Does Participatory Design Have a Role in Packaged Software Development?

Erran Carmel (moderator)
Asst. Professor
Kogod College of Business
The American University
Washington D.C., USA
E-mail: ecarmel@american.edu

Jonathan Grudin
Assoc. Professor
School of Management
University of California
Irvine, California, USA
E-mail: grudin@ics.uci.edu

Dianne Juhl
Usability Specialist
Microsoft Usability Group
Microsoft
Redmond, Washington, USA
E-mail: diannej@microsoft.com

Tom Erickson
Designer, User Experience Architect's Office
Apple Computer
Cupertino, California, USA
E-mail: erickson3@applelink.apple.com

Jamie Robbins
Applications Development Marketing Manager
SAS Institute
Cary, North Carolina, USA
E-mail: sasjhr@vm.sas.com

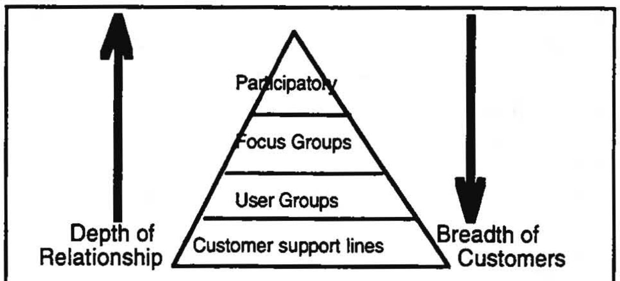
ABSTRACT

For Participatory Design (PD) the domain of packaged software development presents a challenge. PD evolved -- and is largely practiced -- in an environment that produces custom systems for corporations and governments. Package developers are still in the early stages of using PD approaches. The panelists will discuss the opportunities and obstacles for PD use in packaged software development, as well as some early experiences at major companies.

INTRODUCTION

Although only about six percent of the world's software community develop packaged software (or "products"), the impact of packages on the workplace is far greater than that six percent represents.

In millions of workplaces, users interact with packages such as groupware, word processing, accounting systems, reporting systems, etc. Meanwhile, "systems integrators" are creating "custom" systems by mixing and matching packaged software options.



A pyramid of developer-user links in packaged software

What are the key differences between package development and custom development? First, the user (customer) may be unknown to the developer. Second, once users are identified, developers need to select a subset of users to engage in product design. Third, given that the mean distance of the developer from the typical user is well over one thousand miles, developers need to find a way to establish a close relationship with the user during the life of the product. The above pyramid illustrates the hierarchy of links that developers use to listen to and interact with their users. The model suggests that, in the packaged software world, developers need to set up complementary links to cover the depth and breadth of user interactions.

In *PDC'94: Proceedings of the Participatory Design Conference.* R. Trigg, S.I. Anderson, and E.A. Dykstra-Erickson (Eds.). Chapel Hill NC USA, 27-28 October 1994. Computer Professionals for Social Responsibility, P.O. Box 717, Palo Alto CA 94302-0717 USA, cpsr@cpsr.org.

JONATHAN GRUDIN: PD IS BEST USED FOR NOVEL PRODUCTS

The maturity of an application or application type differs across development projects, as does the relative emphasis on functionality and interface. These are not tightly coupled, but novel application development projects are more often focused on functionality and mature applications seek out interface enhancements to differentiate standard functionality. Most internal development projects have a high degree of novelty (as seen by the participants); package or product development is more variable. Users are more likely to be interested in and knowledgeable about high-level functionality, which is closer to their work goals, than they are to interface refinements. Kyng (1994) described an application of participatory design to product development; it provided input at architectural, functional, and interface levels. This is more useful for novel products, where all levels may beopen to change. Similarly, cooperative prototyping, emphasizing changes to the design that can be made in the course of a work session, allow for quick, major course corrections but not for quantitative measurement that may be needed to settle minor design decisions on which not all users will agree. Thus, participatory design and usability testing need to be merged into a single process for package software development, with the former used for novel products or when a major revision of a product is being contemplated and the latter for minor revisions of more mature products or features.

DIANNE JUHL: EXPERIENCE WITH PD TECHNIQUES AT MICROSOFT

Microsoft develops both generic software tools for the corporate or business domain and task-centered applications for the consumer domain, and we work with a variety of users from both domains throughout the entire design process. We recognize that partnership with users is important and that users are the experts about their domain. At Microsoft, we utilize the expertise of Usability specialists and program managers to collect the data, make sense of it, and then infuse that data in software's design. The Usability Group creates a variety of situations in which users can give us input on software productdesign. In the planning stage of the product development cycle, we conduct research using methodologies such as contextual inquiry and field studies. After some basic design decisions have been made, we seek users' input on the paper prototypes of the design. As we move further into the development cycle, we conduct rapid, iterative tests on prototypes to uncover usability issues. Finally, in the quality assurance stage of the design process, we test the fully functional software in the usability lab or at field sites. All of these strategies depend upon an on-going conversation with the user.

In my work with the Microsoft Consumer Division product teams, we are using contextual research to gather data from users in the home. Since many Consumer Division product teams are creating task-based software titles for the consumer market, contextualresearch seems to be the most appropriate tool for collecting the data we need. While it is useful to observe people interact with software products in a usability lab or field setting; in this case, it is necessary to gather extensive, detailed information about our users in their own context. In addition, this methodology, as developed by the Microsoft Usability Group, is a well-defined sequence of activities for gaining an understanding of the users' tasks and activities. This methodology also ensures product teams can easily partner with users over time to inform design decisions for future versions of the software application.

TOM ERICKSON: COST/BENEFIT OF PD IN HORIZONTAL MARKETS

In the canonical case of participatory design, users and designers work together over a long period to craft a system uniquely suited to the tasks, practices, and environment of its users. However, there are difficulties in carrying out this type of practice in the development of horizontal products. There are issues of costs versus benefits for both users and development organizations. Generally, `users' of new horizontal products are only potential users. What will they get out of participating in a long term design exercise, and will that outweigh the necessary commitment of time and resources to something that will ultimately be used by a much broader audience, including their competitors? Similarly, do the benefits of participatory design outweigh real and perceived costs to the development organization? For example, for a development company to agree to a participatory exercise with a third party, concerns about intellectual property issues and loss of confidentiality must be addressed.

In general, it seems to me that the costs of participatory design are higher for developers of horizontal products, and the benefits for users (as participants in the process) are lower. This does not rule out the application of participatory design, but rather modulates the frequency and degree to which participatory design techniques are employed. In the present and immediate future, I see the following happening: First, something approaching canonical participatory design may happen in the development of new product concepts, or new niche products, assuming that a relationship with appropriate users can be developed. Second, participatory design techniques -- minus the long term relationship with a particular set of users -- may be employed in the development of products that fall into existing categories. The farther future holds more interesting possibilities: The development of standards such as OpenDoc that allow different developers to create software components that work together has the potential to blur the distinction between horizontal products and custom-built products. On the one hand, this can allow horizontal product developers to specialize in a particular software component, and to develop custom versions for various niches -- thus raising the benefits of participatory design for that organization. And on the other hand, the availability of a wide range of interoperable software components lowers the cost of customization for users, thus increasing the potential forparticipatory design within users' organizations. How this will all play out remains to be seen.

JAMIE ROBBINS: PD IN HORIZONTAL PRODUCTS AND IN CUSTOMIZABLE PRODUCTS

Since SAS Institute Inc.'s beginnings in the 70's, we have been using participatory design methods to continually enhance the software solution we offer to over 27,000 sites worldwide. We have seen the importance of customer input into software design and have taken proactive steps to get that input. Each year we poll our users to ask what improvements, changes, or additions they would like to see made to the software. Our track record for incorporating the top vote-getters on this ballot is excellent. Our usability labs and alpha and beta test offerings also encourage customers to put the software to the test and participate in the design and development of the production product.

As a vendor of applications development software tools, our customers often ask us for guidance in their applications development. We take this opportunity to encourage participatory design in their organization by using an iterative approach to applications development. The knowledge transfer that occurs between the business unit and the application development staff not only helps both parts of anorganization learn each other's functions, it also provides a basis for applications that meet the end users needs.

REFERENCES

Kyng, Morten. Scandinavian Design: Users in product development. *Proceedings of Human Factors in Computing Systems*, pp. 3- 9, 1994