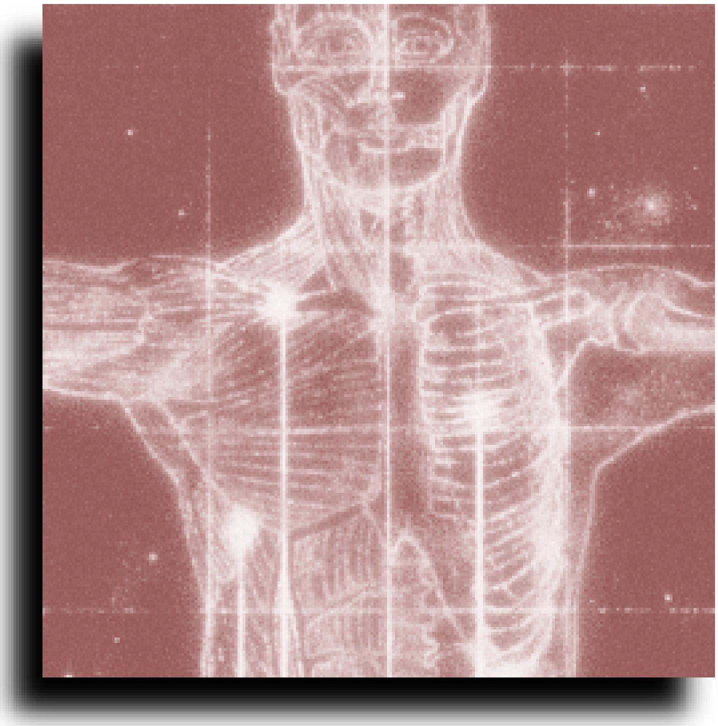
Apprenticing *With the Customer*



Hugh R. Beyer
Karen Holtzblatt

A new computer system changes how its customers work. Designing such a system requires intimate knowledge of customers' work and motives to ensure that the system supports them well. The creation of a new system implicitly means designing the new work practice it will support.

Because system design is so intimately involved with how people work, designers need better approaches to gathering customer requirements.[1] The current interest in participatory design, ethnographic techniques, and field research techniques grows out of the recognition that traditional interviewing and surveying techniques are not adequate for designing today's applications. These new approaches seek to improve requirements definition by creating new relationships between designers and customers [11].

It is the relationship between designers and customers that determines how well the design team understands the customer problem. This includes not only the overall relationship between design team and customer community, but the individual, minute-by-minute process by which a single designer works with a single customer to understand the customer's work. It is by understanding each person in the context of that person's work that the team comes to understand the whole work problem [5, 13]. Through face-to-face interaction, customers and designers define

[1] Throughout this article we use *customer* in the Total Quality Management (TQM) sense of those who depend on a system, directly or indirectly. We reserve *user* for those who actually interact with the system. *Designer* here means anyone who defines what a system will do, whether under the job title systems analyst, user-interface designer, or engineer.

together what the proposed system must address and what patterns of work it must account for. What is the character of the interaction that allows this collaboration to occur? How do we help members of engineering teams create such relationships with their customers?

For nine years, we have brought design teams to the field to gather customer data for their own design problems. The design teams have been made up of engineers, marketing professionals, writers, managers, and the other people commonly found on engineering projects. The teams have used Contextual Inquiry, our field requirements gathering technique, to get the detailed perspective they need on the customer.

We have not found it necessary to have experts in field research on the team (in contrast to others [8]). We have been able to evoke the appropriate behavior in team members by giving them a model of relationship they can use with no more than a day of training. A familiar model of relationship suggests roles, attitudes, and behaviors people use naturally. It allows designers to get reasonably good data on their first interviews and to continue to improve their skills with practice.

In this article we describe the relationship we create through Contextual Inquiry. We show how to tailor the relationship so it fits the needs of design and describe how the interaction between a designer and a customer works in practice.

Finally, we suggest ways a closer relationship between designer and customer supports a larger collaborative relationship between the design team and the community it supports. (Here we focus on creating the right relationship between designer and customer. A specialist in field research would also provide knowledge of work and social structure. We discuss how to make this perspective available to design teams elsewhere [4].)

The Designer as Apprentice

The fundamental problem in the relationship between customers and designers is that of enabling learning: How do designers learn enough about customers' work to design well? What kind of relationship allows customers to impart deep knowledge about their work?

Looking outside the computer field, the relationship between master and apprentice stands out as a useful model. Just as an apprentice learns a skill from a master, designers want to learn about their customers' work from the customer. This traditional relationship underlies the model of relationship we depend on in Contextual Inquiry. The critical aspects of the relationship are as follows:

Teaching ability is not needed. Craftspeople, like customers, are not natural teachers, and teaching is not their primary job. But they do not need to be—the master craftsperson 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 or her skill in the abstract. A master teaches by doing the work and talking about it while working. This makes imparting knowledge simpler, for several reasons.

Teaching in the context of doing the work obviates formal presentations and course materials. It eliminates any need for craftspeople to think in advance about the structure of the work they do. As they work, 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 an action. Observation interspersed with discussion requires little extra effort on the part of either master or apprentice.

Similarly, customers do not generally have the skills to talk about their work effectively; this is not their job. But customers 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: "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."

Seeing the work reveals what matters. Even if the master is a good teacher, apprenticeship in the context of ongoing work is the most effective way to learn. People are not 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 reasons; other actions are habit and no longer have a good justification. Nobody can talk better about what they do and why they do it than they can while in the middle of doing it.

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 [10]. In one case, 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 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 craftsperson to reveal all the details of a craft. While working, the master can describe exactly what he or she is doing and why. As master or apprentice observes a pattern or principle in action, he or she can point it out immediately.

Talking about work while doing it protects the master from the human propensity to talk in generalizations. Customers do this, too. Rather than remember all the details of a task or event, they mix up many similar events and then talk about them in the abstract as though they were all one. This abstraction is divorced from any of the reasons for choosing one action over another and may not match any real instance at all. A design built on such generalizations may not meet anyone's needs. Even when generalizations include detail, it is made-up detail—nothing is more detailed than an organization's software development policy manual, and there are few less reliable guides to how that organization actually develops software.

The apprenticeship model provides an attitude of inquiry and learning to adopt while in the field. **It recognizes that the customers are the experts in their work and the designers are not.**

Every action customers take and every object around them helps them talk about what they are doing in detail. One customer said he would not use the book index to find the solution to a problem: "It's never in the index." He was unable to say what he had looked up or how he had come to this conclusion. All his bad experiences with indices were rolled up into a simple abstraction: it's not there. But when we watched him looking things up, we could see the kinds of terms he was attempting to use and why they did not match the terms in the index. The environment itself reminds customers what to do: In another case, 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.

Seeing the work reveals structure. The apprentice learns the strategies and techniques of work by observing multiple instances of a task and forming his or her own understanding of how to do it. This understanding incorporates the variations needed to do the task well. The master can communicate techniques and strategies without articulating them.

In the same way, designers 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, 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.

The apprentice can learn from the master's experience. Every event can serve as starting point for discussing similar events in the past. In this way apprentices learn from the experience gained by a master before their apprenticeships 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. (Orr describes similar activity among modern-day technicians for similar reasons [9].)

Designers typically have less time to spend with their customers than the years needed for an apprenticeship. But in the same way an apprentice can learn from the master's experience, designers can learn about events that occurred in the past. Events that occur while the designer 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.

In one case, 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 which 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.

Ethnographic and field research techniques [1, 2, 7, 12] seek to provide rich detail about customers by taking designers into the field. The apprenticeship

The job of the designer is to recognize work structure. *The detail the apprenticeship model makes available and the dialog it encourages make it easy to see work structure and reveal opportunities to improve work with technology.*

model provides an attitude of inquiry and learning to adopt while in the field. It recognizes that the customers are the experts in their work and the designers are not. A designer taking on the role of apprentice automatically adopts the humility, inquisitiveness, and attention to detail needed to collect good data.

The apprentice role discourages designers from asking questions in the abstract and focuses them on ongoing work. It also allows customers to shape designers 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.

Building on Apprenticeship

While apprenticeship defines an attitude for designers to adopt, their motive in observing work is not that of an apprentice. Where apprentices want to know how to do the work, designers must determine what a system might do to support the work. This leads to changes in the customer-designer relationship.

The designer must be responsible for seeing work structure. A master teaches his or her apprentice how to do the work the master is expert at. But a designer must understand structure and implication: the strategy to get work done, constraints that get in the way, the structure of the physical environment as it supports work, the way customers divide work into roles, and the recurring patterns in their work—and what all this implies for any potential system. The customer is not an expert in seeing work structure, and does not naturally articulate it [5].

For example, we once asked a secretary how she started her day. Her answer was, "I don't know, I just come in and get started." It was 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."

Her 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 does not even occur to most people as a topic of conversation.

The job of the designer is to recognize work structure. The detail the apprenticeship model makes available and the dialog it encourages make it easy to see work structure and reveal opportunities to improve work with technology.

Designers must articulate their understanding. Apprentices can learn the structure of work without articulating it, by simply copying what they see. But if designers do not articulate the structure they see to their customers, they cannot get their understanding corrected. The system they build based on their inaccurate understanding will not be useful. Apprentices prove they understand by doing the work and being corrected. Designers cannot afford to find out they do not understand by building the wrong system. Even iterating prototypes will be a long and difficult process if the basic understanding of the work is wrong. Better to find out immediately, by sharing it with the customer.

Any system is based on a chain of reasoning starting from a work observation and leading to an aspect of the design [3]. Designers will build different systems depending on how they understand their customers' work. In working with one user of an accounting package, we learned that a sheet of accounts and account numbers were kept next to the user's computer monitor. Here are some interpretations of what this observation might mean, and what it might imply to for design:

- 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 on-line.

- 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 accounts only by name.
- Perhaps compatibility with paper systems is necessary, but referring to accounts by name is far 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 only way to ensure an interpretation is correct is by sharing it with the customer. We fail in the entire purpose of working with customers if we do not share and validate these interpretations. An apprentice does not have to do this, but the apprenticeship model provides a natural context for sharing observations of structure and interpretations of their meaning.

The designer's job is to improve work. An apprentice is assumed to have no skills or knowledge relevant to the work at hand. The designer constantly thinks about ways to improve the work with technology. Designers know about technology, have skill in applying it to real-life problems, and naturally respond to the customer's activities by designing improvements to them.

These are not digressions, because the designer's purpose is to improve the customer's work. The apprenticeship model allows both designer and customer to introduce design ideas as the work suggests them. The customer can respond to the idea while doing the work the idea supports. There is no better time to get feedback on an idea. The designer expands his or her understanding of the work in this way—if the idea does not work out, there is some aspect of the work the designer is not accounting for. And the customer enters into the design conversation— learning what technology can do and how it might be applied.

This gives the customer the power to shape the initial perspectives that will eventually result in a full design. Any iterative technique (such as rapid prototyping) will enable customers to shape a design. But any iteration of a design can only make small modifications to the initial structure. Involving the customers in the initial discussion of work structure can lead to radical alterations of system purpose and structure.

The designer has a specific focus. The apprentice learns whatever the master knows—the master determines the content. But the designer, who intends to support a specific kind of work, needs to guide the customer in talking about this specific part of that work.

One team we worked with was designing an information system supporting scientists. They needed to understand scientists' behavior in pursuing a scientific inquiry but did not need to know all the scientific knowledge that guides an inquiry. Had their goal been to replace the scientist, perhaps by creating an expert system supporting the scientists' decision-making, their focus would have been different. They might then have needed to understand how to encapsulate scientific knowledge, but they still would not have needed the theory behind it. The designers' focus determines the scope of the information they need.

The apprenticeship model allows designer and customer to guide the conversation into areas of work that are relevant to the design. The customer, by revealing all the aspects of the work, broadens the view of designer beyond the initial assumptions. The designer, by focusing on particular aspects, draws the customer's attention to the parts of the work that can be affected in the design.

Using the apprenticeship model as a base, designer and customer can develop an interaction that allows them to explore and understand the customer's work together. The customer is the expert in the work and how to do it; the designer is the expert in seeing the structure of work and the technology available to support it. Although the designer is not an apprentice, the model is an effective form for interacting with customers.

Apprenticeship In Practice

In practice, the customer might work for a while, with the designer looking on. The customer is immersed in the work, thinking about content (as usual). The designer, as apprentice, looks for patterns, for structure, and for things he or she does not understand. At some point the designer makes an observation or asks a question. This interrupts the flow of work. In order to respond, the customer has to stop working, step back, and think about the question.

A simple question gets a factual answer. But a question about the structure of work starts the customer thinking about structure: "So the first thing you do is check for any urgent communication, no matter what form it may have come in?" This question suggests a way of thinking about the work. It makes a previously implicit strategy explicit, and invites a conversation about that strategy.

The customer now responds at two levels: First, he or she addresses the question and a conversation about the work ensues. When the conversation winds down, the customer returns to doing work and the designer returns to watching. Second, the question is an example of seeing strategy where before the customer saw only actions. Customers soon learn to see strategy, and start interrupting themselves to make observations about the work they do. When they—or designers—have another observation about the work, they interrupt it again. This cycle underlies the entire interaction.

Maintaining the apprentice relationship requires attention to the interaction. Otherwise, customer and designer may fall back into more traditional patterns for requirements gathering. Fortunately, the underlying master/apprentice relationship is a natural one and is not hard to maintain as long as the participants pay attention to common pitfalls. Here are some of the pitfalls to watch out for:

Falling into other relationship models. Having apprenticeship as a model gives designers a way of behaving that replaces the behaviors they would naturally use otherwise. But because other relationship models exist, and are habitual, designers must recognize when they have fallen into them and know how to get out.

Interviewer: Designer and customer start to act as though there were a questionnaire to be filled out. The designer asks questions that are not directly related to ongoing work, because ongoing work has ceased. The customer answers the question, then falls silent, waiting for the next. The best solution for this is to go back to ongoing work, which effectively prevents this question/answer interaction.

Expert: The designer becomes the customer's assistant in understanding how to use a tool. This is particularly troublesome when the designer helped design or build the tool the customer uses. Start the interaction by explaining that you are not there to answer questions. Then, if you fall into the role during the course of the interview, step out of the 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 you may have."

Personal friend: Working together on understanding the customer's work results in a close, personal interaction. This intimacy comes from the sense of mutual discovery and from the attention the customer receives from the designer. However, this intimacy can lead to conversations that grow naturally out of such a relationship, but are irrelevant to the design focus. You need to keep your focus on the design issues and refrain from making or following up on comments that are not relevant.

Talking in the abstract. Even though designer and customer are in the customer's workplace, with the customer's work all around, it is easy for the customer to start talking in the abstract. The designer needs signals indicating that the customer is talking abstractly, and must then return them to actual experience.

If the customer is leaning back and looking at the ceiling, he or she is almost always talking in the abstract. This is the position assumed by people who are keeping the reality all around him from disrupting the conception they are building in their minds. People talking about real experience lean forward and usually point at some representation of what they are talking about. Words indicating the customer is generalizing are another signal. A customer who says "generally," "we usually," "in our company," 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. If the customer says, "We usually get reports by email," ask, "When was the last time that happened?" Never try to get more detail by asking what will happen next; this requires the customer to respond with a guess or an abstraction. Instead ask about real events in the past.

Retelling a past event is hard because so much of the context has been lost. People are prone to giving a high-level summary of past events devoid of the detail we need. Most people will start telling a story in the middle, ignoring what went before. Then they will skip whole steps as they tell the story. Listen for missing steps and ask the customer repeat comments as necessary.

For example, when one customer started talking about dealing with a report, the conversation went like this:

> *"When I got this problem report I gave it to Word Processing to enter online—"*
>
> *"So you just handed it on automatically as soon as you got it?"*
>
> *"No, it was high priority so I read it and decided to send a copy to the Claims department."*
>
> *"How did you know it was high priority?"*
>
> *"It had a green sticker on it."*
>
> *"Who put on the green sticker?"*
>
> *"That's put on by the reporting agency. They make the decision about whether it's high priority and mark the report."*
>
> *"Did you just send it on to Claims, or did you write them a note about why they needed to see it?"*

At each step, the designer listens for steps that probably happened but that the customer skipped and has the customer go back to find out about them. In this process, the customer mentally goes through the steps and recalls more about the actual work than if allowed to simply tell the story in order.

Tuning an interpretation. Designers worry about biasing the data and tend to be reluctant to share their interpretations. This is a valid concern, but

The close work with individual customers leads the design team to **discover the detail and structure of a work domain they need for their design.**

we have found that customers are not easily led astray in the midst of their work. Customers who are not used to seeing work practice can say what is going on more accurately by responding to an interpretation than by stating it themselves. Open-ended questions give the customer less guidance in thinking about their work than an interpretation, and this results in less insight.

In our example of the customer starting her workday, the designer might have asked, "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, no matter what form it may have come in?" she can compare this statement of strategy with 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.

Customers responding to the interpretation in the moment of doing the work can fine-tune it quite precisely. Customers commonly make slight changes in emphasis to make the interpretation exact. They can do this because they are given a starting point they can compare with their immediate experience and correct it, rather than having to start from scratch.

Customers may also say "no," but, to be polite, not say "no" directly. Here are some indirect ways customers say "no":

- "Huh?"—This means that your interpretation was so far off that it had no apparent connection to what the customer thought was going on.
- "Umm...maybe."—This means "no." Customers nearly always respond immediately to an interpretation that comes close. A pause for thought means 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.

Designers in the field quickly discover how difficult it is to lead a customer in one direction when they are in the middle of an experience that takes them in another. A design is based on interpretations of work; if these are not shared with the customer, the design will reflect the designer's made-up explanations of customers' behavior. By sharing our interpretations and being honest about interpersonal cues, we take away a valid understanding.

Adjusting Focus. The designer has an idea of the scope of the system he or she might create and the kind of work requiring support. This focus guides the interaction with the customer, determining what to pay attention to and what to ignore. However, the designer's initial focus may be wrong or too limited. The designer may be tempted to dismiss what the customer is saying. Information that is too far out of the designer's expectations just seems wrong. It is easier to say this customer is wrong or strange than to try to incorporate the new information. Instead, this is an opportunity to challenge existing assumptions. Probe into the details. Why is this information so different? What is going on? What expectations are violated? Probing leads to an expanded understanding of the work.

Any requirements-gathering process runs the risk of being too focused. The result is a well-designed system that does the wrong thing. If designers are vigilant in breaking their own assumptions, they will discover the right scope for their system. The open-ended interaction between designer and customer encouraged by apprenticeship allows designers to discover where their assumptions are wrong.

Partnership In Design

We started with the notion of apprenticeship as a good way to learn about the customer. But applying apprenticeship to the special problems of design leads to a shared design conversation and a partnership between customer and designer. This partnership first affects the initial definition of requirement, but can change the whole relationship between customer and design organizations.

Apprenticeship naturally leads to a participatory approach to prototyping. If customers talk best about their work while doing it, then a prototype is tested best when customers use it to do, or mimic doing, their own work in their own workplace [6]. The customers thus demonstrate, concretely and realistically, how the prototype supports or fails to support their own work. Simultaneously, they learn what technology can do, and suggest alternative ways to apply technology to their own work. Having been partners in creating the basis for the prototype through the initial apprenticeship, they now become partners in creating the solution.

An organization that builds products for sale can make full use of this model for gathering requirements. Working closely with a single person on understanding that person's work leads quickly to a sense on the customer's part of having had a real impact on the design process. The close work with individual customers leads the design team to discover the detail and structure of a work domain they need for their design. Repeat visits to customers with prototypes at increasing higher levels of detail demonstrate the company's responsiveness while refining the design at every step. The result is a more effective design—and improved public relations.

Organizations building custom software (such as IT departments) can combine apprenticeship and prototyping with putting people from the client organization on the team. They are full members of the design team, which means they live by the rules the designers live by. The most important of these rules is a commitment to designing from actual, observed data rather than from personal experience. As customers, they give the team their special insight, perspective, and knowledge of the client organization's espoused policies. As team members they learn how to see work and how to apprentice with other customers.

They also learn how often their organization's espoused policies differ from what actually happens. This is a different role from that of "customer advocate" or "customer representative"—they are not the final arbiters of what customers want and are not solely responsible for taking the customers' point of view. As members of the team they learn how to apply technology to their problems and become full partners in working out the design.

When design is done this way, the client organization sees how to influence the design team even before the first prototype is developed. We have seen whole organizations transformed through this process—customers want to be interviewed, get excited about the new design, and feel involved in the effort. Customer team members, having learned to see work for the purpose of design, recognize how they can improve their department's procedures. Some have put together teams using their new awareness to redesign their own business processes.

Apprenticeship is a form of intense listening that leads to mutual ownership and partnership between customers and designers. Agreeing on requirements is easier when a new system is designed with the customer. Using the apprenticeship model addresses the problem of requirements definition by creating an effective relationship between designers and customers. ◘

References

1. Clement, A. A Cooperative Support for Computer Work: A Social Perspective on the Empowering of End Users. In *Proceedings of the Conference on Computer-Supported Cooperative Work, 1990 (CSCW '90)* (October, 1990, Los Angeles, Calif.), pp. 223.
2. Garfinkel, S. *Studies in Ethnomethodology.* Prentice Hall, Englewood, CA, 1967.
3. Holtzblatt, K. If we're a team, why don't we act like one? *Interactions 1,* 3 (July 1994), 17.
4. Holtzblatt K., and Beyer, H. Making customer-centered design work for teams. *Commun. ACM 36,* 10 (Oct. 1993), 92–103.
5. Holtzblatt, K., and Jones, S. Contextual Inquiry: A Participatory Technique for System Design. In *Participatory Design: Principles and Practice.* A. Namioka and D. Schuler, Eds., Erlbaum, Hillsdale, NJ, 1993.
6. Kyng, M. Designing for a dollar a day. In *Proceedings of the Conference of Computer-Supported Cooperative Work, 1988 (CSCW '88)* (Portland, Oreg.) ACM, New York, 1988, pp. 178–188.
7. Lofland, J. *Analyzing Social Settings: A guide to qualitative observation and analysis.* Wadsworth, Belmont, CA, 1971.
8. Monk, A. (organizer). Mixing Oil and Water? Ethnography versus Experimental Psychology. In the Study of Computer-mediated Communication panel at InterCHI '93, Amsterdam, 1993.
9. Orr, J. Narratives at Work Storytelling as Cooperative Diagnostic Activity. In *Proceedings of the Conference on Computer-Supported Cooperative Work, 1986 (CSCW '86)* (December 3–5, 1986, Austin, Texas).
10. Polanyi, M. *Personal Knowledge,* University of Chicago Press, Chicago, 1958.
11. Schuler, D., and Namioka A., Eds. *Participatory Design: Perspectives on Systems Design.* Erlbaum, Hillsdale, NJ, 1993.
12. Suchman, L. *Plans and Situated Actions.* Cambridge University Press, Cambridge, UK, 1989.
13. Whiteside, J., Bennett, J., and Holtzblatt, K. Usability engineering: Our experience and evolution. In *Handbook of Human Computer Interaction.* M. Helander, Ed., North Holland, New York, 1988.

About the Authors:
HUGH R. BEYER is Co-founder of InContext Enterprises, Inc., a firm specializing in coaching teams in customer-centered product and strategy design. Current research interests include the use of explicit customer-centered systems design to drive object-oriented methods.

KAREN HOLTZBLATT is President of InContext Enterprises, Inc. Current research interests include the creation of methods for design from customer data appropriate to varied organizations, products, and team structures. **Author's Present Address:** InContext Enterprises, Inc, 249 Ayer Road, Harvard, MA 01451; email: {beyer, karen}@acm.org