abstract. This is the position of someone who will later be told by him or her that he or she just did something, but gives no indication of what he or she did. Because situations like this are so common, unless the interviewer is careful, the customer will be talking in abstractions before the interview even begins.

"In our company, we're always looking for ways to introduce some degree of personal experience.

The best way is quietly to observe. Eavesdrop on conversations. Note what people are writing, and at what time. If necessary, come back to the written material by email later on.

Use the notes to help the customer focus on the specific instance in hand. If the person is talking about the day before, and you hear the word 'tomorrow,' you can interrupt and ask what he is planning to do today. Perhaps he will sit down and start doing it in front of you.

Some people do not need to be wrenched back to the past and present immediately. You might ask the customer to write down a few things he would like to do in the next month, or to give you a quick rundown of the last week. Or you might ask him to tell you about the most recent experience he had with a system. Or you might ask him whether there is something he would like to change in the system after using it. If the customer is leaning back and looking at the ceiling, it is almost certain he is not going to tell you this.
who will not allow the reality all around him from disrupting the conception he is building in his brain. Someone talking about real experience leans forward, either working or pointing at some representation of what he is talking about. Words indicating the customer is generalizing are another signal. If the customer says, "generally," "we usually," "in our company," he is presenting an abstraction. Any statement in the present tense is usually an abstraction. "In our group we do ..." introduces an abstraction; "that time we did ..." introduces real experience.

The best cure is to pull the customer back to real experience constantly. Every time you do this, you reinforce that concrete data matters, and you make it easier to get concrete data next time. If the customer says, "We usually get reports by email," ask, "Do you have one? May I see it?"

Avoid abstractions by returning to real artifacts and events

Use the real artifacts to ground the customer in specific instances. If the customer says, "I usually start the day by reading mail," ask, "What are you going to do this morning? Can you start?" Return the customer to the work in front of him whenever possible.

Sometimes the work that you are interested in happened in the past and you want to find out about it, so you need to elicit a retrospective account. Retelling a past event is hard because so much of the context has been lost. People are prone to giving a summary of a past event that omits necessary detail. Most people will start telling a story in the middle, skipping over what went before. They will skip whole steps as they tell the story. The interviewer's job is to listen for what the customer is leaving out and to ask questions that fill in the holes. Here is an example of walking a customer through a retrospective account. The customer is talking about how they dealt with a report. We've interpolated the dialog with the missing steps that the interviewer is hearing in the data.

Customer: When I got this problem report I gave it to Word Processing to enter online—

(Why did she decide to give it to Word Processing? Did she do anything first?)

Interviewer: So you just handed it on automatically as soon as you got it?
C: No, it was high priority, so I read it and decided to send a copy to the Claims department.

(How did she decide it was high priority? Is it her decision?)

I: How did you know it was high priority?

C: It has this green sticker on it.

(Someone else made the decision before the report ever got here. Who and when?)

I: Who put on the green sticker?

C: That's put on by the reporting agency. They make the decision about whether it's high priority and mark the report.

(We can better pursue how the reporting agency makes the decision with them; we'll only get secondhand information from this user. Instead of trying to go further backward, look for the next missing step forward: doesn't Claims get a more personal communication than just the report?)

I: Did you just send it on to Claims, or did you write them a note about why they needed to see it?

C: Oh, I always call Claims whenever I send them one of these reports.

At each step, the interviewer listened for steps that probably happened but the customer skipped and then backed the customer up to find out. In this process, the customer walked through the steps in her mind, using any available artifacts to stimulate memory, and recalled more about the actual work than she would if allowed to simply tell the story in order. Using retrospective accounts, the interviewer can recover past events and can also learn more about events in progress. If the end of a story hasn't yet happened, the most reliable way to learn about that kind of situation is to go back to a previous occurrence that did complete and walk through it. Trying to go forward and find out what will happen next forces the customer to make something up; going to another past instance allows the customer to stay concrete.
The key to getting good data is to go where the work is happening and observe it while it happens. Observing ongoing work keeps the customer concrete and keeps them from summarizing. Keeping to the apprenticeship model helps with this; the apprentice wants to see and assist with real work. If the customer starts telling stories, the interviewer can (exerting a little more control than an actual apprentice would) either redirect him to ongoing work or delve into the story, using a retrospective account to get all the detail possible.

**PARTNERSHIP**

The goal of *partnership* is to make you and the customer collaborators in understanding his work. The only person who really knows everything about his work is the one doing it. The traditional interviewing relationship model tilts power too much toward the interviewer. The interviewer controls what is asked, what is discussed, and how long is spent on a topic. This won't get you design data—you don't know what's important to pay attention to, and you don't know what will turn out to matter. The apprenticeship model tilts power, if anything, too much toward the master-customer. It suggests that the customer is in full control, determining what to do and talk about throughout the interview. Traditional apprenticeship would reduce the interviewer to asking a few questions for clarification, at best.

This is too limiting for an interviewer understanding work practice. An interviewer's motive in observing work is not that of the apprentice. Apprentices want to know how to do the work; interviewers want data to feed invention of a system that supports the work. Apprentices are assumed to bring no useful skills to the relationship. Any skills they happen to have they subordinate to learning the way the master goes about working. Designers may not be experts in doing the work, but they must develop expertise in seeing work structure, in seeing patterns and distinctions in the way people organize work. An interviewer has to create something that looks more like a partnership than like an ordinary apprenticeship. This allows them to engage the customer in a conversation about the work, making the customer aware of aspects of the work that were
formerly invisible and bringing the customer into a partnership of inquiry into the work practice.

In one interview with a user of page layout software, the user was positioning text on the page, entering the text and moving it around. Then he created a box around a line of text, moved it down until the top of the box butted the bottom of the line of text, and moved another line of text up until it butted the bottom of the box. Then he deleted the box.

**Interviewer:** Could I see that again?

**Customer:** What?

I: What you just did with the box.

C: Oh, I'm just using it to position this text here. The box doesn't matter.

I: But why are you using a box?

C: See, I want the white space to be exactly the same height as a line of text. So I draw the box to get the height. (He repeats the actions to illustrate, going more slowly.) Then I drag it down, and it shows where the next line of text should go.

I: Why do you want to get the spacing exact?

C: It's to make the appearance of the page more even. You want all the lines to have some regular relationship to the other things on the page. It's always hard to know if it really makes any difference. You just hope the overall appearance will be cleaner if you get things like this right.

I: It's like everything you put on the page defines a whole web of appropriate places for the other things to go.
C: That's right. Everything affects everything else. You can't reposition just one thing.

This is a common pattern of interaction during an interview. While work is progressing, the customer is engrossed in doing it, and the interviewer is busy watching the detail as it unfolds, looking for pattern and structure, and thinking about the reasons behind the customer's actions. At some point the interviewer sees something that doesn't fit, or notices the structure underlying an aspect of the work, and interrupts to talk about it. This causes a break in the work, and both customer and interviewer withdraw from doing the work to discuss the structure that the interviewer found. It is as though they stepped into a separate conceptual room. The customer, interrupted in the moment of taking an action, can say what he is doing and why. The interviewer, looking at work from the outside, can point out aspects the customer might take for granted. By paying attention to the details and structure of work, the interviewer teaches the customer to attend to them also. When the conversation about structure is over, the customer returns to ongoing work, and the interviewer returns to watching. This withdrawal and return is a basic pattern of Contextual Inquiry: periods of watching work unfold, interspersed with discussions of how work is structured.

Over the course of an interview, customers become sensitized to their own work and how it could be improved. Questions about work structure reveal that structure to them so they can start thinking about it themselves. "It's like everything you put on the page defines a whole web of appropriate places for the other things to go." This comment suggests a way of thinking about the work. It makes a previously implicit strategy explicit and invites a conversation about that strategy. Soon customers start interrupting themselves to reveal aspects of work that might otherwise have been missed. Over the course of the interview, a true partnership develops, in which both customer and interviewer are watching work structure, and in which both are thinking about design possibilities. (See Chin et al. [1997] on making customers participants in analyzing their own work.)
Members of a design team also have special knowledge about how to use technology. They notice problems that they can solve and allow them to distract them from the work. They naturally figure out a solution to any problem or apparent problem that presents itself. But this is a distraction from the interview because, rather than listening to whatever the customer is saying, the interviewer is off thinking about the great thing she could make. She can't pay attention to the work while designing something in her mind.

It's not useful to tell designers not to design in the moment—they will anyway. One of the principles of Contextual Design is to work with people's propensities wherever possible. So rather than forbid designing in the moment, we manage it by allowing the interviewer to introduce her idea immediately. The customer is in the middle of doing the work that the idea is intended to support. There is no better time to get feedback on whether the idea works. If the idea works, the interviewer understands the work practice and has a potential solution. If the idea fails, the interviewer did not really understand what mattered in the work. By sharing the idea, the interviewer improves her understanding of the work and checks out her design idea at the same time. In addition, the idea suggests to the customer what technology could do. Customers start to see how technology might be applied to their problem.

Articulating work structure and correcting design ideas during the interview gives the customer the power to shape the way designers think about the work. Any iterative technique (such as rapid prototyping or Participatory Design) enables customers to shape a proposed design. But iterating an existing design can only make small modifications to its structure. That initial structure—the first prototype—was driven by whatever way of thinking about the work that the designer had when she started. A process is truly customer-centered when customers can change designers' initial understanding of the work. Sharing interviewers' initial, unformed ideas with the customer and articulating work practice together allows customers to alter the team's initial thinking, opening the possibility of radical changes in system purpose and structure.

Avoid getting stuck in a rut more familiar to people, which the customer is trying to avoid. You, another expert, have to ask questions that make the silent agenda visible. The customer is the work behind the idea that you are solving with the idea. The customer is the actively proposing expert.

Expressing ideas with the customer helps the customer to get to know the problem. The ten problems are always pairwise opposing, meaning to understand one, you know the other. If you can't help the other, you can't help the customer's way of thinking. System thinking is always ahead of the customer's question. The answer is, always answer if the customer is asking. Help the customer help himself.
AVOIDING OTHER RELATIONSHIP MODELS. The danger in all of this is that customer or interviewer will fall back into more familiar models of relationship. There are many other models available, each with its own set of problems. If you fall into one of these models during an interview, you will pull the customer into the other side of the relationship, prompting behavior that gets in the way of gathering data. If you are aware of what these other relationships are like, you can notice when you fall into them and take actions to shift back into the right relationship. Here are some common pitfalls:

Interviewer/interviewee: Interviewer and customer start to act as though there were a questionnaire to be filled out. You ask a question, which the customer answers and then falls silent. You, anxious that the interview go well, ask another question, which the customer answers and then falls silent again. The questions are not related to ongoing work because ongoing work has ceased. The best solution for this is to suggest returning to ongoing work, which effectively prevents this question/answer interaction.

Expert/novice: As a representative of the design team, you go in with the aura of the expert. You are the one designing the system, with all the technical knowledge. You have to work to get the customer to treat you as an apprentice. The temptation of taking the expert role back is always present, especially when the customer is trying to use a system that you developed. Set the customer's expectations correctly at the beginning by explaining that you are there to hear about and see their work because only they know their own work practice. You aren't there to help them with problems or answer questions. Then, should the customer ask for help (or should you forget and volunteer help), step out of the expert role explicitly: "I'll never understand the problems with our system if I spend the whole time helping you. Why don't you go ahead and do what you would do if I weren't here, and at the end I'll answer any questions that remain." The only exception to this rule is if the customer is so stuck that he will not be able to do any more of the work you came to see. In that case, give enough information to help him find his way out of the problem. Then you'll have to say all
over again that you came to see how he does things and he shouldn't depend on you for answers.

**Guest/host:** Because it is the customer's workplace and the customer is a stranger, it is easy to act like a guest. A guest is polite and not too nosy. A host is considerate and tries to make the guest comfortable by seeing to his needs. Unfortunately, none of this has much to do with doing real work. If you find yourself feeling like a guest, move quickly past the formal relationship to the role of partner in inquiry. This is where sensitivity to culture matters. If the customer won't be comfortable until you've had a cup of coffee, then have it and move into doing work. The relationship should feel like the kind of intimacy people strike up on airplanes, when they tell things that they would not ordinarily share with a stranger. Here, intimacy doesn't come from personal talk; it comes from a shared focus on the work. Move closer. Ask questions. Be nosy. Ask to see anything the customer touches, and get them to tell you about it. You will know you created the relationship you want when the customer says to you, "Come over here—you want to see this." The more you get them to tell you about themselves, the more you will move out of the formal role.

Partnership transforms the apprenticeship relationship into a mutual relationship of shared inquiry and discovery of the customer's work. It retains the close working relationship from apprenticeship while equalizing the power imbalance. This results in an intimate relationship that allows for inquisitiveness about the details of the work. The relationship is maintained by honesty and openness on the part of the interviewer, who reveals insights and ideas as they occur, and guards against allowing inappropriate relationship models that take the conversation off topic and prevent getting good data.

**INTERPRETATION**

It is not enough only to observe and bring back observations. Interpretation is the assignment of meaning to the observation—what it implies about work structure and about possible supporting systems. The language our field uses to describe gathering data for design—data gathering, field research, requirements elicitation—suggests that by implication these are only the facts, on which interpretation is based.

In view of what we have learned, let us return to a question about this fact:

1. Perl requires

   ```
   ref
   ```

2. Perl accounts to a number

3. Perl requires a count any

Which of the above is correct? The interviewer must choose interpretation that differs from the design idea. Interpretation reveals the reasoning behind the design idea. The interpretation of accounts is how the designer can account for the design idea. The interpretation accounts is how the designer can account un...
The four principles of Contextual Inquiry

suggests that what matters is the facts about the work. Good products, by implication, are based on facts. Interpretation says that good facts are only the starting point. Designs are built on the interpretation of facts, on what the designers claim the facts mean. Here’s an illustration:

In working with one user of an accounting package, we learned that she kept a sheet of accounts and account numbers next to her screen. Here are some interpretations of what this fact might mean and what it might imply for our design:

1. Perhaps account numbers are necessary but hard to remember, and all we need to do is make the cross-reference easier. We could put the cross-reference between numbers and names online.

2. Perhaps numbers are unnecessary, a holdover from paper accounting systems, and all that is needed is a way to refer to an account uniquely. We could get rid of account numbers altogether and identify them only by name.

3. Perhaps compatibility with paper systems is necessary, but referring to accounts by name is more convenient. We could keep the numbers but allow names to be used anywhere numbers are used.

Which of these designs is best? It depends on which interpretation is correct; the fact alone does not allow us to choose. The designer must choose which interpretation to lay on the fact. It’s the interpretation that drives the design decision.

Interpretation is the chain of reasoning that turns a fact into an action relevant to the designer’s intent. From the fact, the observable event; the designer makes a hypothesis, an initial interpretation about what the fact means or the intent behind the fact. This hypothesis has an implication for the design, which can be realized as a particular design idea for the system. For example, the second interpretation above starts with the fact (the chart of accounts is kept next to the screen) and makes the hypothesis that this is just a holdover from paper accounting systems. This interpretation, if true, has implications for the system: it doesn’t matter whether the system provides numbers, but it must provide some way to refer to an account unambiguously. This implication can be acted on by requiring
the system to identify accounts through unambiguous names only. This entire chain of reasoning happens implicitly any time anyone suggests a design idea. Usually it happens so fast, only the final idea is made explicit. But the whole chain must be valid for the design idea to work.

If the data that matters is the interpretation, we must have a way to ensure it is correct, and we can only do that by sharing it with the customer. We fail in the entire purpose of working with customers if we do not share and validate our interpretations of their work—the most important data we bring back would not be validated. Sharing interpretations ensures that the work is understood correctly. Sharing design ideas walks the chain backwards; if the idea doesn't fit, some link in the chain was wrong. When it's the customer coming to you with design ideas in the form of wish lists, treat them the same way: walk the chain backwards to understand the work context driving the wish. Understanding the underlying work practice yields much more flexibility in how to respond—many design ideas can spring from a single origin. Understanding and fixing the underlying problem in the work practice can address many design ideas with a single solution. The partnership we have built up with the customer provides a natural context for sharing observations of structure and interpretations of their meaning.

Can you really check an interpretation just by sharing it with the customer, or will that bias the data? Will customers be prone to agree with whatever you say? In fact, it is quite hard to get people in the middle of doing work to agree with a wrong interpretation. It's not at all hypothetical for them because they are in the midst of the work. The statement that doesn't fit is like an itch, and they poke and fidget with it until they've rephrased it so it represents their thought well:

"It's like a traveling office," you say, looking at how a salesman has set up his car. "Well—like a traveling desk," he responds.

The difference between the two is small but real, and people will be uncomfortable until they get a phrasing that fits exactly.
Furthermore, remember that the data that matters is the interpretation of the facts, not the facts themselves. You can't form an interpretation without getting involved with the events, without trying to make sense of them for you. Where an event contradicts your assumptions, you have to inquire and probe, or you'll never be able to replace your current, flawed understanding with one that works. This probing is driven by your expectations and prejudices, yet it is the only way your prejudices can be overturned.

Finally, since customers are not generally experts in seeing the structure of their own work, the interpretation you suggest shows them what to pay attention to. Open-ended questions give the customer less guidance in thinking about their work than an interpretation and result in less insight.

We might have asked a customer who was starting her workday, “Do you have a strategy for starting the day?” Even though the customer just went through the morning routine, she is not used to thinking about strategy driving ordinary work events. The most likely response would be “No, not particularly”—or a blank stare. But if asked, “You check for any urgent communication first, no matter what form it might have come in?” she can compare this statement of strategy to her own experience and validate it or refine it. She might respond, “Yes, lots of things here are time-critical and we have to deal with them right away”—simply validating the interpretation, adding detail but leaving it essentially unchanged. In fact, she responded, “Actually, things from my boss are most important because they are for me to do. Messages on the answering machine or faxes might be for anyone”—refining the interpretation, accepting the broad outline, but adding a new distinction.

Because customers respond to the interpretation in the moment of doing the work, they can fine-tune it quite precisely. Customers commonly make slight changes in emphasis such as those above to make the interpretation exact. They can do this because they are given a starting point that they can compare with the experience they are now having and adjust it, rather than having to start from scratch. In this way, we use the close relationship

Sharing interpretations teaches customers to see structure in the work

Customers fine-tune interpretations
Chapter 3 Principles of Contextual Inquiry

between interviewer and customer to get very reliable data. In fact, it's the only way to get reliable data: if we don't check it with the customer immediately, we take away an understanding that is at least partially made up.

However, interviewers do need to be committed to hearing what the customer is really saying. They may say "no" to an interpretation, but to be polite may not say "no" directly. Here are some indirect ways customers say "no."

"Huh?"—This means the interpretation was so far off that it had no apparent connection to what the customer thought was going on.

"Umm . . . could be"—This means "no." If the interpretation is close, the customer will nearly always respond immediately. A pause for thought means that they are trying to make it fit their experience and cannot.

"Yes, but . . ." or "Yes, and . . ."—Listen carefully to what follows the "but" or "and." If it is a new thought, this is the right interpretation and yours was wrong. If it builds on yours, this is a confirmation with a twist or with additional information. Customers say "yes" by twinkling their eyes at you as they realize your words match their experience or by elaborating on what you said—or by saying "yes" flatly, as if the whole point was obvious.

We ensure the interpretation is true by creating and maintaining the right relationship with our customer. With apprenticeship as the starting point, we create a close, intimate partnership. Partnership is a natural consequence of a contextual interview. For the entire time, we pay close attention to this person, what he does and how he does it, what gets in his way, and everything that's important to him. We take an interest. Most people have never been the focus of so much positive attention or had such an extended opportunity to talk about what they do. They become invested in making sure we get it right—that we see everything that's relevant and that we take away the exact right shade of meaning. The closer our relationship and more invested the customer, the less willing they are to allow us to leave thinking the wrong thing. This is our safeguard that our understanding is true to their experience.

---

**Focus de...**

Focus de... work. Our... created a... should h.... matter at... over what... the master... interview... needs to... relevant to... from the... friends... about—th... thing on... drop with... Taking... whole list... Consider... Our... ties of... follow-

An... the... the... she... she... Each... are "true... what is... Having... viewer w... distinguish the... the...
In fact, it's clear that there are costs to being at least partially interpreting what the customer says, and what the interviewer means by it. Direct ways to address what the customer says, the customer's interpretation, and the interviewer's interpretation, are relatively easy. What is more difficult is determining what the customer means by what he says, because that is dependent on the context in which the customer says it.

Maintaining a collaborative relationship as the customer's equal partner is the only way to approach any interview. The interviewer must pay close attention to what the customer says and how he says it to understand what that means. The customer has control over what matters, and the interviewer must steer the conversation to keep it on track.

Focus
Focus defines the point of view an interviewer takes while studying work. Once the interviewer is in the customer's workplace and has created a collaborative relationship with her, what should he pay attention to? What aspects of work matter and what don't? If the customer has control over what matters, how can the interviewer steer the conversation at all? The apprentice learns whatever the master knows, and the master decides what's important. But the interviewer needs data about a specific kind of work. The interviewer needs to guide the customer in talking about the part of her work relevant to the design. Focus gives the interviewer a way to keep the conversation on topics that are useful without taking control entirely back from the customer. Focus steers the interview the same way that friends steer conversations with each other. The topics the friends care about—the topics in their focus—are what they spend time on. Anything one friend raises that the other doesn't care about is allowed to drop without discussion.

Taking a focus is unavoidable. Everyone has an entering focus, a whole life history defining what they notice and what they don't. Consider three interviewers watching a scientist go about her work:

One interviewer, a software developer, notices the quantities of paperwork the scientist uses to define the procedure she follows, to record her actions, and to report her results.

Another interviewer is more familiar with the lab technology and sees the kind of instruments she has and the problems she has getting them set up and calibrated.

The third interviewer was once a scientist and sees how the scientist moves about her lab, getting out glassware and chemicals and putting them on the bench near the equipment she will use.

Each interviewer sees a different aspect of the work, all of which are "true," but which may be more or less relevant, depending on what is being designed.

Having a focus means that the interviewer sees more. The interviewer who knows that paperwork is important will learn to distinguish the different kinds of paperwork: the method that defines what the scientist will do, the notebook that records her actions for her
experiment, the log books that record calibrations of equipment for the lab, and the formal report of her results. Each of these distinctions serves as the starting point for a new inquiry, pushing the interviewer's understanding of the lab work wider and wider. A focus gives the interviewer a framework for making sense of work.

To ensure the team sees aspects of work important to the problem at hand, we set focus deliberately to guide the interview toward relevant aspects of work. This project focus gives the team a shared starting point, which is augmented by each person's entering focus so they each bring their unique perspective to bear. (We discuss how to set focus for different types of problems in the next chapter.)

If focus reveals detail within the area it covers, it conceals aspects of work that it does not cover. Different people will naturally see different things. Someone who notices paperwork cannot help but notice when papers are being dragged around the lab; someone who never thought about paperwork cannot help but overlook it until his attention is drawn to it. Meanwhile the first interviewer is ignoring physical movement around the lab to get equipment, to the next lab to borrow supplies that have run short, and into another scientist's office to consult on the method used. These aspects of work may be equally important to the design problem. The first interviewer's focus has revealed rich detail in the use of paper, but how can she expand her focus and learn about the other aspects of work? First, we set focus deliberately to give the team a common starting point, an initial way to see the work, allowing them to build their own distinctions and interpretations on that base. Then, we use group interpretation in the cross-functional team to allow team members to learn and take on each other's focus over time and bring their own focus to bear on each other's interviews (we discuss these sessions in Chapter 7). Finally, during the interview, we use intrapersonal triggers—the interviewer's own feelings—to alert the interviewer when they are missing something.

**HOW TO EXPAND FOCUS.** Pay attention to intrapersonal triggers to create a deliberate paradigm shift, from the understanding of the work the interviewer started with to the understanding of work that is real for the customer interviewed and relevant to the design concern. The interviewer must be committed to seeing where an
Equipment for the lab work is set up and distinctions are pushed toward relevance in the interview, pushing the interviewer toward setting different distinctions. The problem is to begin, to start focusing on the entering focus—so they can see how to set focus appropriately.

The cross-examiner’s focus is to see different aspects of the interviewee’s work. They cannot help paying attention to the paperwork and how the interviewer is paying attention is important. The interviewer is ignoring the next lab scientist’s focus; paper work may be a way of entering focus. The interviewer’s focus may be expanded by setting focus in the initial way to see how the interviewer and interviewer are here, and how the cross-examiner is in each other’s work. The interviewee is paying during the conversation personal feelings—

Understanding of work and understanding of work that isn’t fit and changing it, not to confirming existing expectations. Inner triggers are flags telling the interviewer when an opportunity for breaking a paradigm and expanding the entering focus exists. They work because your own feelings tell you what is happening in the interview and how to act to fix it. Here are some triggers to watch out for:

**Surprises and contradictions:** The customer says something, or you see them doing something, that you know is “wrong.” It’s something no one else would do, something totally idiosyncratic. Or else it’s just random; they had no particular reason for doing it. Any one of these reactions is a danger signal. It means that you are—right now—allowing your preexisting assumptions to override what the customer is telling or showing you. The tendency is to let it pass as irrelevant; the solution is to do the opposite. Take the attitude that nothing any person does is done for no reason; if you think it’s for no reason, you don’t yet understand the point of view from which it makes sense. Take the attitude that nothing any person does is unique to them; it always represents an important class of customers whose needs will not be met if you don’t figure out what’s going on. Act like the apprentice, who always assumes a seemingly pointless action hides a key secret of the trade. Probe the thing that is unexpected and see what you find.

**Nods:** The customer says something that fits exactly with your assumptions, and you nod. This is the reverse of the first trigger, and it is tricky. What you are doing when you nod is saying that you can hear the customer’s words, match them with your own experience, and know as a result that everything that happened to you happened to them. Is this a safe assumption? Instead, take the attitude that everything is new, as if you had never seen it before. The apprentice never assumes the master has no more to teach. Do they really do that? Why would they do that? What’s motivating them? Look for the paradigm shift. Look for ways that what they are doing differs from what you expect.

**What you don’t know:** The customer says something technical that you just didn’t understand or is explaining something and you just aren’t getting it. Now what? Are you going to admit your ignorance? Wouldn’t it be easier to research the subject a bit back at the office? No, admit your ignorance. Make the customer go back and take the explanation step-by-step. Treat this as a good opportunity to
step away from the expert role. You are there to learn, and you might as well learn about the technology, too. No one else will be able to tell you better what this individual is talking about. Even if the customer doesn't really understand it either, the extent of their knowledge and misinformation can be valuable for design. Furthermore, if you don't ask, you'll get more and more lost as the conversation continues.

The easiest way to design a system is from your own assumptions and prejudices. Breaking out of your preconceived notions of what the system should be and how it should work is one of your hardest design tasks. Using the customer to break your paradigm intentionally counterbalances the natural propensity to design from assumptions. Triggers alert you to specific opportunities during the interview to widen your entering focus, and the open dialog encouraged by apprenticeship allows you to inquire when you need to.

**THE CONTEXTUAL INTERVIEW STRUCTURE**

The principles of Contextual Inquiry guide the design of a data-gathering situation appropriate to the problem at hand. The principles say what needs to happen to get good data, but the design problem and the nature of the work being studied control the exact procedure to use. Studies of office work can be conducted much more simply than studies of surgical procedures. The most common structure for Contextual Inquiry is a contextual interview: a one-on-one interaction lasting two to three hours, in which the customer does her own work and discusses it with the interviewer. Each interview has its own rhythm, set by the work and the customer. But they all share a structure that helps interviewer and customer get through the time without losing track of what they are supposed to do. Every interview has four parts:

**The conventional interview:** You, as the interviewer, and the customer need to get used to each other as people. Running the first part of the interview as a conventional interaction helps with that. You introduce yourself and your focus, so the customer knows from the outset what you care about and can start with work relevant to the focus. You ask for any uses (if relevant) the work to which you are listening is connected to and that you need the tape record or notecards to ask for anything the customer will say. You ask for any time; don't do it too soon. You may want to begin by asking about the customer's understanding of the new rules. You state your own understanding of how to behave and ask whether the customer wants to be part of training others about the new rules. This should only happen if it's interesting and the customer wants to be part of the training. You have in front of you the tape record, so if you need it, you can ask men to begin it now. You are no longer conventional.

**The contextual interview:** When the customer is doing her work, you introduce yourself and your focus, so the customer knows from the outset what you care about and can start with work relevant to the

**Commit to challenging your assumptions, not validating them**

...
You might be able to tell the customer knowledge and if you don't continues.

Assumptions of what the work is one of customer to balances assumptions.

Explain the new rules of contextual interview— the customer will do her work while you watch, you will interrupt whenever you see something interesting, and the customer can tell you to hold if it's a bad time to be interrupted. Anytime you want to break social norms, it's best to define the new rules for social interaction so everyone knows how to behave appropriately. If you declare “lady's choice,” ladies will ask men to dance and no one feels awkward. Here, you want to create the new rules for the contextual interview, so you state them explicitly. This should take all of 30 seconds, but it's a crucial 30 seconds; if you don't do it explicitly, you run the risk of spending the entire time in a conventional interview.

The contextual interview proper: The customer starts doing her work task, and you observe and interpret. This is the bulk of the interview. You are the apprentice, observing, asking questions, suggesting interpretations of behavior. You are analyzing artifacts and eliciting retrospective accounts. You are keeping the customer concrete, getting back to real instances and drawing on paper when the customer draws in the air to describe something she doesn't have in front of her. You are taking copious notes by hand the whole time; don't depend on the tape to catch everything. You are nosy—after a phone conversation, you ask what it was about. Follow her around—if she goes to the files, you go along and peer over her shoulder. If she goes down the hall, you tag along. If someone comes to the door and looks diffident about interrupting, you tell him to come on in. And, of course, if the customer says she needs a break, you let her
have one. The principles of context, partnership, interpretation, and focus guide your interaction during the interview.

The wrap-up: At the end of the interview, you have a chance to wrap up your understanding of the work she does and her position in the organization. Skim back over your notes and summarize what you learned, trying not to repeat verbatim what happened, but saying what is important about the work, to her and to the organization. This is the customer’s last chance to correct and elaborate on your understanding, and she usually will. Allow 15 minutes for the wrap-up.

Running a good interview is less about following specific rules than it is about being a certain kind of person for the duration of the interview. The apprentice model is a good starting point for how to behave. Then the four principles of Contextual Inquiry modify the behavior to better get design data: context, go where the work is and watch it happen; partnership, talk about the work while it happens; interpretation, find the meaning behind the customer’s words and actions; and focus, challenge your entering assumptions. If all these concepts start to become overwhelming, go back up to the higher-level idea of apprenticeship. You want the attitude of an apprentice; you want to create an intimate relationship in which you and the customer collaborate in understanding their work, using your focus to help determine what’s relevant. That’s enough to run a good interview.
What are we supposed to do?” an engineer asked us. “Knock on people's doors, asking them to let us watch them use our product?” The answer in this case was “Yes, do that.” Not without setting up the visit ahead of time, of course, and there's some planning to do, but in the end it all comes down to showing up and watching. Sometimes the most difficult barrier to introducing a new way of working is people's assumptions about what is or is not “done.”

But once people accept the idea that they are going to do something they never considered a possibility before, they need to know exactly what steps to follow. Otherwise no real action can take place.

We're now ready to discuss the concrete actions that will enable a Contextual Design project to get started. We will deal with team formation in a later section; here, we will describe how to set the focus for a project, how to plan who to talk to, and variations on the data-gathering process that may be required by different problems.

**Setting project focus**

Before you can do useful work, you must define the problem you intend to solve in terms of the work you plan to support. Typically, a project's mission is defined in terms of the solution it will deliver: “an ordering system for all departments,” “the next version of product X,” “an electronic clipboard for doctor's offices.” (As we discussed in Chapter 2, this is the kind of problem statement that is usually given to the project team by marketing or by the internal client.) To figure
out what to do next—who to talk to and what to look for to decide what is important in this domain—the project team must transform this statement about the solution into a statement about the work.

Your initial project focus will usually be too narrow, too much restricted to exactly the work of the tool you expect to build. To see the whole work context and identify opportunities and potential problems, you want to expand the focus beyond tool use. Ask: What is the work we expect to support? How does this work fit into the customer's whole work life? What are the key work tasks? These are the aspects of work to find out about. Who is involved in making the work happen? Who are the informal helpers? Who provides the information needed to do the job, and who uses the results? These are the people to talk to. Where does the work happen physically? What is the cultural and social context in which the work happens? These constrain the interview situation you can set up. These questions will guide you in thinking about how your system fits into your customers' overall work. Use them to identify what kind of people you want to interview, what tasks you want to see performed, and what you want to watch for while you're there. Remember this is a focus, not a checklist. Use it to guide what you pay attention to during the interview.

To expand your perspective on the work, look for metaphors for the work—unrelated kinds of work that have the same structure as the work you want to support. If you are studying online search and retrieval, you can study how people search for physical objects in libraries and grocery stores. This will help you understand the basic structure of finding, independent of technology and content. If you are studying PC maintenance groups, look at taxi dispatch services; the maintainers need to go out on calls without losing contact with a central organization in much the same way that a taxi is dispatched by the central office while maintaining contact with the office and with other taxis. Studying a taxi service would give insight into the problems of maintaining this kind of coordination and suggest different ways of organizing the PC maintenance group. Metaphors like this give you insight into the work you are supporting, suggesting hidden aspects that might be important. Use the metaphor to structure your thinking, and conduct interviews focused on what you need to know how the work is structured.

Designing for context

A project can be approached in many ways. Each one provides a different perspective.

Designing starts with a class of problem and a word or phrase to describe it. Usually, you must also know how the work will be performed before you can begin.

Gather the necessary information to meet the needs of the people doing the work practice. You must understand and use competencies of the people and systems at all times. The design team needs to be able to gather the information they are interested in and to use that information for the design. They must take into account the capabilities of the people and systems in a competitive, effective way. The result is a detailed task analysis that identifies function for the system, for the user, and for the tool.
Setting project focus

I need to decide whether to transform or to build. To see if there are opportunities to expand the work we do, I need to fit into the key work domains to find out where we do the job. Who are the people who do the job, where does the work take place? With a clear statement of project focus, you are ready to apply it to the particular project situation, starting by defining how to gather data. Different kinds of projects will constrain the data-gathering process in different ways: If you are extending an existing system, that system defines the work you need to study. If you are addressing a new work domain, you need to be open in what you study. The kind of data you look for will be driven by the work you plan to support, but also by the goals of the project.

DESIGNING THE INQUIRY FOR COMMERCIAL PRODUCTS

A project in commercial software may be generated in three principal ways. Each different starting point implies a different set of issues and a different way of collecting data.

Designing a known product: A "known product" is one of a class of products that is known and accepted in the marketplace, like a word processor or a spreadsheet. Competitive products are already established. The market has expectations for this kind of product—you must include certain capabilities to be taken seriously. This may be the next version of a product you are already shipping.

Gather data on people using competitive products. You must meet the market expectations they create. Gather data on the basic work practice of the market, whether the customers use competitors, your products, or no automated system at all. Look for the new delimiters: the unrecognized needs aspects of work are currently not well supported.

Designing your product to support these unmet needs will differentiate your product from the rest of the market. If they are important enough, you will define the new field of competition for the next generation of products, just as the formatting capabilities of early versions of Lotus 1-2-3 defined the new ground of competition for spreadsheets. At the same time, gather data on detailed tool use. You want to make sure that you do the expected function just a little better than anyone else. You also want to pay interviews in the metaphor's work domain if it would be useful to know how it really works.

With a clear statement of project focus, you are ready to apply it to the particular project situation, starting by defining how to gather data. Different kinds of projects will constrain the data-gathering process in different ways: If you are extending an existing system, that system defines the work you need to study. If you are addressing a new work domain, you need to be open in what you study. The kind of data you look for will be driven by the work you plan to support, but also by the goals of the project.
attention to what aspects of existing products get in the user's way, and design ways to streamline it.

**Addressing a new work domain:** A new work domain is totally new. It has been created by changing work or life practice (the fitness industry) or new technological possibilities (telecommuting) and is not addressed well by any product. Any new product will change the way people work in the market, and there's no existing product to use as a guide. The danger lies in thinking that because the work will be changed, there's no way to study it. Before spreadsheets were invented, people did the work—they used paper ledgers to chart their accounts. Before word processors were invented, people did the work—they used typewriters. Define the work your new systems will replace, and study it to learn what matters and how it is structured so the market can make the transition to your new products. (This will not stifle any innovation in your products. Both the first spreadsheets and the first word processors were developed through detailed understanding of the people in their prospective markets.) Define the intent people are trying to achieve. Gather data on people achieving their intent with current tools. Look at how they use paper, informal contacts, and whatever else is available to do what they need to do. Look for problems and places where the lack of tools keeps them from trying to achieve their real intent. Use metaphors to think about what may be important in the new work domain.

The new market may be best addressed not by a single product, but by multiple products working together to support the work comprehensively. When we discuss designing the system in Part 5, we'll show how to manage multiple coordinated products.

**New technology:** Sometimes a project seeks to take advantage of a technology that has just become available or affordable. Instead of being tied to a particular work domain, the project is looking for opportunities to use the technology. You may define specific products, you may design alterations to existing products to take advantage of the new technology, or you may discover that whole new markets open up once the technology is available.

Look for analogs of the technology and how they are used in the real world. If you are automating something that already exists, such as sound or text-to-speech, look for places in everyday life where
sound or speech is already used effectively. Look at the context: What else happens when people talk, such as eye contact and nonverbal cues? When is silence important? Look at what the new technology replaces: for example, infrared links replace signal-carrying wires, so where are wires used? Network wires, control pad wires, speaker wires. Look for the underlying metaphor of the new technology and study that: a PDA (personal digital assistant) is like a Day-Timer with smarts, so look at Day-Timers and ask what you could do with them if they were smart. Look at the fundamental new characteristics introduced by the new technology: Wireless links allow moving around, so how is movement important? PDAs are small, so how does size matter? And use metaphors for the technology to get a different perspective of its use. Go to the places where the new technology can make a difference to stimulate your thinking about how it might be used.

Designing the inquiry for IT projects

IT projects tend to be driven by business needs. However, the statement of need tends to focus on the immediate problem as perceived by the customer. Responding only to the stated problem usually results in a patchwork of small systems, each addressing a small part of the work in isolation, and none working well with any of the others. It's often necessary to negotiate the project focus with the customer so that customer needs are met but the resulting system also ties work together. The proper role of IT is to work with the customer to step back, determine the underlying issues that resulted in this problem, and work out a solution that ties the work and the information systems that support it together. IT organizations always want to create and deliver coherent systems that work together to support a business seamlessly. Any new system should be defined to fit into the overall business strategy. Tying the work together means IT organizations always want to be in the business of process redesign. Rather than automating whatever idiosyncratic work practice exists, IT benefits from working with the customer to imagine changes to their process that take advantage of the new technology are used in the real world
of technology. There are three kinds of requests IT usually has to deal with.

**Upgrades:** The request is to add or modify a feature of an existing system. Typically this is called "maintenance" by the IT department. We avoid this term because "maintenance" implies that no new, interesting work happens in this task. In fact, much of IT's workload is in this kind of "maintenance," and much of the improvement or degradation of the information systems taken together is the result of "maintenance" work. So we borrow a term from the commercial vendors and call these "upgrades." The upgrade request is often stated in terms of a design change: "Just make it so I can enter several orders at once." Your challenge is to understand the reasons behind the request and design a solution that fits the need, keeps work practice coherent, and preserves the integrity of the system design. Look at the whole of the work task and related tasks to understand how the change affects the work as a whole. Look at detailed tool use to see what UI mechanisms work and which get in the way. Look for other point requests that can be addressed with the same mechanism.

**New systems:** The problem as stated is to provide a system to support some aspect of the business (e.g., order processing). There is no explicit intention on the client's side to change the way they work in any major way. Introducing a new system to automate the inefficient ways that things are done currently is a waste. The challenge is to move the design team and the client together to invent ways to improve the work. The result will be to define new ways of working and the software systems that support them. Expand your statement of focus by looking at the whole work process that the original request is a part of. How does it support the real work of the department? If this is the primary intent of the process, look at how the intent is accomplished. If not, ask what the intent is and whether it can be accomplished in a more direct way. Is the process contained in one department, or does it span departments? Plan interviews with people at each point in the process.

**Process redesign:** The project is started to implement a business process reengineering directive. Typically the directive does not specify exactly what will be improved. Many requests for process and claims-handling improvements have settled on the common "just make the process cheaper..." people who have observed the process with their own eyes suggest that "redesign the process to do what the need, and not just reduce the cost of doing the work in the new way of implementing it. The redesign should include requirements for improvements in process changes.

The redesign of the process suggests that you should spend time to converge on a detailed plan for what you are going to go. The plan should include the following sections needed:

### Design a Situation

Your initial situation observation will also reveal the overall flow of the work, as well as the interviews needed, and the tasks and interactions between people. Decide which people have the special demands of the process.

Non-IT employees should be thoroughly consulted.
CD develops the details of business process redesign

**Designing the interviewing situation**

Your initial inquiry into the work gave you a focus for the project and also revealed some characteristics of the work domain and told you what work tasks you need to observe. Exactly how you will set up the interviews is driven by the nature of these tasks. The key questions for defining the interviewing situation are always: How do I get close to the work? How close can I get? How do I create a shared interpretation with the customer? Different kinds of tasks make different demands on the interview.

**Normal:** A normal task can be planned, is performed in a reasonably continuous session, and can be interrupted by the interviewer.
Writing a letter, delivering mail, installing software, and writing code are all normal tasks. The interviewer can plan to be present to observe a normal task and can interrupt at will to understand it. Normal tasks can be studied through a standard contextual interview. It may be useful to ask the customer to save work of the sort you want to study to do during the interview. This does alter the normal workflow, but very minimally, and the increase in relevant data makes it worth it. Audiotape these interviews, but videotape is rarely worth the extra trouble. Videotape them only if the work is so UI-intensive that you have to see the interaction to understand what's going on, or if it's especially important to communicate the customer experience to developers who can't go on interviews themselves.

**Intermittent:** An intermittent task happens at rare intervals over the course of a day. It cannot be scheduled and does not last long. It's so infrequent that the chances of observing it during a standard contextual interview are low—you'd spend hours to get five minutes of data. Looking something up in documentation and recovering from a system crash are intermittent tasks. The key to learning about them is to create a trail that will enable the user to re-create a retrospective account of the event. In documentation, you could ask the user to keep a paper log of every time they use the documentation, perhaps numbering the pages themselves so they can walk through the story later. You could design the documentation so the user can keep their log right in the documentation itself. You might instrument online help, so the software automatically records what the user did. Start with a face-to-face interview, then leave them to log what they do. Return later to perform an interview that follows the form of a retrospective account, walking through each artifact in turn to discover what the user did.

**Uninterruptable:** Some tasks simply cannot be interrupted to do the interpretation. A surgical operation, a high-level management meeting, and a sales call are all situations that cannot be stopped to talk about what is going on. In these situations you want to capture the events clearly enough that you can recall all the details later. You might plan interruptions, such as providing for regular 15-minute breaks in a long meeting where participants can discuss what has happened so far. You might view some of the user's interactions as too intrusive to stop or altered to suit the customer's need, and you might need to say so. You lose the user's trust if you reveal they did not observe what was going on.

**Extensive:** Major software projects have all tasks that involve working in teams for three hours or more. In these situations, understanding the processes is the key first, interviewing can be done at key points in the process. You might record the process with a video camera to capture patterns which are infrequent in the customer's work, but which are typical of the process. Include preparation and installation. You also process the data back and forth concerned with the processes, the creation of the artifact, and the improvement in the product.

**Extensive:** Major software projects have all tasks that involve working in teams for three hours or more. In these situations, understanding the processes is the key first, interviewing can be done at key points in the process. You might record the process with a video camera to capture patterns which are infrequent in the customer's work, but which are typical of the process. Include preparation and installation. You also process the data back and forth concerned with the processes, the creation of the artifact, and the improvement in the product.
discuss what happened in the part of the meeting just concluded. You might videotape the event, then review the videotape with the customer, stopping to discuss events as they occur. If even videotape is too intrusive, you can at least keep good notes and review them with the customer. If you videotape, interpret the tape with the customer. You lose too much insight and cannot be sure of your interpretations if you review the tape alone later.

**Extremely long:** Some tasks take years to complete. Shipping a major software system, developing a new drug, and building a 747 are all tasks that take substantially longer than the two to three hours of a typical contextual interview. To understand tasks of this sort, pursue two strategies: first, interview a wide range of users at different points in the process and playing different roles in the process. Since work strategy repeats, common patterns will emerge even though the cases are different. Then, choose willing customers with the best examples and do a work walkthrough, which is like an in-depth retrospective account. Set up an event in which customers bring in project documentation from all parts of the process and walk through the history of the project, week by week, meeting by meeting. Use the project artifacts to ground the inquiry. Include project documents, such as plans, reports, and designs, and also process documents, such as the calendars and email of those most concerned. Use the artifacts to drive the conversation. Expect this recreation to take a day or two.

**Extremely focused:** Sometimes the problem is so focused on the minutia of a person’s actions that it’s too hard to run a standard interview. You might be polishing the detailed interaction of a computer user with an application’s UI or studying the details of how a craftsman manipulates his tools. You would miss too much if you depended on unaided observation, and you would also get in the way of the work too much if you interrupted every moment. This is a case where videotape can be useful. It will capture the details you would miss, and you can run it repeatedly until you understand a particular interaction. But view it and interpret what you see with the user. You cannot understand all their motivations on your own.

**Internal:** Sometimes the inquiry needs to focus on internal mental processes, such as how decisions are made. In this case, the interviewer...
must be present when the mental process is happening because there's no way to recover enough in a retrospective account. You may need to create events that will cause the mental process to happen so that you can be present. Then interrupt a lot; make a lot of hypotheses about what the customer is taking into account in their thinking. Warn the customer this will be very disruptive, but as long as the customer has to make the decision, they will keep working through it and you will learn something about how they do it.

**Deciding Who to Interview**

At this point you know what you are looking for and you know how to set up the interview for the tasks you need to observe. Now you must start putting names on the customers you will visit. In general, you want to interview two or three people in each role you identified as important to the focus. You want to collect data from 10 to 20 people in all, unless the focus is very narrow. Six to ten interviews is sufficient if there is only a single role or you are studying detailed UI interaction instead of overall work process. If you are making commercial software, you want to go to at least four to six businesses to see variety. In choosing sites and individuals, go for diversity in work practice. You are looking for the common underlying structure that cuts across your customer base. You will do this best by studying very different customers, rather than studying similar customers to confirm what you learned.

Diversity in work practice usually is not equivalent to diversity in market segment. Financial institutions, high tech, and retail may be different market segments, but office work is done very similarly in any modern corporation. These different types of companies will not give you substantially different perspectives. In fact, office work is so similar it is actually hard to get a different perspective. One design team studied the military and Japanese companies, in an attempt to find cultures that would be substantially different; they found little that was new. To get different work practice, look for different business strategies (doing the work as a business for hire vs. doing it as a department in a large company). Look for cultural differences (a trucking company vs. a high-tech company).
Deciding who to interview

When you must interview people, you must prioritize who you interview. It is a good idea to interview those who you believe have the most to contribute. If you are interviewing one person who is an internal member of the team, it is important to include people from other companies who have similar work practices. This will help you understand the work process better and make it easier to design a system that is efficient and effective.

To find the right people to interview, you can look at the different places where the work is done. This can help you understand the different situations that exist, such as a company distributed across several states vs. a company located at a single site. Look for differences in scale (a small business vs. a large corporation). If your customer is internal, see if you can study similar work practices in other companies. Look for other places in your own company where similar work is done, and study it. Use metaphors to give you different ways of thinking about the work.

Given these parameters for numbers and diversity, choose the people you will interview. It's okay to be smart when choosing—include the important client who has to buy into an internal project. Focus on customers from the key markets you think are most likely to spend money.

Expect setting up customer visits to take a couple of weeks, by the time you've found the right person to interview, talked to all the people who are affected, and have set everyone's expectations correctly. However, don't get too far ahead in lining up the visits. As you study the data, you will change your idea of what to find out about next. You don't want to be locked into studying ten documentation writers after you've studied three and discovered that, for your purposes, they all work in much the same way.

Make sure you talk to the people you will interview individually in advance and that they understand what will happen.

Your inquiry into the work that the project supports will yield lots of detail about the work and what to look for. It will be too much for anyone to keep track of during an interview. So boil it down to a short statement of the key characteristics of the work. This statement can be written by interviewers in their notebook and will keep them on track during an interview. A focus for an ordering system might be, "how people learn about what is available, through catalogs, friends, and local experts, whether formal or informal; who is involved in the decision and how they come to agreement; what processes have to be used to make the request and who gets involved in filling it." The initial focus will be revised and expanded through inquiry into the work. (In the above example, the team discovered that it matters to people to track the requests they have made and when they are expected to be filled.)
to be filled.) Focus statements are best when they use simple language. People looking for “requests” will think more broadly about what a request might be and how it might be filled than people looking for a formal-sounding “order.” The result will be greater insight into the work and consideration of a greater range of possible solutions.

**Making it work**

For commercial software and internal systems alike, the crucial first step is to ground the design in relevant customer data. This part of the book has given you a solid grounding in the basics of setting up and running a successful interview. This way of collecting customer information is new, and most organizations do not have the procedures in place to make scheduling these interviews easy.

The groups that have the easiest time are those who already create events with individual customers, such as usability tests or focus groups. There can be internal resistance, too. The sales force, marketing, or the internal customer representative can be suspicious of letting engineers talk directly to customers. (See Chapter 20 for strategies on dealing with resistance.) But reactions to the visits are nearly always enthusiastic. Customers feel like they are being listened to for the first time, and the sales force and marketing soon come to recognize the benefits. When the customers are internal, they feel like they have control over the new system. Teams developing custom software often do more interviews than strictly necessary to allow everyone to participate.

As with all skills, experience comes with practice, but you need neither experience nor practice to get started. Whether you are working on the initial requirements for a large system or are refining the UI of a small system, you can define a data-gathering strategy appropriate to your project. A few interviews run along these lines will return a wealth of data on the customers you serve and the work they do. Increased interviewing skill will come with experience.

But be warned: it’s addictive. People who get used to having contextual data when they design often have a very hard time breaking the habit.
A Language of Work

For customer-centered design, the first task of a design team is to shift focus from the system that the team is chartered to build and redirect it to the work of potential customers. Work, and understanding work, becomes the primary consideration. But "work" is a slippery concept. What is work? You could keep a log of each action I take throughout the day. Is this work? I talk to a colleague and agree on who will handle which parts of a writing task. Is this work? I worry about the latest merger and whether my job will be cut. Is this work? I get up, walk down a corridor, up a flight of stairs, and into a locked room to get a printout, only to discover that the print queue is hung and I will have to restart it from my office. Is this work? If you want to know about work, what do you pay attention to?

This question is particularly acute for a design team. Any system is the result of agreement between engineering, marketing, customers and customer representatives, documentation, and testing. If these disparate people are to use their different disciplines to contribute to the system, they must come to a shared perception of how customers work. Putting work experts—psychologists, anthropologists, or domain experts—on the team helps, but they need to learn how their unique insight contributes to system design. Other members of the team may not be experts in understanding work practice, how it is structured, and how it hangs together as an organizational and social whole. As we discussed in Part 1, what people see and talk about is constrained by their entering focus—by what they have concepts for. If they have no concepts for work, they will talk about things familiar to them: the technology they can use to build the system, its internal structure, and its user interface. To take best advantage of techniques such as Contextual Inquiry, people need to learn
concepts that show them what to see when they are with the customer—and given the constraints of real engineering projects, they need to learn these new concepts quickly.

**Using language to focus thought**

A formal language for talking about work organizes concepts that help people learn to see work. It is natural for people to embody a new domain of knowledge in a language that expresses ideas in that domain. A language makes the key concepts of the domain concrete in symbols or words. This is what jargon is—specialized words and specialized uses of ordinary words that embody concepts useful to some domain of expertise. So knitters create “purl” to describe a stitch, use “knit” to describe another stitch (as well as the whole activity), and when they “cross stitches,” they do something quite different from the “cross-stitch” in needlepoint. In the same way, mechanics use “ping” and “knock” to describe specific symptoms common to engines, confusing those of us who think those words just represent noises.

A specialized language of this sort creates a focus—a set of things to pay attention to. Expertise about the knowledge domain is captured in the language and becomes available to anyone who learns the language. Once you know that engines ping and knock, you can start to make sense of the noises your engine makes. The language gives you a way to see—a framework for interpreting the things you observe and a structure of understanding you can elaborate as you learn more. (Once you know about ping and knock, you can ask what other noises your engine makes. Do they provide more clues to potential problems?) The new language expands the team’s entering focus to include work concepts, enabling team members to see more of the details of work when they interview. And just like the interviewing focus, team members can expand on their language of work, creating new concepts and distinctions unique to the work domain they are designing for.

Because a language creates a focus, it is not neutral. It directs your thought. Any language is designed to say certain things easily—the
Graphical languages give a whole picture

Languages don't have to be textual. Graphical languages—formalisms or diagramming techniques—share all the advantages of a textual language. Instead of words, graphical languages use symbols, each conveying a defined concept. Just as syntax rules restrict how words can be combined in a textual language, drawing rules restrict what can be drawn in a graphical language.

For design, a graphical language has definite advantages. Because the number of symbols in a graphical language is small—usually less than 100—a graphical language focuses thought even more intensely than a textual language with thousands of words. It is possible to learn all the symbols of a graphical language, and once learned they suggest how to use them. Just as when you learn a new word, you suddenly notice the word used everywhere, the symbols of a graphical language cause a design team to notice the distinction they represent. They become part of the design focus, revealing more detail about work.

Unlike a textual language, graphical languages let you take in a whole picture at once. A textual language must be read and parsed; this is not only a difficult chore, but the information has to be taken in sequentially, one idea at a time. Given reasonable methods for handling
complexity, a picture can be scanned and taken in as a whole. A picture is a better external representation than a page of text because it's easier to see what you are talking about. A picture reveals overall pattern and structure by showing each part in relationship to the whole. This is critical to creative work and to design (Suchman 1989). Once a team understands how work fits together, they can identify sets of problems and needs to address together. Without a coherent understanding of work, each need stands alone and can only be addressed as a point problem. It's impossible to see when a solution to one problem creates new problems elsewhere—just as automated phone systems solved the problem of giving quick answers to standard questions, but made it difficult to get to a live person to deal with non-standard situations. A diagram supports systemic thought and makes it possible to create a coherent design response that fits well with the work it supports. (Hutchins [1995] discusses how artifacts support and enhance thought.)

**WORK MODELS PROVIDE A LANGUAGE FOR SEEING WORK**

For these reasons, we use **work models** as a graphical language to capture knowledge about work. They provide a shared focus on work that gives the team an external, concrete form to record and communicate what they saw on customer visits. As long as work practice remains insubstantial and invisible, there's no good way to share what you learned, to validate your understanding with the customer, or to check that your design really accounts for the work practice you discovered. Models make concepts concrete, creating a physical artifact that the team can share, talk about, and touch. Teams can use them to understand what each team member is really saying about the work. If the team includes work experts, models give them a way to make their insight explicit and communicate it to the rest of the team. The team can share their understanding with customers to ensure that it is correct. And designers can check the models to ensure they are not forgetting some aspect of the work that will cause their design to fail. Creating

---

**Writing things down is a central tool for creativity**

---

...
Work models provide a language for seeing work

Concrete artifacts is critical to creativity—it’s a cliché that great designs are first recorded on the back of a napkin. Models provide a way for people to record their thinking so it can be seen and manipulated.

By providing a coherent, synthetic view of work practice, work models give design teams effective ways to handle qualitative data. Any qualitative technique such as Contextual Inquiry produces huge amounts of detailed knowledge about the customer. This knowledge is critical to system design, but it isn’t amenable to reductive statistical techniques: you can’t take the average of 20 interviews to identify the “typical” customer.

Graphical models organize huge amounts of data

Work models provide a coherent way of structuring all this detailed data, revealing underlying structure without glossing over the detail.

Graphical languages do exist already in systems design. Process flows, state transition diagrams, object models, data flow diagrams—all use graphical languages to represent some aspect of system design. Each, by the concepts it presents, focuses the designer on a certain way of thinking about the problem. But few of these diagrams focus on people and how they work. A data flow diagram focuses on the flow of data, and the operations performed on it, independent of the people involved (Yourdon and Constantine 1979). A process map shows processes and tasks, but not how they map to a person’s responsibilities or environment. An object model shows things—objects—and the operations that the objects perform or their responsibilities.

It is logically possible to use a technique like object modeling to represent other concepts, but in practice it can’t be done. It’s like saying that all programs could be written in machine code so any other programming language is unnecessary; it’s logically true, but actually writing any of today’s systems in machine code would be so overwhelming that they would never be written. In the same way, you might represent all the aspects of work in an object model, but the conceptual task of interpreting the model would be overwhelming. You would have to inquire into each object and interpret what it said about the work. The work of people is still invisible. Furthermore, the language of work needs to focus on the concepts of work that matter; generic object modeling cannot provide a focus. It cannot guide our thoughts. (Sumner [1995] provides further research into designers’ use of multiple representation.)
Instead, design teams need a representation of work that makes the important aspects of work for design apparent. The models will stand in for seeing the work itself; once team members are familiar with them, the team should be able to look at a model and envision people doing the work it represents. A mental translation from the distinctions in the model to relevant work concepts gets in the way.

**Work models reveal the important distinctions**

Contextual Design provides five different types of work model to represent customer work practice: flow, representing the communication and coordination necessary to do the work; sequence, showing the detailed work steps necessary to achieve an intent; artifact, showing the physical things created to support the work, along with their structure, usage, and intent; culture, representing constraints on the work caused by policy, culture, or values; and physical, showing the physical structure of the work environment as it affects the work. (The next chapter describes each in detail.) Each type of model provides its own perspective on the work and synthesizes all aspects of work in its focus into a single, coherent diagram. Having multiple types of work model gives a team more ways to see issues and structure in the work, while allowing each model to focus cleanly on one aspect of work.

We find that these five models are usually sufficient to support all the design conversations a team needs to have—the combined focus they provide covers the main issues for most design problems. As we will see, they support the chain of reasoning from data to design. As with any focus, the work models both reveal detail in the areas they cover and conceal detail that falls outside. When it's necessary to expand a focus to explore issues that the work models do not cover, having work models suggests that new models might be created. For example, though the flow model shows the overall coordination between people, it does not show the continuous give-and-take between two people collaborating on a project. It also does not show what is going on interpersonally between people.
Work models reveal the important distinctions over the course of a conversation. When a design problem requires understanding these or other aspects of work, we create new models to show them explicitly. (See “Readings and Resources” for other approaches to modeling work.)

After interviewing each customer, the team runs an interpretation session to recapitulate the interview and record what they learned (interpretation sessions are described in Chapter 7). During the session, they draw work models relevant to their project focus. Once a team has generated a set of work models for each customer interviewed, they can use the models to look across customers and identify common patterns and structure. This is the basis of our consolidation process, which takes a team from the work of individual customers to understanding the work of a whole market or department. Since any system will be used by multiple people, this is a critical step in design. Without an explicit way to build a representation of how potential customers of a system work, the design team must generalize in their heads from specific instances. The models make this an external step that can be communicated, shared, and validated. The final consolidated models are the basis of design—the single statement of the work practice that must be supported, improved, replaced, or obviated if a new system is to be successful.

We’ll discuss the consolidation process in Part 3. In the following chapter we’ll discuss each work model in turn and then describe the interpretation session in Chapter 7.
Each of the five types of work models has its own concepts and symbols representing one aspect of work for design. The five models were developed over time to meet the needs of the design problems we encountered. They represent the key aspects of work that design teams need to account for in their designs. We have found these five to be necessary to almost every problem and sufficient for most.

Work models are first built to describe work from the point of view of the one person interviewed. They do not and are not intended to represent everything that a person or his organization does. Each interviewer learned about some part of the customer’s work as it related to the project focus. They also learned something about the work of the organization, as understood by this one customer. The first models we build represent this individual perspective. We even use conventions to show which parts of a model are built from the customer’s actual experience and which represent the customer telling us how his organization is supposed to work.

**The Flow Model**

To get work done, people divide up responsibilities among roles and coordinate with each other while doing it:

A rush order comes in. The woman who receives it calls the person responsible for filling it and mentions, in passing, that a rush order is on the way. The rush order will be shipped on time only because of her informal advance warning. When a new order-processing system is introduced, it does not allow this advance warning and rush orders start shipping late.

A purchasing department is responsible for paying invoices as they come in. But they don’t know if the goods were
actually received; they have to figure out who received the goods, send the invoice to him for approval, and pay it only when he returns it signed. Making the purchase and paying for the goods have been separated from the actual work of the organization. Formal sign-off and review processes keep the system working. The purchasing department gets so involved in maintaining these formal processes that they cannot handle finding vendors and making purchases well.

A specialist in another organization gets ready to produce a report. In times past he would have had a secretary type in and format the report; these days he not only creates the content, but he also defines the formatting and layout, checks spelling, and proofs the document as well. He has more control over the document in his own hands, but it's not clear that it's cost-effective for a highly paid professional to do basic stenographic and editorial tasks.

In each of these cases, the key issue is how people's roles are defined and how they communicate to get the job done. The order receiver had to communicate with the order processor to get rush orders accomplished on time; the invoice payer had to communicate with the goods user to find out if the invoice should be paid; the content provider became the page designer, instead of handing the content off to a secretary who could have played that role. All work in this world involves other people to some extent. Books are written for an audience, based on sources, submitted to reviewers, and passed to publishers. Code is written by developers for its users, from requirements, tested by a testing group, marketed by a product marketing group, and distributed to customers. Departments exist because a single person alone can't get the work done; the work must be broken into parts, which then must be coordinated. Different departments coordinate the different parts of the work, and people within a department coordinate to get its work done. The flow model represents this communication and coordination necessary to make work happen.

**Recognizing communication flow**

*Work flow* (Figures 6.1 and 6.2) defines how work is broken up across people and how people coordinate to ensure the whole job gets done.
The flow model

**Flow Model Distinctions**

The **individuals** who do the work. In the consolidated models, the roles they play (see Part 3 for a discussion of consolidation). Each person or group is shown as a bubble. The interviewee’s bubble is annotated with user number and job title. Everyone else’s bubble just has job function.

The **responsibilities** of the individual or role. This is a list of what is expected of them—“coordinate schedules of all managers,” “ensure samples are processed in the shortest possible time.” Every bubble and place on the flow model is annotated with responsibilities.

**Groups**, sets of people who have common goals or take action together. Outside people may interact with the group as a unit, without knowing any individuals in the group. They say things like “I sent it to purchasing”; the particular person in purchasing doesn’t matter. Groups are represented when a person has the same interaction with all its members. We may also show the interaction between a group member and the group as a whole.

The **flow**, the communication between people to get work done. Flow may consist of informal talk and coordination, or it may consist of passing artifacts. Flow is shown as arrows between individuals.

**Artifacts**, the “things” of the work, which are thought of and manipulated as if they were real. An artifact may be physical, such as a document or message. It may also be conceptual; for example, if a design conversation is thought about as though it has members, a history, attributes (public or private), and an existence separate from any one member or topic, it may warrant representation as an artifact. Where appropriate, the **mechanism** is shown—email vs. paper, for example. Artifacts are shown as small boxes on a flow.

The **communication topic or action** representing the detail of the talk or coordination represented by a flow. These are actions as opposed to artifacts, such as talk to set up meetings, arranging for review, asking for help. Examples might be “question about the system” or “request for help.” Communication is written on a flow without a box.

**Places** that people go in and out of in order to get their work done, if it is central to the work of coordinating and collaborating. This is often a meeting room or communal space such as a coffee area. It is shown as a large box annotated with name of place and responsibilities.

**Breakdowns** or problems in communication or coordination, represented as a red lightning bolt (black in this book).

How do job responsibilities get assigned to people? What are the different roles people take on to get work done? How do new tasks get passed to a person? Who do they get help from? Who do they have to work with to accomplish their tasks? How do they use physical places and artifacts to help them coordinate? Who do they give the results to and in what form? Work flow is the rich pattern of work as it shuttles between people, the interweaving of jobs and job responsibilities that gets the work done. Work flow represents every phone call between
Requires lots of iterations
Two people, every document passed for review, every email message, every conversation between people in the hall. These are all instances of passing an artifact, communicating information, or coordinating to
The flow model

FIGURE 6.2 Creative work. This flow model is typical of creative work. We see communication with those who depend on the work and with those who assist in the creation. But most of the interactions are focused on the task of creation. Compared with the "hub" type of job, this work is much more continuous and coherent.

do a job, whether as part of a formal process or as an informal way to get the job done.

When people coordinate through email or paper, it's easy to see. It's harder to see how casual conversation and handwritten notes support work flow. Here's what to watch for in an interview.

COORDINATION. Any artifact received or handed on indicates coordination with someone else. Where did it come from? Who created
it? Who will see it next? Find out the whole story to see how the work fits together. Any discussion with someone else, through a phone call, email, or by dropping in personally, also indicates coordination. Is this discussion critical to the work? Where are the problems in coordinating? Do people forget? Do they spend a lot of time on it? Look for opportunities to automate communication that is currently manual and haphazard. See if you can eliminate the need for coordination by providing information directly or by combining roles that are currently separated.

**Strategy.** What strategy is implicit in how the roles are organized? Listen to how the customers talk about their job. How do they see themselves in the organization? What do they consider to be their unique contribution to the department? What is the unique contribution of the department to the company? How does it further the business? Ask whether the role is really critical to the business. If not, why was it put in place? Could that intent be accomplished more directly, or is the intent irrelevant to the business? (One purchasing department has a role devoted to providing PO numbers. PO numbers support their process, but give no direct benefit to the business.)

**Roles.** What makes a coherent role? Watch the tasks people do. How do they hang together? Which tasks require similar knowledge, tools, procedures, or data? When does doing a task require knowledge of the progress made in doing another task? These tasks tend to be performed by the same role. Technicians, for example, need to know the history of a problem and of prior attempts to fix it in order to serve the customer well. If problem calls are handed out to the first available person, regardless of history, service will be poor.

**Informal structures.** Look at the ways people go beyond the formal structure: A secretary becomes known as the expert on creating forms. Soon whenever anyone has a particularly difficult form to create, they pass it to her and she does it for them. A scientist has special instructions to communicate to her lab technician. She writes a note on a materials tracking tag, knowing he will see it. A manager requires a status report. In a small organization, the person taking these positions might not even be aware that they are doing it. They might be operating outside of the official structure, but their behavior affects other people. Their actions make a difference. They can, however, be managed—specifically, managed by those people operating within the formal structure. An informal structure is one that is not part of the formal structure but is still an essential part of the work.
The flow model offers a bird's-eye view of the organization, showing the people and their responsibilities, the communication paths between people independent of time, and the things communicated—either tangible artifacts or intangible coordination. People and organizations are bubbles on the model, annotated with their position and responsibilities (roles are not represented directly until we consolidate models across people). Flow is indicated as arrows between bubbles, with the kind of communication written on the line. Artifacts are shown in boxes on the line; informal communication and actions are written without a box.

Where places such as meeting rooms or virtual places such as shared areas support communication, the flow model shows them as well. When a place is important to coordination—meeting rooms, bulletin boards, and shared drop-off areas—they appear as large boxes at the end of a flow. Just as individuals are annotated with their responsibilities, places list their responsibilities in supporting communication and coordination. Automated systems and databases usually should not go on the flow. The only exception is when they are acting like a physical place or like an automated person, and they are critical to coordination between people. Then they are shown as a large box with responsibilities.
When communication breaks down—people don't get something they should have received or don't respond when a response is needed—we show the problem with a lightning bolt.

Do not limit the model to the formal definition of how work is supposed to be done. The defined process of the organization is not a good guide to how work is actually accomplished. Every day, the people in the organization design how their jobs will really be done. As they encounter problems and obstacles, they create solutions, and the solutions become part of the real work. The flow model needs to capture how work is really done, including all the informal interactions that make it work. From this representation, you can find good work practice to incorporate into a system, identify problems to eliminate, and see the pattern of communication a system must allow for.

THE SEQUENCE MODEL

Work tasks are ordered; they unfold over time. But the steps people take aren't random; they happen the way they do for a purpose:

A man reads a mail message and, after replying, saves it in a folder called "Phone book." He'll never need that message again. He's just saving it because it has the sender's telephone number on it, and it's a convenient way to look it up. So telephone numbers matter even when email is the primary form of communication, and telephone calls may be triggered by email. Anyone trying to build the complete personal organizer can build on this to tie phone contacts and email together.

A woman paying her bills first gets out her checkbook, bills, paper record of accounts, envelopes, and stamps; then records the amount of every bill and makes sure she can pay them all; then writes each check in turn; and then puts each in an envelope and addresses it. So the stages of paying bills are collect and organize; plan what to pay and how, making sure not to overdraw the account; actually pay the bills; and put them in envelopes to send out. A home accounting program can build these steps in directly.
A scientist is interpreting the results of an experiment. He puts the raw numbers in one column, then in each successive column shows the result of one transformation. He needs to see not just the final result, but the process by which those results are achieved. An analysis tool that hid the calculations, and only revealed the result, would not be acceptable.

The actions people take in doing their work reveal their strategy, their intent, and what matters to them. A system that builds on these can improve the work they do. Understanding the real intent is key to improving work practice; you can redesign, modify, and remove steps as long as the user can still achieve their underlying intent. An intent is stable—for example, people have had the intent of communicating over a distance for ages. The steps, the way that intent has been achieved, have changed over time—from handwritten messages to the telegraph, the telephone, and videoconferencing. Supporting the current work steps just automates the way things are done currently (and because paper is almost always faster than computers, if the system does nothing but automate existing steps, it almost always loses). The goal is to change the work steps to make work more efficient. But the system must support all the intents concealed in the work, not just the primary espoused intents. If users have an intent of planning how to pay bills before they start writing checks, and the system doesn't support planning, the system will not be accepted.

All work, when it unfolds in time, becomes a sequence of actions—steps to achieve an intent. A sequence model (Figure 6.3) represents the steps by which work is done, the triggers that kick off a set of steps, and the intents that are being accomplished. They are your map to the work that your new system will change. Sequence models supply the low-level, step-by-step information on how work is actually done that designers need to make detailed design decisions. The sequence model is most similar to flow diagrams or task analysis (Carter 1991), but is unique in stating the intent and trigger for the sequence. A sequence model starts with the overall intent of the sequence and the trigger that initiates it. Then it lists each step in order, at whatever level of detail the interviewer collected. Any steps that cause problems are labeled with a lightning bolt. When modeling the work of an individual, the
**Intent:** Plug in

**Trigger:** Return to the office

Scan message list for important message—

*Use sender, subject*

**Intent:** Handle emergencies

*Choose urgent message*

*Read message about unhappy user*

*Decide more info needed*

*Make phone call*

*Leave phone message*

*File in phone folder*

*See list of messages*

*Choose message 9: subject indicates university news relevant to department*

*Read message*

*Delete message*

*See message 10 automatically*

*Read message 10*

_Figure 6.3_ Sequence model for handling mail. This sequence model shows how one user handled mail on one specific day. The intent is stated at the top left: "Plug in." This conveys the nature of handling mail for this user: much of his communication is through email, and when he left his office, he separated himself from this communication. Returning and checking mail was a reconnection, a "plugging in." This is implied by the trigger for starting this sequence, which indicates he does it whenever he returns to the office. The arrows indicate the sequence of steps. When he completed handling an emergency, he saved the message in a folder he uses as a phone book. This action indicates an unrelated intent, keeping a contact list up-to-date, which he handles opportunistically.
The sequence model does not attempt to show pattern or repetition; we identify those when we consolidate. Sequences may be studied at any level of detail, from the high-level work to accomplish an overall task to the detailed interaction steps with a particular user interface.

**Collecting sequences during an interview**

Collect sequences in an interview by watching people work or by getting a detailed retrospective account of their work. The hardest thing about seeing sequences is knowing what to pay attention to, and this changes depending on the project focus.

**Steps.** If you are studying the work across the department, or if you are learning about a new market, you will collect sequences at a fairly high level of detail. You want the actions people take, but not necessarily broken down into each movement. So writing a letter might look like: Get project information from project manager. Extract deliverables and delivery dates important to the customer. Write

**Sequence Model Distinctions**

The *intent* that the sequence is intended to achieve. Secondary intents will be embedded in this primary intent, and they are named as they are identified.

A *trigger* causing the sequence of actions. It is the notification to the user to take action. Triggers we have seen include the height of a stack of paper on a desk, the arrival of mail, receiving a request, and seeing a misplaced line of text in a document.

*Steps*, the action or thought preceding an action. In an actual sequence model, a step represents what actually happened. As we step back from the actual steps and look for purpose and strategy, the steps become more abstract. They move away from specific behaviors toward fundamental purpose.

*Order*, loops, and branches indicated by arrows connecting the steps. These reveal strategic and repetitive patterns of work. When the customer must make a decision about how to proceed, we show that as a branching path. The order gives us an access road map to ensure smooth transitions between tasks and allows us to see what steps could be combined or skipped without serious violation to the users' conception of what is going on in their work.

*Breakdowns* or problems in doing the steps shown with a red lightning bolt (black in this book).
introductory paragraph describing current project state. Enter dates. . . . This level of detail shows the overall structure of the work and how it fits together without giving huge amounts of detail about each task.

If you are designing a system or tool, study the tasks the tool supports in more detail. Look at what people do and also how they do it. So writing a letter might look like: Scroll window to find last letter written. Open it. Delete all content. Save under new name. Enter name of recipient. Pull Rolodex closer. . . . At this level of detail, we see the structure of the task and the actions that make it happen.

If you are designing the user interface, look at eye movement, hand movement, hesitations, everything. So writing a letter might look like: Use vertical scroll bar until icon for last letter written comes in view. Double-click on item to open. Read recipient name and scan first paragraph to make sure this is the right letter. Choose "Select All" from Edit menu. . . . This level of detail shows how the user interacts with the UI and reveals the issues for the UI to address.

In practice, the levels of detail blur somewhat, and it's safer to get more detail rather than less. Each action has a purpose in the user's mind. If it looks random to you, that's only because you don't know what the purpose might be. In a word processor, we repeatedly saw the user, with the cursor at the end of the line, hit the right arrow, see it move to the next line, then hit the left arrow to move it back. Even this was not random; he was checking to see if he was really at the end of the line or if there was extra white space because, in that word processor, the white space would make the line wrap.

HESITATIONS AND ERRORS. Notice when the customer hesitates or makes errors. These are your clues to his thoughts. Intervene and ask questions to find out what he is thinking about. Hesitations and errors indicate places where the customers' understanding of work is being contradicted by the tools they are using. This is an opportunity for your system to do better. If a task is largely a thinking task, hesitations reveal decision points in the process.

Customers' actions are never purposeless

Any glitch reveals a thought step

Stop the sequence at that moment and ask about issues.

TRIGGER AND SEQUENCE. Notice when there is a trigger to the sequence, that particular point in the process when the customer makes a particular move or input. This may be a particular physical action, such as pushing a button or moving a pointer, or it may be a mental action, such as getting the customer to think about the steps in the automation to be designed. This is an opportunity for the system to do better. Or, it may be that a particular product or tool is being used as a trigger to move to a new state and more data.

INTENT AND SEQUENCE. Notice when there is intent, when the customer is moving from one intent to another. There will be particular words or actions that the customer uses to signal a change. If there is a lot of overdriving or overthinking, you may want to ask when the customer is done. For example, the customer may say:

"I think I'm done."

So, you can ask if this is what they want or what they are thinking about. This is an opportunity for the system to do better. If a task is largely a thinking task, hesitations reveal decision points in the process.
Stop the customer and ask him to explain what he is trying to decide at that moment. Try to get him to think aloud, to reveal more of the issues.

**TRIGGERS.** Every sequence has a trigger—the event that initiated it. Triggers may be discrete events, such as the ringing of a telephone, the arrival of an invoice, or a person arriving at the door. Triggers may be based on time, like the first of the month or the first thing in the morning. Triggers may be less tangible, such as the pile in the in-box getting too large. Whatever the trigger, if the work is automated, it must have an analog in the new system. The system needs a way to tell the user there's something to be done. Otherwise, the user won't take action—for example, one mail product simply gets slower the larger the in-box gets. This doesn't act as a trigger for the user to clean it out; it just makes the product more and more frustrating to use.

**INTENTS.** The intent defines why the work represented by a sequence matters to the user at all. Every sequence has a primary intent, which applies to the whole sequence. Then there will be secondary intents, which drive the particular way the work is carried out. So our bill payer has a secondary intent of not overdrawing his account and of redefining what to pay and how much to pay so that important bills are paid and the account is not overdrawn. Intents are usually identified after the sequence is written, when there is time to look it over and think about what lies behind the customer's actions.

Sequences capture the most basic information about work practice. Not only do they tell you how work is _really_ done, they show how it is structured and the intents people care about. They present the detailed structure of work that designers will need when it comes time to structure the system. And they cut across the other models, tying them together. Because sequences are time-ordered, they show how different roles interact in different places, using artifacts to support communication and actions to get the work done.
**The artifact model**

People create, use, and modify things in the course of doing work. The things they use become *artifacts*, like archaeological findings. They each have their own story to tell about the work:

In one organization, a first-level supervisor prints the spreadsheet he uses to track projects weekly and gives it to his manager. His manager makes check marks against each project to indicate his approval and may make additional notes on the side. Then he signs at the bottom and gives it back. In this way the supervisor's personal tracking sheet becomes a sign-off mechanism and a way for the manager to communicate problems and issues. It suggests that sign-off and feedback are part of the job; an automated project-tracking system could build these features in.

Another woman builds a spreadsheet to calculate end-of-year results. The calculations take 15 minutes to do—then she spends the next 45 minutes making the spreadsheet look good so she can hand it out at the next management review. When a spreadsheet is given careful formatting, it's clear that the way information is presented is an important consideration and that spreadsheets are presentation tools as well as calculation tools. The original spreadsheet tools only displayed text; they were replaced with tools that could do fancy fonts and gave full control over the look.

Another organization has the goal of raising the level of cost consciousness among its people. They have a standard form for making a request for a purchase. The form has a place to describe the item and a place to justify why it's needed but no place to show the cost. When a purchasing form has no place to show cost, it suggests that cost is not a big concern in the organization. An automated purchase order request system could raise cost consciousness just by making cost prominent on the screen.

Artifacts are the tangible things people create or use to help them get their work done. When people use artifacts, they build their way of working right into them. Artifacts show what people think about when they work and how they think about it. An artifact reveals the assumptions, concepts, strategy, and structure that guide its use. It captures traces of actions, intentions, decisions, and interactions: forms, check marks, notes, signatures.

In another relationship, the history of artifacts is often a source of the meanings people assign to them. The practice of giving a card to someone's mother on a special occasion provides a setting where the card is a reminder of a family event. As the card is passed around the family, spatial locations, dates, and personal anecdotes provide a context for the card. The card is transformed into a personal keepsake, an artifact of the past. When the card is given to the next generation, it is transformed again as a symbol of the present.

Artifacts serve as guides to understanding a practice. Artifacts capture traces of the thoughts and actions of people. Highly stylized artifacts, such as documents, models, drawings, and annotations, make it easier to analyze what people are trying to do. Highly stylized artifacts make it easier to understand what people are doing, and what they are trying to do. Artifacts help us to see the world as it is seen by people who are working in it. Artifacts allow us to understand the intentions of people who are working in the world. Artifacts allow us to understand how people are working in the world. Artifacts allow us to understand why people are working in the world. Artifacts allow us to understand the structure of the world as people work in it.
guide the people who work with it. Artifacts might be to-do lists, forms, documents, spreadsheets, or physical objects under construction (circuit boards, cars, airplanes). Artifacts may be bought, designed intentionally, or created on the fly. They are manipulated in the sequence models and passed between people in the flow model.

In their structure—how they are arranged into parts and the relationship between the parts—artifacts show the conceptual distinctions of the work. When displays showing the status of a network are separated from displays of trouble alerts, this indicates that tracking ongoing status is different work from responding to alerts. When notes are written on a presentation handout, not where there is white space to write them on, but jammed in next to the text they refer to, this indicates that the close spatial relationship of text and note matters to the writer. When the list of things that a person would like to get is separated from the shopping list, this indicates that a clear distinction exists in the person's mind between the nice-to-have-someday items and the I-will-buy-this-today items. An automated shopper's planner had better provide a way to track long-term possible purchases separately from today's shopping list (Johnson et al. 1988).

An artifact model (Figure 6.4) is a drawing or photocopy of the artifact, complete with any handwritten notes. The model extends the information on the artifact to show structure, strategy, and intent. Highlight structure with lines and labels marking the different parts. Annotate the location of the parts showing how they are placed to give them prominence or support the artifact's usage. And write intents directly on the part of the artifact that supports the intent. Lightning bolts show where the artifact interferes with the work, whether because the defined structure does not match the work, because needed information is missing, or because it is too cumbersome to use.

**Collecting artifacts during an interview**

Artifact models always require interpretation to reveal their intent and usage. You can do this best with the customer during the interview. Look for and inquire into:
Figure 6.4 Artifact model. This physical model shows the structure of an artifact, in this case a personal calendar. The usage of this calendar reveals that it is not only about managing time; it is organizing an entire life. The rubber band makes the distinction between past and future. The calendar is acting as a storage place for reminders and to-do lists as well as a calendar. When the calendar gets too full, this is a convenient trigger for dealing with the to-do lists. The usage of the day view shows additional distinctions: meetings are listed from the top of the day down, but reminders of a more general nature are written from the bottom going up. Reminders are attached to a day; they are not kept in the provided “notes” area, so it is not used.

**Structure.** All artifacts have structure, even the most informal. People naturally create a structure to represent their thought, even when they start from a blank page. If they didn’t create the artifact on the fly, they may start from a given structure, either because it came as part of an artifact they bought or because they designed it themselves before starting the actual work. In this case, the structure inherent in the work wars with the given structure, and the artifact will show every place there is a mismatch. So the notes space on a daily calendar may be used for notes, but it may be left blank or used as a rolling to-do list. If everyone uses it like a to-do list, then organizing the day and scheduling are intimately intertwined.

Look to see how the artifact is structured. How does the presentation—layout, fonts, formatting, and white space—reveal structure? Assume every grouping of information corresponds to a conceptual distinction in the customer’s work. Can you and the user figure out what it is? Can you make these distinctions real in your system?
**Artifacts Model Distinctions**

*Information* presented by the object, such as the content of a form (e.g., a doctor's name, nurse's name, patient's name, and diagnosis).

*Parts* of the object, which are distinct in usage, such as page, kind of page (table of content vs. title page), headline, or figure in a diagram.

*Structure* of the parts explicitly in the object as given and implicitly in its usage: the division of a form into a section for the doctor's use and a section for the nurse's, the grouping of cells in a spreadsheet to represent part of the data for a single purpose, or the way some people use the top of a day within a calendar for meetings and the bottom for reminders.

*Annotations*, which indicate the informal usage of the object beyond that allowed by its explicit structure: Post-its stuck to a document, highlighting, and notes written on the side of a report.

*Presentation* of the object: color, shape, layout, font, white space, emphasis, and how they support usage.

*Additional conceptual distinctions* that are reflected in an artifact and that matter in its creation and use: past, current, and future in using a calendar; structure and content that repeats in a report from month to month; x-height and caps height in page layout.

*Usage* of the artifact—when created, how used, how people move through the parts of the artifact.

*Breakdowns* or problems in using the artifact, represented as a red lightning bolt (black in this book).

**Information Content.** The content of an artifact is the information, specific to the work, that the artifact carries. The content of an artifact tells the story of a part of the work—how the content was put in, how it was used, and who used it. The content fits into the structure of the artifact—or it doesn't, in which case customers modify the defined structure. Seeing how the content is manipulated reveals the artifact's usage—how it supports the work and also the detailed interaction with the artifact in the course of working. So each meeting on a personal calendar suggests the story of the work task that the meeting supports, but it also suggests the detailed story of how the user interacted with the calendar to put the meeting on it.

Look for the information the artifact carries and how it is used. Use the artifact to drive a retrospective account, as we discussed when describing interview principles in Chapter 3. Why is this artifact an...
appropriate carrier for this information? Who will see it and when? What would happen if the artifact didn't exist? Can you make the needed information available more simply in your system?

**Informal Annotations.** Informal notes and annotations are a gold mine of information. They tell you about the actual usage of the artifact. Did the defined structure get used? Was it extended? Was the artifact used to carry additional information by writing notes on it? Why was it used? What made the artifact the convenient carrier for the message? Can you put other channels in place to make this unnecessary? Can you see how the artifact didn't match the work, and can you see how to make your system fit the work better?

**Presentation.** Content and structure are revealed in the artifact's presentation. Look at formatting, the layout of parts on the page, and the use of white space. How does the artifact attract attention to some parts of the content and downplay others? The presentation supports the intent of the work if well designed and gets in the way if not. If the artifact is redesigned or put online, how should your system present it for easy interpretation in the same kind of way?

**Inquiring into an Artifact**

There are two levels of inquiry into artifacts. The first is to see how an artifact supports the customer's intent. The presentation, content, and structure are all clues to what matters in the work. So notes scribbled on a materials-tracking card telling the technician how to handle the material show that direct communication between user and handler is important. Any system that interrupted the communication (such as an automated tracking system) would cause problems in the work. To be successful, such a system would have to provide another way to accomplish the same intent. At this level of inquiry, we look at structure and usage to derive intent, to show why the artifact matters and what any automated system needs to account for. (See Muller et al. [1995] for an example of such an inquiry.)
If you think that the artifact might be supported or automated, then a detailed inquiry into the interaction with the artifact provides clues in how to structure the system. Things that cluster in the artifact are conceptual groups that should be kept together. The natural pattern of interaction with the artifact is a good guide to appropriate interaction with the system. So the notes on the materials-tracking card indicate that, if we want to automate materials tracking, we have to support informal communication between user and handler. This communication may happen at any time after the materials are received, so a single note that can only be entered when the materials are received won't do. Since the handwritten note is its own record, and having the record matters, the automated system needs to keep instructions related to the material available over time.

Artifacts are the concrete trail left by doing work. They capture multiple stories of how work happened, making it possible to walk through a retrospective account of those events. As a physical object, an artifact makes the way customers think about their work tangible, so you can see and inquire into it. But artifacts do not speak on their own; collect examples that have been used and interpret them with the customer during the interview to reveal their meaning.

The Cultural Model

Work takes place in a culture, which defines expectations, desires, policies, values, and the whole approach people take to their work:

A vendor creates a product that helps development teams control their development process. The product is well designed and well made, but fails in its target market of UNIX shops. UNIX shops pride themselves in getting code out without needing a formal process.

Another vendor makes an instrument so straightforward that unskilled operators can run it with ease. Their customer base won't buy it because they consider themselves highly skilled professionals who can run complicated systems.

Another company gives their scientists software that simplifies the reporting of experimental lab results. The scientists
reject the system because they consider proper reporting of
results to be part of the job of a scientist and don't want it
simplified.

In each of these cases, there was nothing wrong with the system
delivered. It was designed and built well and solved a real problem.
There was no technical roadblock to its use at all. In each case, what prevented the system's success was
the culture of its proposed users. If a system conflicts with its customers' self-image, or doesn't ac-
count for the constraints they are under, or under-
cuts the values important to them, it will not succeed.

The cultural context is to us like water to a fish—pervasive and
inescapable, yet invisible and intangible. Cultural context is the mind-
set that people operate within and that plays a part in everything they do. Issues of cultural context are
hard to see because they are not concrete and they are not technical. They are generally not represented
in an artifact, written on a wall, or observable in a single action. Instead they are revealed in the language people use to
talk about their own job or their relationships with other groups.
They are implied by recurring patterns of behavior, nonverbal communications, and attitudes. They are suggested by how people deco-
rate and the posters they put on their walls.

The cultural context includes the formal and informal policy of
an organization, the business climate created by competitors and by
the nature of the business, government require-
ments, the decor of the site, the self-image of the people doing the work, and the feelings and fears
created by the people or groups in the organization.
Culture influences work by altering the choices peo-
ple make. Because they don't want to have to deal with a certain group, or because they consider themselves professionals, or because they are worried about what their competitors are doing, people change the way they do their work. Design teams that understand these constraints can build their systems to account for them.
Cultural Model Distinctions

Influencers who affect or constrain work, shown as bubbles. These may be individuals or formal groups in the organization. They may be a collection of people who are not a formal group but are thought of together ("management"). They may be external influencers such as customers (and possibly multiple customer organizations), government regulatory bodies, standards groups, or competitors. They may represent the overall culture created by the organization or shared by the people doing the work.

The extent of the effect on the work shown by the amount the bubbles overlap. It suggests whether essentially everything about the work is affected by this influence or whether the influence is more partial. So the Food and Drug Administration influences the work of food and drug companies through its reporting and testing requirements, but this influence does not constrain everything about developing the food or drug product. On the other hand, everything an assembly line worker does is affected by the requirements of the assembly process.

Influence on the work. Arrows represent the direction of influence (who is primarily affecting whom) and how pervasive it is (whether this is an influence of one individual or...
group on another or whether it is more pervasive across an organization). We also represent pushback; in real situations it is rare that influence is all in one direction.

**Breakdowns** or problems interfering in the work, represented as a red lightning bolt (black in this book). Because all influences restrict work in some way, we only show breakdowns on the cultural model when they are especially harmful.

The following kinds of influence tend to be relevant to design:

**Standards and policy** that define and constrain how work is done or what can be used or bought, or the lack of such standards as a policy. So many companies define a standard PC configuration that they will support: "Use this configuration or you're on your own." Other companies live with standard procedures defined by themselves or imposed on them by the government or by customers: "Prove your process is compliant or we'll use another vendor."

**Power**, both formal in the organizational structure and informal through people's networks, expertise, and history. Power shows up in who has the right to decide who will do what in their work and the extent of autonomy a person can have. So one boss sets up his secretary's computer environment, limiting her ability to recover when anything breaks down: "I'll fix your machine in the way I think is important." In another organization, reimbursement for expenses is controlled by administration, which enforces the requirements for filling out paperwork and can choose to allow exceptions: "Jump through my hoops and I'll let you have your money."

The **values** of a company or team: what they stand for that produces a set of expectations about how people will interact and work. So one organization has the expectation that a project will be completed the same way as it was the last time, resulting in a feeling that innovation is unwelcome: "If it's a different plan, be prepared to justify it."

A group's own sense of **identity**, the way in which what they do is affected by how they think of themselves. So one UNIX shop held that they did not need to do formal up-front analysis and design because "we don't do process."

People's **emotions** about what they do, including fear about being laid off or getting in trouble for raising issues, or people's pride in what they do. So knowing that "email can be read by anyone, including management" led people in one organization to discontinue its use.

The idiosyncratic **style, values, and preferences** of an individual or team, creating a work environment that circumscribes others. So one boss will not use the computer, forcing his secretary to handle all his email communication: "Use the computer for me because I won't." Or a team can't work past 4:30 because everyone has outside activities that pull them away: "We are committed to home activities; schedule around them."
Recognizing the influence of culture

Culture is invisible, but can be deduced from things you see and hear.

**Tone.** When you walk in the door, what is the tone of the place? Industrial and sterile? Carefully designed and trendy? Formal and elegant? Messy and haphazard? When the customers design their workplace for elegance, they are unlikely to accept a system that looks haphazard. When they spend little time designing their workplace, just the bare minimum so that they can work, they are unlikely to accept a system that is overdesigned, which looks like time and money was wasted on elegance.

**Policies.** What are the policies people follow, and how are they recorded? Are there policy manuals, and are they used? Do people wanting guidance on doing their work routinely check them? Or is the operational policy—the policy that affects work on a day-to-day basis—really passed by word of mouth? If so, how much is based on real directives, and how much is folklore? Is policy generated by fear of a regulatory agency, of another organization, or of a manager? You can hear policy in the words people use: “We won't buy anything but UL-rated power supplies. They had a non-UL supply catch fire over in building 10 a while back.” If UL rating matters, you can highlight UL-rated equipment in the catalog you develop. “Better get these procedures documented properly. One of our competitors was cited for out-of-date documentation, and their stock dropped three points.” If written records are an important part of the work, you can implement systems that maintain them. The policies that people care about point to problems you can solve.

**Organizational influence.** Are there organizations, individuals, or job functions that keep showing up, either as troublesome or helpful? What are the organizations or job functions that always seem to get in the way? Who are the people who constantly show up as the ones who can solve the problem? Listen to how people talk about others: “Don’t call maintenance about this. They’ll take it
away to check it out and you won’t see it again for a week.” Can you change the design of your system so that maintenance doesn’t have to take the machine away to run diagnostics? “Oh, I can’t give this report to Mike looking like this. He runs this whole place—I’ll put it in my word processor and make it look really good.” If the reports that your product creates are given to management, you can make them high-quality presentations.

**Making culture tangible**

The cultural model (Figures 6.5 and 6.6) provides a tangible representation for these intangible forces. In a cultural model we represent influencers (people, organizations, and groups) in the customer’s culture, showing how they influence each other. Influencers are shown as large bubbles. Because culture is felt as a weight or pressure influencing actions, the bubbles sit on one another, showing how one organization forces another to take or not take actions. We represent influences as arrows piercing the bubbles and label the arrows to represent the type of influence. Influences are labeled with language representing the experience of the people doing the work, so the influence from an internal help organization might read, “We are unreliable and will wipe your hard drive on a whim.” No one in that help organization would ever actually say those words, of course, but the people who use their services operate as though they were saying exactly that. Using direct language on the model makes the culture it represents stand out. Where an influence stands out as being particularly harmful and counterproductive, we mark it with a lightning bolt, our universal symbol for problems or breakdowns.

Cultural models do not map to organization charts. They show how power is experienced by people, rather than the formal power of the organization. So it’s unusual to see the whole management chain represented on a cultural model. Individual managers will appear when they are part of the work, as when a manager makes his secretary interact with the computer for him. There’s often a bubble to represent the organization’s culture, with
You aren't our primary user; we'll fix bugs for you in our own time

Our technology is standard; use it even if it doesn't work

Oh, I'm just a secretary. He processes our bug reports that way. You aren't our primary user, you know.

Can you hear my manager talking? "Oh, I can't talk right now." This. He doesn't like us. He processes our bug reports that way, you know.

The cultural model

The culture of a product development organization. This is a typical cultural model in a product development organization. In the center we see the interviewee, U9. Since cultural models are initially built as the result of an interview with one person, they represent the point of view of that one person. U9 is in the development organization, and the model shows two major constraints on them. The marketing organization constrains them through ill-specified product requirements. Competitors constrain them by creating a climate in which keeping up with the number of features is the primary goal. The basic appearance of this model—the interviewee surrounded by influencers—is very typical.

Influences like “We are totally customer-focused” or “Spending money is not a problem.” In adversarial situations, “management” may appear to represent how “they” do things to “us”—“We think you salesmen are children who need to be watched every moment” might be an example. Individual managers appear as managers only when they are charismatic figures who define the organization’s culture. In this case,
Figure 6.6 The culture of a customer-centered organization. This cultural model is typical when there is a definite corporate culture to account for. This cultural model represents a department store that has made customer satisfaction its first priority. Unlike many places that espouse that goal, this company has really done it—so much so that people throughout the organization are conditioned to think who their customer is and how to give them the best service. Paired with this focus on the customer is an equally pervasive understanding of the business—so much so that "We sell socks" is a watchword within the company. We show the pervasive company culture as an umbrella over everything, with individual influences going back and forth between the bubbles. The lightning bolt indicates a place where two values conflict: being customer-centered leads the store to avoid setting standards for computer configurations, but such standards would make the PC support analyst's life easier.
their power is experienced as direct and personal. McDonald's franchise owners used to tell about how Ray Kroc, then CEO of McDonald's and a fanatic about cleanliness, canceled a franchise because he found one fly in its kitchen (Boas and Chain 1977). Everyone lived in fear that he would show up in their kitchen next. Ray Kroc would appear on a cultural model.

**The physical model**

Work happens in a physical environment that either supports and enables the work or gets in the way:

One company creates a page design product in which the look on-screen doesn't quite match that of paper. They think it is close enough because they expect their users will print draft versions and use the paper output for the final draft. They don't know that most of their users don't have printers by their desks, or even close by. So users spend time running back and forth to the printer and copying good drawing elements from one document to the next.

Another company gives their sales force portable computers to do presentations. They don't know that salespeople are only given a few minutes at a site. The salespeople don't have time to bring up a computer, and they depend on leaving materials behind with their customers. The portable computer doesn't fit either need.

A utility company gives their electricians documentation in a three-ring binder. Only later do they discover the electricians are trying to balance this awkward binder on a cherry picker in all types of weather. They redesign the documentation as a small, spiral-bound flip book with laminated pages and a clip so it can be hung from a belt.

Any product or system must live with the constraints of the physical environment as it exists. If it ignores those constraints, it creates problems for its users. In each of the above examples, a system created problems for its users because it assumed things about the workplace that were not true. Studying the users' workplace ensures that the system accounts for the physical environment.

The physical model reveals design constraints
The physical environment constrains what people can do, but within those constraints people do have some control over their environment. Studying the workplace offers important clues to the way people structure and think about work. People restructure their workplace to support doing work in the way they prefer, to the extent they can. Because they structure their environment to be convenient, the structures they create mirror their thought. The structures show what people group together into conceptual units and coherent tasks. An office worker sets up places in her office to keep her work organized. The chair receives urgent messages from coworkers; the space next to the computer is kept clear so that when she starts a task, she has a place to lay it out; the in-box is the "guilt pile"—things she feels that she ought to deal with, when she has time. The places she creates mirror the way she thinks about her work: urgent, current, guilt pile. They make work distinctions concrete. A system that makes these distinctions real will fit with the work easily. The workplace shows us issues in doing work; from the elaborate system of piles that people create, we can deduce that tracking multiple little tasks is a problem, and people might benefit from better ways to track them.

The physical environment is the world people live in: the rooms, cars, buildings, and highways they move about and work in; how each of these spaces is laid out so that it supports work; and how they use these spaces in the process of working. It includes how they move about, how the space supports or hinders communication, and the location of the tools people use (hardware, software, networks, machines) to do work. The physical environment affects how work is done at every scale: the multiple sites and their relationships to each other, the structure of a single site, and an individual's workplace. The work site may be structured as an open "bull pen" with supervisors' offices around the outside. It may consist of many individual cubicles dividing up a large room. A person's workplace may be an entire building or buildings, if they are maintaining equipment. It may be a car or airplane if they work on the road. Within a work site there are places to do work, which may be offices, labs, workbenches, or workstations. Workstations may be dedicated to one person or shared.
Physical Model Distinctions

The physical model shows the places in which work occurs: rooms, workstations, offices, and coffee stations. It models whether the space is small or large, a primary or secondary workplace, private or open, clustered, or empty space available for changing work activities.

The physical structures that limit and define the space: sites, walls, basements, desks, file cabinets, and other large objects.

The usage and movement within the space—how people move about in it and move things about in it in the course of accomplishing their work.

The hardware, software, communication lines, and other tools (calculator, Rolodex, in-basket, measuring tools, Post-its, printer, fax) that are present in the space and support the work or seem related. We show network connections, not to model the network itself, but to emphasize who is connected to whom and therefore what communication among people we can automate.

The artifacts that people create, modify, and pass around in support of the work—folders, spreadsheets, to-do lists, bills, ID cards, approvals, piles of stuff. The physical model shows the artifact and its location, not the detailed structure and usage of the artifact.

The layout of the tools, artifacts, movable furniture, and walls in relationship to each other to support specific work strategies.

Breakdowns or problems showing how the physical environment interferes in the work, represented as a red lightning bolt (black in this book).

Seeing the Impact of the Physical Environment

The physical environment is easy to see—it's all right there. It's harder to tell what matters. What will affect the design problem, and what will not? Here are some things to look for.

Organization of Space. Are there stations, and how do they relate to the work? Are stations grouped to follow the flow of work to make work efficient, or are similar stations placed together to make management efficient? Are the people who made the decision conscious of the trade-off? This will indicate what they care most about and therefore what the most important problems for you to solve are.

Division of Space. Where are the walls, and how do they break up the work? Do they follow the structure of the work, or do...
they interfere with it? If they interfere, how do people overcome them? Do they run back and forth a lot? Do they shout? (During one interview, the user directed a question at the wall, and the wall answered. It was so thin he could carry on a conversation with his colleague on the other side.) Every communication breakdown creates an opportunity for you to ameliorate it: Who needs to communicate? How and when? Can you obviate the need by providing information where it's required, or can you make it easier?

**GROUPING OF PEOPLE.** How are people grouped into the spaces? By function or by project? Does each person have their own separate office area, or do they mix and share spaces? Often specialists sit with other specialists, not with the project they are assigned to. Creating a sense of belonging to the project team becomes difficult. Conversely, developers who are seated with their internal clients tend to identify with them. They tend to adopt their perspective against that of the development organization. What can you do to make the whole interrelated set of information systems apparent to all developers, so they are continually reminded of the effect their short-term fixes will have on the whole?

**ORGANIZATION OF WORKPLACES.** How are the individual stations, offices, or work areas organized? How do they support the work? What is kept out (immediately visible), ready to hand (accessible without moving), and available (in a drawer or across the office)? What does this say about what's most important to the work? Things kept together tend to be used together. What does this say about the structure of a task? Can you see what makes up a whole task in what is kept together for easy access? Can you design your system so that the most important function is available where needed and so that whole tasks are coherent in the system?

**MOVEMENT.** When do people move? Why do they leave one place and go to another? What triggers them to do so? Is this intrinsic to the work, as when a maintenance person goes to look at a machine? Does it provide a way of access to a physical place, or does it provide a way of access to information? Do they have a real need to visit people? Do they need to see the office where some information will help them? What does the office support?
Work Models

Does it provide an opportunity for informal discussion and problem solving?
Do the customers see it as a problem, or are they like system support people, who generally enjoy getting out of their offices? Understanding why the movement happens will help you decide whether it makes more sense to support it better or eliminate it.

**Showing what matters in the physical environment**

A physical model (Figures 6.7 and 6.8) is a drawing of those aspects of the workplace that are related to the project focus. The physical model shows how the physical environment affects the work. It is annotated to show how the space is used and to show strategies, intents, and cultural values that are revealed by the way space is used. A good physical model evokes the experience of the workplace in the same way as a caricature. Aspects of the environment are only represented if they matter to the work; for example, “basement” might mean “far away, uncomfortable, and inconvenient to get to.” If the worker must nonetheless go there or worry about what happens there, we represent it in the model. Wherever the physical environment interferes with the customer’s work—things are too far away, or too cramped, or the right tools aren’t where they are needed—we show it with a lightning bolt.

The physical model shows how people respond to the environment by restructuring it. Do people accept the workplace as it is, or do they work around it? If the environment consists of doorless cubicles, do they put things in front of the door to gain a measure of privacy? How else does the work as it is experienced mismatch work as the environment wants it to be? What do people do about it?

A physical model is not a floor plan for the work site. Nor is it an inventory of the computer room. Either a floor plan or an inventory can be collected easily, without resorting to contextual techniques. A physical model does not show extraneous detail unrelated to the project focus—potted plants, kids’ toys, and family pictures are usually not relevant and can be omitted when you’re designing a system. If you were designing the work environment itself, you might have to take them into account.
Chapter 6  Work Models

The five faces of work

Each of the above work models presents a different perspective on the work. These perspectives interlock: a person plays roles; a role has responsibilities, undertakes tasks, and exchanges artifacts with other people to discharge these responsibilities. The sequence models show how these tasks are accomplished in detail and how artifacts are used in accomplishing them. The responsibilities and manner of accomplishing them are represented by the data elements in the diagrams. The data elements are connected by actions, which suggest the sequence in which the actions are performed. The process of working is thus represented by the sequence of actions and the sequence of data elements. This representation is a useful tool for understanding the work process and for identifying opportunities for improvement.

Figure 6.7  Physical model for a university environment. This university has set up its workstations so that anybody can use any workstation. The small boxes represent the workstations—over 1000 distributed all around the campus. To indicate their independent nature, we show them as standing alone and show the users separated from them, to indicate any user can access any workstation. "U1" means "user 1" and indicates the office of the user we talked to in a central building, with the central VAX machines in its basement. All user files are stored on the VAX. The "central PO" is a piece of software that routes mail between users. We have shown the routing of one message because we were designing a communications product. This work model shows the value of choosing a representation that is expressive of the data—in this case, that there are many workstations spread out over the campus according to no particular plan.
The five faces of work

Figure 6.8 Physical model for an office. This physical model shows the workplace of one user. The model represents a cubicle and shows how she has structured her environment to help her get work done. The placement of her IBM Selectric in the doorway, the in-box next to the door, and the shelf used as a drop-off place all suggest a strategy to minimize interruptions caused by working in an open cubicle. The phone, Rolodex, and calendar are all grouped together, suggesting that these tools work together to support communicating and coordinating with others. And the open space around her workstation suggests an intent to keep this area clear so she can lay out her next task. The team has annotated the model to reveal these distinctions and to show breakdowns, such as the printer being too far.

them are driven by organizational context and culture as shown on the cultural model. The work represented by the sequences is done within the work environment described by the physical model. Stepping back and looking at the models together reveals all the different
aspects of work and how they relate to each other. It reveals how the whole work of one person hangs together.

Seeing how customers work drives design. A design team needs to know what they should make—what work might be supported, what the big problems are, what the customers care about. They need to know what they must account for in their design: the roles and how they interact, physical and cultural influences and constraints. They need to know how to structure their design: the strategies people use to get their work done, the way they break up and think about their work conceptually. By organizing and presenting customer work clearly, work models make it possible to answer these design questions. They provide an integrated view of the customer’s work practice and also show the details of work structure that guide the fine points of design.

The individual work models as described above represent the work of each customer that a team interviewed independently. In Part 3, we’ll see how to consolidate models so that instead of showing each customer independently, they show the common structure and pattern of work across all the customers a system needs to support: a whole market, a department, or multiple departments. With consolidated models the design team has a single statement of the work they need to address, rather than trying to support each individual separately. We do this by first observing, inquiring into, and representing the work of specific individuals. Then we consolidate the models of each type. We bring all individual flow models together into one consolidated flow model to reveal the common roles and their interaction. We consolidate all the cultural models, all the physical models of whole sites, and all the physical models of individual workplaces. We consolidate all the sequences representing similar tasks and all the artifacts achieving the same intent.

These consolidated models make the underlying patterns of work across customers explicit. At the same time, they capture the variation between customers by showing any unique structure or details put into practice by each customer site. The design team can then decide what aspects of work they want to support. They can take a good idea for approaching the work implemented by one customer site and build it into the system to make it available to all. They can streamline
the work, removing extra steps and taking advantage of technological possibilities. From this redesigned work practice, they can design a system that supports the new work practice and drives the design of the user interface and system implementation. The rest of this book discusses these steps.