3

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).

Design processes work when they build on natural human behavior

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

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.

When you're watching the work happen, learning is easy

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 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.

Seeing the work reveals what matters

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.[1]

> 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 apprentice relationship provides the opportunity to do so.

Seeing the work reveals details

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.

Seeing the work reveals structure

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 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]

Every current activi recalls past instances

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.

[2] 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.

Contextual Inquiry is apprenticeship compressed in time

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.

Contextual Inquiry tailors apprenticeship to the needs of design teams

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 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?

Go where the work is to get the best data

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.

Avoid summary data by watching the work unfold

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.

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 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?" 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.

Avoid abstractions by returning to real arti and events

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.

Span time by replayi past events in detail

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.

Keep the customer concrete by exploring ongoing work

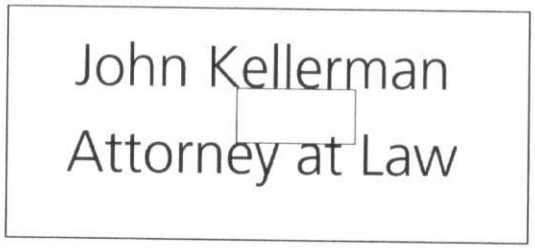
PARTNERSHIP

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.

Help customers articula their work experience

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.

Alternate between watching and probing

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.)

Teach the customer ho to see work by probing work structure

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.

Find the work issues behind design ideas

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.

Let the customer shape your understanding of the work

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.

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.

You aren't there to ge list of questions ansu

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

You aren't there to an questions either

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.

It's a goal to be nosy

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.

Partnership creates a sense of a shared quest

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*—

Determine what customer words and actions mean together

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

Design ideas are the e product of a chain of reasoning

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.

Design is built upon interpretation of facts—so the interpretation had better be right

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:

Sharing interpretations with customers won't bias the data

> "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.

Sharing interpretation teaches customers to see structure in the work

> 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

Customers fine-tune interpretations

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.

Nonverbal cues confirm interpretations

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

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.

Clear focus steers the conversation

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.

Focus reveals detail

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.

Focus conceals the unexpected

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

understanding does not 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:

Internal feelings guide how to interview

Surprises and contradictions: The customer says something, or you see them do 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.

Commit to challenging your assumptions, not validating them

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 promise confidentiality, get permission to tape, and start the tape recorder. Explain that the customer and her work is primary and that you depend on the customer to teach you the work and correct your misunderstandings. You ask for any opinions about the tools the customer uses (if relevant) and get an overview of the job and the work to be done that day. This is summary data, not contextual data, so don't pursue any issues; instead, watch to see if they come up in the body of the interview and pursue them then, when they are in context. Unless the work domain is unfamiliar, this part should last no more than 15 minutes.

Get to know customers and their issues

The transition: The interviewer states the new rules for the 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 off 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.

Explain the new rules of a contextual 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

Observe and probe ongoing work

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.

Feed back a comprehensive interpretation

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.

4 Contextual Inquiry in Practice

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