Using Participatory Design of User Documentation in Support of Learning at Work

Trudy Williams
NYNEX
Science & Technology, Inc.
500 Westchester Avenue
White Plains, NY 10604 USA
+1 914 644 2325
trudyw@nynexst.com

ABSTRACT

This is a brief account of using participatory design with a group of telecommunications engineering drafters to develop user documentation for a new software application that replaced an existing on-line system.

Participatory design surfaced the drafters knowledge about how they use documentation in learning at work. These insights were then used to generate design requirements of user documentation that would best support their learning processes at the time of system deployment and beyond.

Keywords

User participation, end user documentation, learning at work

INTRODUCTION

Although used in software development, a Participatory Design (PD) approach is not often used to develop user documentation.

During the deployment phase of software development, common assumptions about documentation are that:

- The deployment of a software application involves giving the people training sessions and user documentation.
- The documentation will mostly sit on the shelf gathering dust throughout the implementation process and thereafter.
- Documentation deserves its wall flower status; it is probably hard to use or doesn't cover the right things. Even if it is used somewhat right at the very beginning, it begins collecting dust soon after.

Data from a user community of telecommunications engineering drafters engaged in participatory design of user documentation give this view of documentation:

- The user documentation delivered at the time of software deployment forms the basis for a series of activities as the users go about organizing their learning.

  Except if the documentation is inadequately written, the fact that it is used most intensively at the beginning and less so thereafter is not a reflection of its deficiencies but a natural event in a learning process.
- As they organize and facilitate their own learning processes, people "do" things with the documentation: they harvest from it to make their own versions, they re-arrange it, change it, invent their own forms of it. The standard documentation delivered at deployment is the initiating source for creating these documentation artifacts.
- The documentation artifacts that accompany various stages in the learning process are often a product of both collaboration with co-workers and expansions of artifacts of prior documentation.

Using insights into how they used documentation to support their learning activities in relation to the existing software, the drafters generated the design requirements for the user documentation for the new software that was coming into their workplace.

What follows is a description of:

- the user community, their workplace, and the type of work they do
- shared experiences and insights that surfaced during participatory design about how they used documentation in their learning activities
- the resulting design requirements for documentation that supports their learning at work.

The final section provides some more details about how the project developed and raises the issue of on-line documentation.

THE USER COMMUNITY

Approximately four years ago a company-wide posting went out for job openings for engineering drafters. Training was provided.

In *PDC'96 Proceedings of the Participatory Design Conference.* J. Blomberg, F. Kensing, and E.A. Dykstra-Erickson (Eds.). Cambridge, MA USA, 13-15 November 1996. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

Eleven people transferred into the job; only two of them had experience working with a computer. They received training sessions followed by continuous coaching on the floor by a manager and assistant manager. The drafters and their managers agree that it took about three to four months for people to really learn the system, and about a year before they felt they had fully integrated using the system.

The drafters (nine women and two men) work a compressed work week consisting of four 10-hour days, with either Monday or Tuesday as a day off.

They work in an open space divided into areas by the configuration of desks. Each desk has a drawing board (made obsolete by the drafting software tools) that many drafters have tilted upwards for some privacy. Areas are personalized with mementos, plants, photos, jokes and favorite sayings.

The group has been very stable over the four years. All have been there from the start except two who left and were replaced by two others six months after the initial group formed.

Their work consists of using a software application to convert engineers' markups of central office plans into on-line drawings and/or to update drawings that are already in the on-line system.

A collaborative community exists. Two or three people will gather around a monitor to figure out how to draft something. Others may go to the manager or assistant manager, find the answer to a drafting problem and share the solution with other drafters back on the floor. Everyone knows who is good at what, who to approach for advice on how to solve what types of drafting problems.

The software they are currently using is being replaced by another software which has a number of features which facilitates the speed and accuracy of their work. The new software is not just an enhancement of what they have been working with for four years, but an entirely new application that they must learn to use.

USING PARTICIPATORY DESIGN

The drafters were given no formal introduction to "participatory design" or "design requirements". I asked them to work as a group with me (as the development team member actually responsible for writing up the documentation) so that they could use what they know about doing their work, and how they learned to do it, to generation the design requirements for documentation of the new software that would support their learning activities.

The drafting manager (who was a member of the application development team which included engineers, software designers and other managers responsible for the requirements of the engineering and drafting applications) was supportive of this effort and responsive to requests for scheduling time with the drafters individually and as a group over three weeks time. Drafters were enthusiastic; two people expressed relief that "finally" they were asked to participate.

We met initially as a group away from the computers and then at the computers. At any given time, there were three to nine of the drafters present. Meetings at the computers sometimes resembled 'grand rounds': the group, in varying size, gathered around one workstation to the next for an explanation, demonstration and discussion of the documentation artifacts that were used to help learn the system and to do the on-going work.

The documentation produced from the participatory design was piloted along with the software trial. Drafters worked in teams of two and three per workstation to have the benefit of collaborating in learning. As they worked at learning how to use the new software, they evaluated the documentation in terms of what worked, what didn't work, what would work better. They continued to evaluate the documentation at the time of deployment and afterward.

HARVESTING AND DOCUMENTATION ARTIFACTS

The participatory design process yielded agreement that documentation supported their learning activities and surfaced these shared views and practices:

- The documentation delivered with the "old" software was thorough but hard to learn from: it gave more technical information than was needed to do work; it was organized according to groupings of menu options and application functionality rather than according to the sequence of how the work gets done.
  - The drafters harvested the information from the given documentation. In their own styles they re-worded it, re-sequenced it and/or made illustrations. They used the documentation they created in place of the standard documentation to learn how to use the system to do their work.
- They created document artifacts from the harvested information.
  - Some drafters assembled boxes of index cards organized around work sequences, others made rolodexes of menu options related to the sequence of drafting, several people posted (and re-positioned as needed) key information cards around the outside edge of their monitors.
- The harvesting and creation of documentation artifacts took place in a collaborative environment.
  - Most drafters created folders of problem drafting jobs along with the solutions that were solved in collaboration with other drafters. It was considered OK to look at another person's folder even if they were away from their workstation.
  - When someone needed help, they walked around asking "has anyone done a drawing like this....". The solution was usually written up, copied and passed out to everyone.
  - The collaborative effort produced solutions based on a common characteristic: things were solved best when the collaboration produced a full view of

how the work got done rather than by exact knowledge of each functional entity accessible from the software menus.

THE RESULT: DESIGN REQUIREMENTS

The group produced these design requirements for the documentation for the new software:

- Organize the documentation using the agreed on sequence of how the work gets done.
- Produce the document in a loose leaf binder so pages can be added, deleted or re-arranged.
- Leave room in the margins for notes.
- Pilot the documentation at the same time the software is deployed in a field trial.
- Illustrate the work process, if possible.
- Include a section that puts drafting in the context of the engineers work.
- Include how to interface with other software applications at the point it needs to happen when doing the work. Do not direct users to separate software documentation.
- Have a section of conventional description of each menu item for people who want to develop alternate ways of doing the work.
- Include a chapter of drafting symbols and required parameters so drafters do not have to reply on memory or consult unwieldy hardcopy across the room.
- Use participatory design until the system is fully deployed. Continue to use it when there are changes and updates to the system.

The documentation produced from these requirements has the appearance of standard documentation. A closer look shows that chapters are diverse in content, with most of them organized not around application functionality but according to aspects of the work processes.

In the software and documentation trial and through the current limited deployment, drafters have reported that the documentation supports their learning activities and is easy to harvest.

ABOUT WHAT HAPPENED

The approach and methodology for the design of user documentation was grounded in the participatory design assumptions of:

- the value of worker's knowledge about what they do [1], [2], [3], [4], and the need to include them in design of artifacts that affects their work lives [5]
- the importance of user-developer understanding and communication [6], [7], [8], [9], [10], [11]
- the usefulness of looking on-site at how people communicate, problem solve and learn as they do their actual work [4], [12], [13], [14], [15].

The project's plan called for a combination of contextual inquiry techniques [16], individual meetings and group meetings using various activities to establish common ground and develop shared representations [2], [7]. The nature of the 'grand rounds' that happened after the initial introductory meeting and the first few individual workstation meetings with drafters was unanticipated.

Supported by the notion that a PD approach retains an element of improvisation [5], plans for meeting away from the workstations were abandoned. The 'grand rounds' format was cultivated for its potential to:

- provide a context for shared representations and the emergence of user knowledge in understanding how documentation was used in support of learning
- retain the benefits of workplace 'reminders' for surfacing insights possibly overlooked in a site remote from the place where work gets done [16]
- avoid the problem of 'user proxy' whereby mediated representations rather than direct artifacts are used for generating common understanding [7].

One area worth exploring that could not be investigated is the role of on-line documentation in the support of learning at work. Since the new drafting application did not have on-line documentation, there is no data from this project as to whether or not on-line documentation (such as user manuals, help screens and hypertext) inhibits or facilitates the creation of documentation artifacts in the 'harvesting' processes as workers engage in organizing their learning.

CONCLUSION

Using participatory design in documentation development surfaces users' knowledge of what documentation artifacts exist for systems or work processes currently in use, created when and by whom, to accompany what phases of learning. This allows the participatory design team to generate the requirements for user documentation for new or updated applications that supports the individual and collaborative efforts involved in learning a new system.

ACKNOWLEDGMENTS

For their vision and support, I would like to thank Ivy Eisenberg, Marsha Ellis, Jim Euchner, Jerry Fiorello, Genevieve Foley, Larry Kamenetsky, Patricia Sachs and Rangnath Salgame.

REFERENCES

1. Whyte, W.F., Ed. *Participatory Action Research.* Sage, Newbury Park, 1991.

2. Kensing, F., and Munk-Madsen, A. PD: Structure in the toolbox. *Communications of the ACM,* June 1993, 36, 4. 78-85.

3. Grudin, J. Interactive systems: Bridging the gaps between developers and users. *IEEE Computing,* April 1991, 24.

4. Suchman, L.A. *Plans and Situated Action: The Problem of Human-Machine Communication.* Cambridge University Press, Cambridge, 1987.

5. Clement, A. and Besselaar, P. Retrospective look at PD projects. *Communications of the ACM,* June 1993, 36, 4.

6. Greenbaum, J. and Kyng, M. Eds. *Design at Work: Cooperative Design of Computer Systems.* Erlbaum, Hillsdale, N.J., 1991.

7. Kyng, M. Making representations work. *Communications of the ACM,* September 1995, 38, 5.

8. Keil, M. and Carmel, E. Customer-Developer links in software development. *Communications of the ACM,* May 1995, 38, 5.

9. Kensing, F. and Madsen, K.H. Generating visions: Future workshops and metaphysical design . In *Design at Work: Cooperative Design of Computer Systems.* J. Greenbaum and M. Kyng, Eds., Erlbaum, Hillsdale, N.J., 1991.

10.Schuler, D. and Namioka, A., Eds. *Participatory Design: Perspectives on Systems Design.* Erlbaum, Hillsdale, N.J., 1993.

11.Anderson, W. The wall: An artifact of design, development and history. *Procedures of the Participatory Design Conference,* 1994.

12.Bodker, S., Greenbaum, J., and Kyng, M. Setting the stage for design as action. In Greenbaum, J. and Kyng, M., Eds. *Design at Work: Cooperative Design of Computer Systems.* Erlbaum, Hillsdale, N.J., 1991.

13.Nardi, B.A.. Studying context: A comparison of activity theory, situated action models, and distributed cognition. In *Context and Conscionsness: Activity Theory and Human-Computer Interaction.* Nardi, B.A., Ed.. MIT Press, Cambridge, 1996.

14.Holtzblatt, K. and Beyer, H. Making customer-centered design work for teams. *Communications of the ACM,* October 1993, 36, 10.

15.Sachs, P., Transforming work: Collaboration, learning, and design. *Communications of the ACM,* September 1995.

16.Holtzblatt, K. and Jones, S. Contextual inquiry: A participatory technique for system design. In *Participatory Design: Principles and Practice.* A. Mamioka and D. Shuler, Eds. Erlbaum, Hillsdale, N.J., 1993.