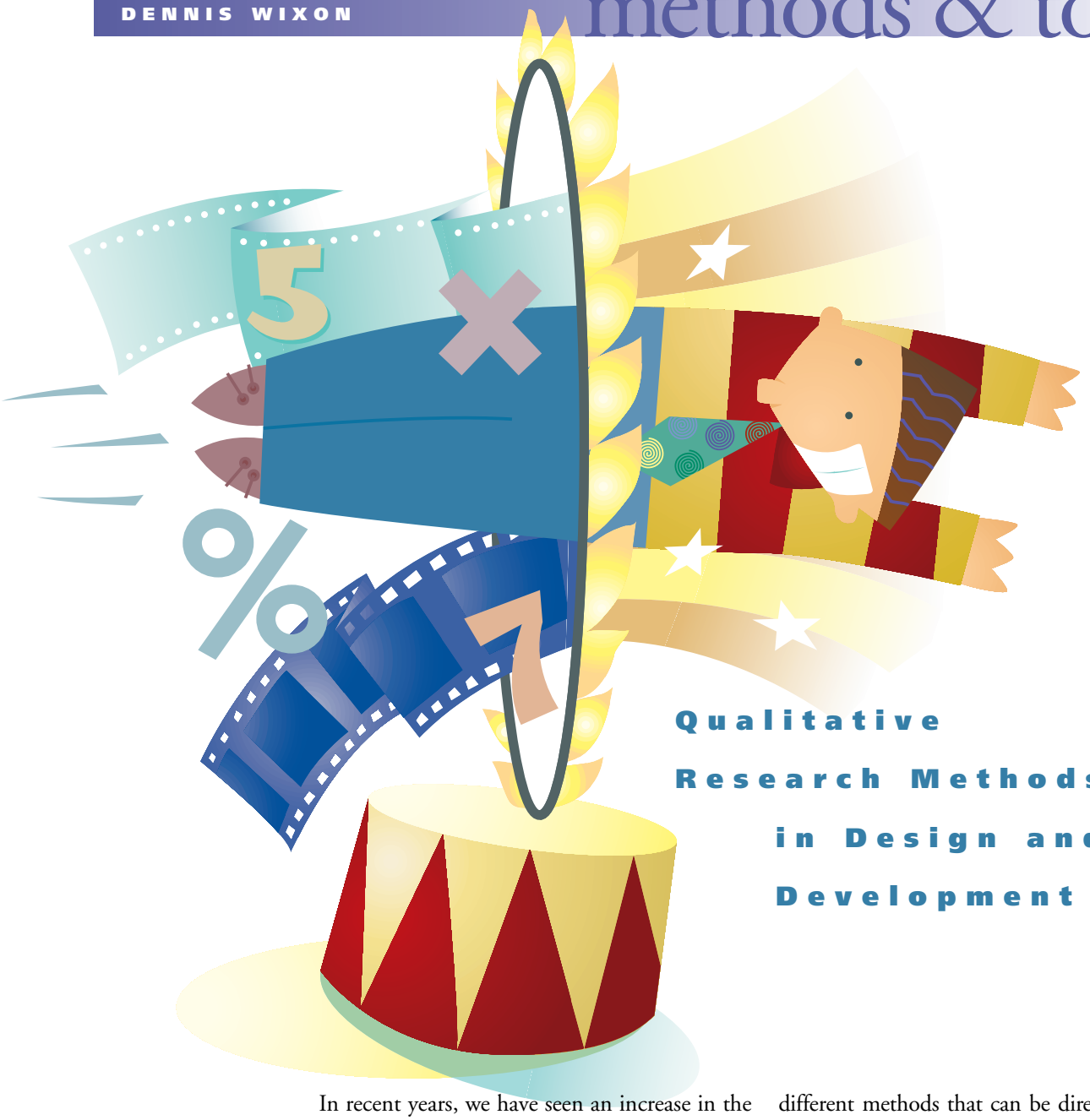
DENNIS WIXON



Qualitative Research Methods in Design and Development

In recent years, we have seen an increase in the use of qualitative and ethnographic methods for the design of systems. This rise has been motivated, at least in part, by the following factors: the need to understand user work before design begins; the inadequacy of many traditional forms of data and requirements gathering when applied to interface design problems; the need to involve users in the design process; and the increasing recognition that the transition to a modern interface metaphor, that is, a GUI, is insufficient to produce a usable system.

At the same time, we have learned that qualitative and ethnographic methods of data collection and analysis are not simply new and different methods that can be directly "plugged into" an existing development process. The methods, procedures, and fundamental assumptions of qualitative techniques have a different origin than those of classic engineering approaches. Qualitative approaches fly in the face of the engineering and marketing cultures in many companies. It's not surprising that many designers who have introduced these techniques have also described themselves as cultural change agents and offer prescriptions for cultural change and alternative approaches to engineering systems [1, 15]. The discrepancies between the engineering culture and the roots of qualitative research are often implicit, and are the source of

the challenges faced by those using qualitative methods for design in commercial settings. With perseverance, conviction, and ingenuity, and aided by the power of their findings, they have overcome many of these frustrations and have had a profound impact on products and systems. However, the fundamental discontinuity between the cultures of traditional corporate culture of engineering/marketing and the foundations of qualitative research remains unexamined. Instead of concentrating on this discontinuity, practitioners have focused specific methods for turning the "stream" of qualitative data into design. Part of their effectiveness stems from the fact that their analysis methods have been geared to answer questions inherent in the design of modern systems.

Questions such as what do users do and when do they do it; what is the intention behind user work; and how do users think about their work, are the types of questions that qualitative research methods answer.

Dennis Wixon
Digital Equipment Corporation
110 Spit Brook Road
Nashua, NH
03062-2698
wixon@usable.~enet.dec.com

Questions such as what do users do and when do they do it; what is the intention behind user work; and how do users think about their work, are the types of questions that qualitative research methods answer. This paper surveys some of these methods and offers a framework for their consideration. The framework provides a basis for organizing current practice and generating new methods. Unlike other valuable frameworks that have been proposed [12, 17, 20], this framework establishes a set of dimensions based on consideration of the kind of understanding the analysis produces and the purpose that understanding has in the broader development process.

Turning Qualitative Data into Design

When we consider ways to turn data into design, we can consider at least two methods that require little or no analysis (although they may be supplemented by more analytic methods).

The Total Immersion Method

We can start by asking if we need to do any analysis at all. Simply exposing the members of the development community to their users—particularly when it is done in such a way that the developers watch real work and absorb its richness and rhythms—can have a profound effect on design. Some practitioners have likened this process to apprenticeship [2]. At its root, this approach assumes that the richness of work cannot be captured in an abstracted analysis. If engineers "truly experience" the user's work, that will serve as an implicit understanding from which to design. In this model, much of work practice remains essentially implicit. To create tools that are "ready to hand," [11] the engineers need to embed themselves in work.

Minimalist Analysis

A close relative to the no-analysis approach is the minimalist analysis. This approach involves producing video tape records for design teams to review. The video captures the richness of work and thus avoids the filtering involved in any analysis and any potential distortions. Like the total immersion method, it can generate powerful motivations to fix problems. Even known problems become much more serious when engineers see users at major customer sites struggling with the system [24]. Minimalist analysis also can be a very fast process; highlights of field interviews can be shown on the same day that the data were collected.

Although the underlying assumptions involved in the immersion and minimalist methods may eschew formal analysis, in practice formal analyses are often used to complement such approaches. Thus, it is common for design teams

as a whole to participate in customer visits and then for a subteam to perform data analysis. These approaches not only preserve resources, but also maintain both an experiential and an analytical understanding of customer work.

Several practitioners have pointed out the effectiveness of having engineers directly observe laboratory-based tests [4, 7]. The common element may be that engineers believe what they directly observe and are skeptical of things they are told.

If these methods are quick and effective, why do any analysis? Why not simply use either of these observational methods exclusively? Although the speed of such methods is indisputable, their effectiveness needs to be examined more closely. They are most effective at producing awareness of the user's point of view, and they support designing the system from that perspective within an engineering organization that has lost sight of the user's viewpoint. To move from empathy for the user to design for the user, however, may require a more profound understanding and deeper analysis of user work. In some of our early design work, we observed a single user, redesigned the system, and observed the next user. At the end of a series of such iterations, we found that we had designed some of the original problems back into the system. By repeatedly applying superficial solutions to design issues, we had gone in a complete circle. To generate consistency and coherence in a design may require more than just repeated observations of users and interface Band-aids.

More sophisticated and time-consuming analysis techniques have different purposes, answer different questions, and, ultimately, provide different types of knowledge about user work. Therefore, it is useful to consider analytic methods as lying along various dimensions. These dimensions provide a heuristic framework for considering which method is needed in design and for generating new methods and techniques.

The dimensions are:

- Filtering vs. Reflection
- Comprehension vs. Action
- Discovery vs. Decision
- Separating vs. Integrating

Filtering vs. Reflection

Field-oriented techniques produce a vast set of detailed data. The source of this richness is partly due to the fact that the data reflect the richness of the user's work and do not derive from a limited set of questions or measures. The content of the questions and the coding of the answers are not preset as they are in many other forms of data gathering. Thus, one of the first steps in any analysis is to pull out the "relevant" data. The relevance of the data is determined by nature of the data collected and by the question that needs to be answered at this point in the design process.

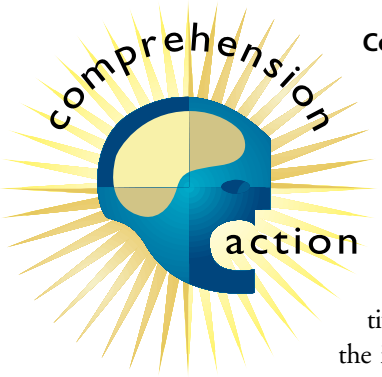
An early step in any data gathering is filtering, that is, separating the data relevant to design as a whole from that not relevant. Often, this filtering takes place in two stages. At an early stage, one simply records the "relevant" data. At this stage, the questions are conceived around broad design issues, for example, how will we deal with interrupted tasks or in what circumstances do users make errors. At later stages of the design process, the segments of this relevant data are examined in more detail because they are relevant to specific design issues. For example, when are people likely to get interrupted, when do they get to resume their work, what kinds of errors do users make, and what seems to cause these errors.

Filtering techniques answer the question: "What's relevant?" and reflective techniques answer the question "What does it mean?"

At an opposite pole of this dichotomy is reflection. Reflection differs from filtering in that in reflection, one accepts the current data set as given, but considers it from an independent perspective or vantage point. One "reflects" or bends back from the data to draw higher level, more abstract, or more general conclusions. The vantage point may involve an emerging technology; for example, how could virtual reality be applied to this problem [8]. This reflective process generates emergent design conclusions, that is, conclusions that are not stated in the data itself, but

Comprehension methods answer the question "What do we know," whereas action methods tend to answer the question "What can we do."

are generated from a variety of sources such as technological possibilities, a broader understanding of the context of work, the designer's previous experience and knowledge, and an analogy from a different, but related, domain. For example, reflection on the navigation problems in a menu system, combined with our knowledge of direct manipulation systems, led us to design a "object/operation" menu for a fourth-generation language. This one menu compressed the choices from over a dozen menus and provided a simple picture of the overall system: "Here are the objects and here is what you can do with them." In some cases, the designer is confronted with a highly cognitive task: to generate an advanced design requires one reconceptualize how the work is done. These kinds of problems require more reflection.



Comprehension or Knowing vs. Action or Doing

A second dimension to consider in both gathering and analyzing data is the dimension of understanding ("knowing") versus action ("doing"). At the understanding end of the continuum, data collection tends to be more descriptive and narrative. The goal is to preserve the integrity of the narrative of user work. It can be very detailed; for example, an entire analysis may be devoted to a single day's activities by a young child [3]. Such methods preserve the context around the work and the richness of the work itself. Thus, the details of the broader work situation are included in the analysis.

Similarly, Greenbaum and Kyng [10] have argued for consideration of a broader background in the analysis of the relationship between the "figure" of user work and the "ground" of the context of that work. They also argue that it is important to understand this broader context, at least to create a design that fits into that environmental niche. The goal of understanding leaves the researcher in a position to anticipate that the appropriate design actions will emerge from reflecting on the work of users.

When conclusions do emerge, they often have profound implications. For example, the work of Nardi and Johnson [21] suggests the fundamental approach to the design and marketing of most commercial software, that is, that adding features to large and complex products should be rethought. Instead of such large and complex products, one might create a set of interoperable task-specific products, which could be combined to match the demands of work. It is precisely the profoundness of such a conclusion that makes it challenging to apply it to the ongoing process of the development of any specific product. This sort of conclusion has implications for how products in general are designed, advertised, sold, and distributed. However, other cases are a little more straightforward. For example, designers at Kodak maintain a database of pictures of people taking pictures and making video tapes [16]. This represents a pool of information that

Table 1 **Sample Data/Action Table**

Heading	P	Action	Data	Benefit
Hardware document improvements	1	Make sure adequate (and the required) document ship with the modules and the hub	User could not configure (option)	User can configure the [option] and use the hub management tool
	1	Document 48v user programming system (UPS) interface (make clear that its UPS)	User could not find any documentation	It would be possible to use the 48v backup
	1	Explain why to use four pins/pin out setup (Quick start)	Nonstandard use caused user grief when two wires were bad	Easier for user to trouble shoot
Software document Improvements	1	Create a Quick Reference Card	User likes starter books	User can learn basics quickly

can be tapped when new photography systems are designed. It also represents a recurring return on the high investment of collecting the data.

At the other end of the understanding versus action dimension is the data/action table [22]. This table specifies the action the development team can take, the data on which the recommended action is based, and the benefit to the users of taking the action. In addition, a priority (P) is specified (in this case 1, for high, 2 for medium, and 3 for low).

This analysis is designed to produce as many implications for action as quickly as possible, while preserving the link between the data and conclusion.

This kind of approach tends to address the concern that the sponsors of qualitative research often have, that is, when will we see some results? At the same time, broader implications may be left untapped because of the constrained time and framing in which such analysis takes place.

The results of many research efforts move from understanding to action. An example is the work on "minimal" documentation. This work proceeded from a set of abstract conclusions derived from observations of the difficulties users had with traditional documentation to research demonstrating the effectiveness of redesigned manuals to a set of heuristics specifying how to design minimal documentation [6, 23].

This continuum is not intended to imply that understanding and action are mutually exclusive. Nor do we intend to imply a necessary movement from one pole to another. Instead, we are claiming that the forms of analysis, the data considered, the time taken, and the purpose of the analysis all tend to differ as one moves from one end of this continuum to another.

Discovery vs. Decision

A third fundamental dimension behind both data analysis and the overall consideration of data gathering methods in general is whether one is gathering data to discover aspects of user work or whether one is aiming to make decisions or to prioritize development work for an engineering team.

Field methods live at the discovery end of the continuum and, thus, avoid the prior specification of user tasks, metrics of performance, and measurement in general. Instead, the emphasis is on discovering what users do, how they do it, in what context they do it, and why do they do it. The emphasis is on discovering the intention behind work or the problem statement behind the solutions users often suggest. These methods are oriented toward comprehensivneness and real-world relevance.

Discovery-oriented methods answer the question "What have we learned?" Decision-oriented methods answer the question "What should we do?"

On the extreme of the decision end of the axis are traditional experimental approaches. The methodology for an experiment is a formalized decision process of rejecting one alternative at an arbitrary level of confidence. Modern product design rarely comes down to a simple decision between two unidimensional alternatives. Thus, experiments in their purest form are relatively rare in the product development process.

Some processes, such as usability engineering, include elements of the experimental approach like operational definitions and testing with specific tasks in controlled circumstances. However, usability engineering is more discovery oriented in that user behavior is directly observed and problems are identified, quantified, and prioritized. The prioritized problem list, in conjunction with an overall measure of user success (in terms of avoidance

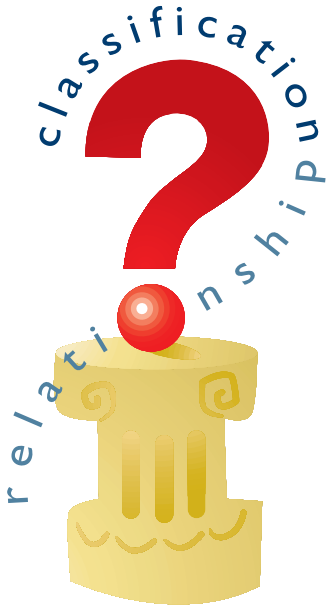
of errors, ease of learnings, efficiency, or desirability), serves as a basis for making engineering decisions. These methods are oriented toward clarity and precision.

Discovery-oriented methods are more appropriate in the early parts of the development process and in some domains could be viewed as a prerequisite to the more decision- and measurement-oriented parts of the process. The methods can be and have been used in complementary ways; for example, discovering typical user tasks has provided a basis for the empirical testing component of usability engineering.

From a design perspective, the classification answers the question "What is it?" and relationship answers the question "Where should it be placed?"

Classifying elements vs. Generating Relationships

A successful design combines the right features and organizes them into a coherent what flow. To know what the right features are, we need to see user work discretely—as the application of a set of discrete or elementaristic capabilities to a given task. Within this mindset, one analyzes qualitative data in order to generate a discrete list of customer needs. This kind of analysis provides input to design methods like quality factor deployment (QFD) [25], where user needs are first isolated, then prioritized, and, finally, compared with a possible set of engineering solutions.



Alternatively, it is possible to view work as a process that is essentially a sequence or flow of actions, or to focus on the relationship between elements. Flows and narratives emphasize the continuity of work and tools [5]. From this perspective, a product is seen in terms of how it organizes a set of controls and capabilities [14].

This distinction between the continuous and the discrete has been applied to many forms of knowledge and may represent a fundamental category of human thought roughly paralleled in verbs (connecting actions into narratives) and nouns (isolating things in space and time) [18]. This distinction can also be seen as parallel to the distinctions between usefulness (what the product does) and usability (how it does it) [9]. In fact, both perspectives are needed in the development of products. The "elemental" is a prerequisite for approaches that involve prioritization and decision. The continuous view lends itself to the sequence aspect of design such as work flow, dialog box layout, menu organization, the structure of support materials, and tasks for usability engineering.

How do analysis techniques fit in

We can look at the choice of analysis methods in terms of the dimensions outlined above. These dimensions capture the various purposes for analysis. We'll now consider some of the specific techniques [13, 19] used in qualitative analysis, and map them into the dimensions above.

Sequence models: One of the simplest forms of analysis is the sequence model. It specifies the order of actions that the user has taken to accomplish a particular task. It is primarily a filtering process based on the user's goal or intention. It tends to be oriented toward understanding and discovery, rather than action and decision. The relevant questions are which data are part of this sequence; what is the goal or intention of this work; and when does the intention change? These sequence models are often consolidated into more general activity models. They can also be captured in story boards. The process of going from a set of sequence models to a work flow model moves toward reflection, action, and decision. However, both sequence and work-flow models retain their process flavor.

Physical models: Physical models map the physical environment of the user. As such, they tend to be minimalist in tone and oriented toward understanding and discovery. Because neither time nor sequence is considered, these models are more elemental than process oriented. They are often used a check for later development decisions. For example, did we take into account that the user's work space is small and open?

Context models: Context models show the work context in terms of the people and organizational structure. They focus on influence and authority that individuals have over one another. These tend to be more reflective and oriented toward

understanding the broader design context. They are more discovery and process oriented.

Affinity diagrams: These represent a hierarchical classification of the data elements. They are produced from the bottom up, using a process of synthesis rather than analysis; that is, the categories for an affinity diagram are generated by considering the meaning contained within the data. The success of this process requires careful reflection on the data elements in order to structure them in a way that captures their meaning in relation to the design question. The approach emphasizes understanding, tends to be discovery oriented, and breaks up work into classes. The underlying question is what does the data mean? It maps well into decision-oriented processes such as QFD because these processes require that the data be abstracted into elements.

Two very broad classifications of analytic methods are matrices and diagrams [19]. These two broad categories encompass a number of specific examples, which are not covered here. These two broad classes, however, represent the dimension of elemental (matrices) versus process (diagrams).

Looking beyond initial data gathering and preliminary analysis, as the design process moves forward, models tend be consolidated across users and designers begin to focus on work redesign. Thus the analytic techniques tend to shift toward reflection, action, and decision. These methods can retain either an elemental or a process flavor, depending which aspect of design one is focusing on—features or flow.

Conclusions

We have offered a set of dimensions in which to consider qualitative or ethnographic data gathering and analysis in relation to design. The dimensions are based on the underlying purpose of the analysis. Other classification schemes are more straightforward and consider such dimensions as when can the method be used, how involved are the participants, how many people can participate at one time, and where was the method developed (United States versus Europe and industry versus academia); whereas, by contrast, the framework offered here looks at the purpose of or question behind the analysis and the form of the representation of the results.

Fundamentally, we believe that no analysis of methodology is categorically right or wrong. Instead both methodological frameworks, like methods themselves, need to be judged according to their purpose. Thus, the dimensions proposed here are intended to be descriptive, rather than prescriptive. Furthermore, different qualitative methods tend to complement each other, and other, approaches. When the results gathered from different techniques point to similar conclusions, our confidence in the conclusions is increased. In the future, we can expect to see a proliferation of qualitative and field methods as designers focus on extending technology to new domains and on uncovering ways for new technology to transform existing work. In other words, the future of field-oriented design methods looks bright.

References

[1] Allen, C.D. Succeeding as a clandestine change agent. *Communications of the ACM*, 38, 5 (1995): 81–86.

[2] Beyer, H., and Holtzblatt, K. Apprenticing with the customer. *Communications of the ACM*, 38, 5, (1995): 45–53.

[3] Barker, R. G., and Wright H. F. *Midwest and Its Children: The Psychological Ecology of an American Town*. Archon, Hamden, CT, 1971.

[4] Butler, M. B., and Ehrlich, K. Usability engineering for Lotus 1-2-3 release 4. In Wiklund, M., ed. *Usability in Practice: How Companies Develop User-Friendly Products*, Academic, 1994.

[5] Carroll, J. M. *Scenario Based Design*. John Wiley, 1995.

[6] Carroll, J. M. *The Nurnberg Funnel*. MIT Press, 1990.

[7] Dumas, J., and Reddish, J. *A Practical Guide to Usability Testing*. Ablex, Norwood, NJ, 1993.

[8] Good, M. Participatory design of a portable torque-feedback device. In *Proceedings of ACM CHI'92 Conference on Human Factors in Computing Systems* (1992), pp. 439–446.

[9] Gould, J., Boies, S., and Lewis, C. Making usable, useful, and productivity-enhancing computer application. *Communications of the ACM*, 34, 1: 75-85.

[10] Greenbaum, J., and Kyng. M. The design challenge-creating a mosaic out of the chaos. *Opening Plenary Address, CHI'95.*

Acknowledgments

A number of the examples presented above are taken from a workshop organized by Dennis Wixon and Dr. Judy Ramey at CHI'95. We gratefully acknowledge the participants: Kristin Bauerfeld, Diane Brown, Mary Beth Butler, Janette Cobel, Susan Dray, Bob Graf, Shanon Halgren, Diane Juhl, Michael Muller, Stan Page, Martin Rantzer, David Rowley, and Larry Wood. We also acknowledge the work of Alicia Flanders, Anne Smith Duncan, Mary Beth Raven, and Minette Beabes in extending the concepts of Contextual Inquiry. Finally, we acknowledge Karen Holtzblatt and Sandra Jones for originating a number of the concepts and methods of Contextual Inquiry, and Michael Muller for his thoughtful editorial comments.

[11] Heidegger, M. *Being and Time*. Harper and Row, 1962.

[12] Hix, D., Hartson, H.R., and Nielsen, J. A taxonomic model for developing high impact formative usability evaluations. *Workshop at CHI'94, in CHI'94 Conference Companion*. Boston, MA, ACM, 1994, p. 464.

[13] Holtzblatt, K., and Beyer, H. Making customer centered design work for teams. *Communications of the ACM*, 38, 5 (May): 93-103.

[14] Huntwork, P. K., Pietras, C., Muzzey, D., and Wixon, D. Changing the rules: a pragmatic approach to system development. *Digital Technical Journal*, 5, 4: 18–36.

[15] Hutchings, T., and Knox, S. Creating products customers demands. *Communications of the ACM*, 38, 5 (May 1995): 72–81.

[16] Kaplan, S. Making usability a Kodak product differentiator. In Wiklund, M., ed. *Usability in Practice: How Companies Develop User-Friendly Products,* Academic Press, 1994.

[17] Kensing, F., and Munk-Madsen, A. PD: Structure in the toolbox. *Communications of the ACM*, 36, 6 (1995): 78–85.

[18] Maxwell, J. A. A synthesis of similarity contiguity distinctions. *Poster at the Annual Meeting of the American Anthropological Association.*

[19] Miles, M. B., and Huberman, A. M. *Qualitative Data Analysis*. Thousand Oaks, CA, Sage Publications, 1994.

[20] Muller, M. J., Wildman, D. M., and White, E. A. Taxonomy of PD practices: a brief practitioners guide. *Communications of the ACM*, 36, 4 (1993): 25–28.

[21] Nardi, B., and Johnson, J. User preferences for task-specific vs. generic application software. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems* (1994), pp. 392–398.

[22] Raven, M. B., and Flanders, A. Using contextual inquiry to learn about your audiences. *Journal of Computer Documentation* (in press), 1995.

[23] van der Meij, H., and Carroll, J. M. Principles and heuristics for designing minimalist instruction. *Journal of Technical Communication*, 233–251, 1995.

[24] Wixon, D., and Jones. S. Usability for fun and profit. In Rudisil, M., Lewis, C., Polson, P., and McKay, T., eds. *Human Computer Interface Design: Success Cases and Real World Context*, Morgan Kaufmann (forthcoming).

[25] Zutlner, R. E. TQM for development teams. *Communications of the ACM*, 38, 5 (May 1993): 79–91.