Usability literature is full of "oughts" and "what should be's". Too often we hear complaints that the "oughts" and prescriptions cannot be implemented in the workplace. There are a number of industries developing products with user-centered approaches that can serve as "real life" or "role" models for those trying to start or improve their processes. This paper results from a CHI '91 SIG (same title) whose goal was to focus on real-life cases illustrating processes and evaluation: how to begin a usability process and keep it viable, how to bring about feedback into design, work within and around established norms, and set a process in a new direction. The contributors, who represent consulting from both within and outside of a company, are concerned with the mechanisms and values which create, build on, and nurture successful processes, as well as the pitfalls we might encounter.

User-Centered

Evaluation Processes

photos © Photoworld/FPG International

We look at various development stages and types of processes in different settings, with "micro", "macro" and "global" foci. The "micro" process view is a case study of the usability professional and the developer working together. Karimi and Cypher illustrate how a design is improved and extended as it is refined by user input. They show the process through which an engineer and a social scientist conducted a user study and how their complementary perspectives influenced the interface design for Eager, an intelligent assistant for users of the HyperCard environment. The "macro" view is a case study of a software development group and user-centered processes built within that group. Kvavik discusses mechanisms and pitfalls in initiating and expanding usability processes and participation within a large development group where designers, developers, HCI consultant, end users, and others are involved. Finally, the extended or "global" view of a consultant from outside of the company examines when it is possible to put usability engineering processes in place. Mayhew elaborates what catalyzes changes, effective practices for the professional who is the change agent, and factors which contribute to success or failure of usability processes.*

Interdisciplinary Cooperation—Feedback into the Design Process

There is a lot of talk about the benefits of interdisciplinary work throughout the design cycle. However, in reality there are organizational and personal barriers to effective interaction among people in different disciplines. n his paper, "Interdisciplinary Collaboration", Scott Kim discusses the importance of "noticing the assumptions in your own discipline that might limit your view of the world" [1]. We present our experiences as a case example of some of the problems and successes in a collaborative effort to improve the design of the user interface for Eager, a programming-by-example system.

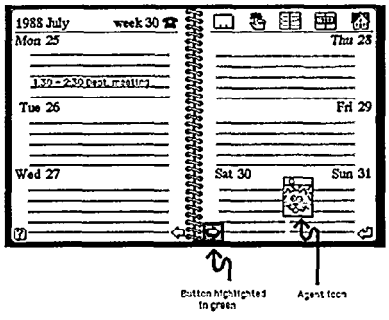
About the Authors:
ALLEN CYPHER
is a research scientist at Apple Computer, working in the area of programming by example. email: cypher@apple.com

SHIFTEH KARIMI
is a user studies manager in the Advanced Products Group of Apple Computer. email:(karimi@harry.apple.com)

What is Eager?

Eager is a program for automating repetitive tasks [2]. It constantly monitors user actions, and when it detects a pattern, it writes a generalized program to perform that repetitive activity. The following is a screen illustration of Eager

The design of the interface was based on the principle that the program should cause minimal intrusion into the user's normal interaction with the computer. The design included the representation of an "agent" icon on the screen and the use of a new interface technique, called "anticipation", where the agent indicates what it expects the user to do next by highlighting that item in green. Since intelligent agents are innovative software features and users are not accustomed to interacting with them, it was necessary to test empirically the initial reaction of users to these features. The two main issues were the ease of use and the functionality of the system.



The "Eager agent appears, and the next anticipated user action is highlighted in green

Our Collaboration

Organizationally, we were separated. We both were in the Advanced Technology Group, but we worked in two different departments. Although there were no formal mechanisms for collaboration across departments, there was a general sentiment that cooperation across disciplines is valuable. We knew about each other's work and did not need to go through formal approval procedures.

User-Study Design

Allen, the engineer, approached Shifteh, a user studies specialist at that time in the Human Interface Group, with a list of questions. He had ideas about how a user study should be conducted. For example, he expected to first explain and demonstrate the program and then have the subjects use it themselves. But since the main goal of the study was to understand users' initial reactions, Shifteh suggested that they observe subjects' first expe-

riences with the system without providing any prior explanation.

A longitudinal study would be an ideal approach in unfolding user responses over time in their natural work settings. However, given the constraints of time and resources for conducting a longitudinal study, Shifteh suggested a study design based on specific tasks that were relevant to the user's everyday work life.

In addition to observation methods for collecting users' reactions, it was important also to follow the actions of the computer's intelligent agent behind the scene. Allen suggested the idea of modifying the program to produce trace data to give us an accurate recording of the actions of the intelligent agent. This mechanism of data collection was an additional tool that augmented the existing data collection methods.

One of the questions Allen wanted answered by the study was when the agent should appear on the screen to assist the user in completing a repetitive task. From his technical perspective, he wanted the agent to appear as soon as possible so that it could be of most help to the user. However, there was a tradeoff. The sooner the agent appeared, the more likely it would be for it to incorrectly guess the pattern in the users' actions. Allen wanted to test two versions of the program: one with early and one with late appearance of the agent.

From a psychological perspective, Shifteh knew that users would be confused if confronted with an intelligent agent that made mistakes. The idea that a program could be incorrect was in sharp contrast to people's current mental model of computers. So, instead of testing two versions we used only the late appearance version, the one with a greater likelihood of correctly detecting the pattern. Indeed, the results of the study supported this decision to such a degree that Allen later changed the program to delay the appearance of the agent even longer.

Feedback into the Design

Allen was present throughout the testing sessions as a silent observer. He wanted to experience those detailed aspects of user reactions that could not be captured on video or in a formal report. At the end of each testing session we discussed our observations about users' responses and reactions. Our different orientations often led us to different interpretations of the data.

For example, Shifteh noticed that some users did not recognize that the green highlighting and the agent icon were two aspects of the same system. Allen, on the other hand, could not imagine that this misunderstanding could occur. He knew the system so well that this finding could not fit his perception of the program. He kept trying to explain the data in a different way to convince Shifteh that her observations were incorrect.

We tried to resolve our differences by reviewing the transcripts of the session. We found comments like these:

- "I was curious about him [the agent] but I didn't know exactly what it was, and I was seeing some green things."
- "I thought it [the agent] was a button, and the green [highlighting] seemed to have something to do with what I had done. The green came on at the same time; it showed what I was going to do next, and the icon was gonna do the task for me."

To Allen, the fact that users had noticed the two features and talked about their connection was proof that they understood the relationship. Our discussions revealed that subjects were verbalizing their reactions because they were having a difficult time understanding the relationship between the two interface features. As a result of this study, Allen changed the agent icon to appear in green, so that its connection with the green highlighting is more obvious.

Conclusion

Our experiences demonstrate how engineers and psychologists can go beyond organizational constraints and personal biases of their disciplines and turn the design process around. By our willingness to work together in conducting a user study we learned about each other's world, and clarified misperceptions about how to interpret user data. We recognized the value of combining trace data with observational methods. This interplay of objective and subjective methods was particularly useful in our situation where we had limited time and resources.

Starting and Expanding User-Centered Processes

This is a case study about building processes for user-centered software within a development group, with attention to the techniques, pitfalls, and values involved in those processes. With a management mandate for usable software, the usability effort began by establishing field research practices and feedback into design and getting co-optation of end users, developers, management, and others. Expansion efforts consisted of building participation through group processes for both interface redesign and guidelines development.

About the Authors:
KAREN H. KVAVIK
is a consultant on user-centered design and development issues and has her own firm, Karen Kvavik & Associates. Her experience also includes six years as an HCI and AI Consultant with Unisys Corporation. email: (kvavik@vx.acs.umn.edu)

The development group portrayed builds expert-system based software tools for configuration of correct and complete orders of company hardware and software products. At the time, the author was an HCI professional with the group. Targeted end users are company-internal marketing and sales world-wide. In addition, the development group interacts with others ("clients"). The clients contribute to company product development (management, hardware engineers, field engineers, and information providers). In reality, some clients may also be end users of the software tools but are not targeted as such.

Starting—Getting Acceptance and Buy-In with Field Research

Field research was a powerful vehicle for achieving iterative end-user participation and evaluation by involving end users, developers, and the HCI professional in a mini-participative process, and for getting client involvement and end user acceptance of the configuration tools [3]. The developers and the HCI professional set goals to be accomplished during an evaluation which used an ethnographic approach (at the user's workplace), with think-aloud techniques and structured probing when appropriate, and a session-final wrap-up to ensure that the user feedback was understood correctly. These particular techniques were used because of time pressures to obtain data quickly for design or development.

Adoption of field research occurred unevenly across the group, as in any technology change and spread; some developers were outwardly resistant at the beginning. With acceptance, developers saw user data as necessary for design and improvements. End users were pleased to be consulted and to see their feedback in the software; this led to acceptance and ownership of the new tools. The user data and improvements resulted in a better management commitment to usability efforts. Technology transfer took the form of informal explanations, teamwork, and "hands-on" approaches for developers, and explanation of rationale to users. A formal training session was given to an ancillary (client) group who later tried field research techniques on further clients (information providers). This ultimately led to improved information for the development group and better company documentation on a product.

Expanding Processes: Interface Committee and Redesign Efforts

The interface committee process began because management wanted design guidelines for ongoing development. Management believed that the guidelines were a short-term task and could be done in several weeks because "everybody knows what our practices are." The end users of the guidelines would be "developers." However, software group members disagreed as to what the existing interface practices were and recognized that inconsistent implementations and differing system architectures were a problem across the various software tools. Also, user data and configurator development for new hardware and software products indicated that a long-range interface design effort was necessary. Moreover, as potential users, developers believed they should participate in developing guidelines they would have to use. With management permission, six development group members voluntarily formed an interface committee with interface redesign and guidelines as two separate but intertwined goals.

The committee realized that it needed to build ongoing processes for design innovation and resolution of consistency issues, and that it needed to build group acceptance of new designs and of the interface design process. As the redesign issues emerged, different individu-

als volunteered to participate, with management requiring several to attend meetings, including an ancillary client group member. The committee structure was fluid and open with representative viewpoints and expertise (e.g. from system design and architecture, software libraries, prototyping, HCI consulting, to the configurator software hotline). Besides the collective and participatory values, members saw themselves as "missionaries" for a user-centered interface, with "responsibility" to improve coding practices, and to work "hands-on" with less-experienced colleagues. In reality, it took time and patience for members to understand the diverse viewpoints and come to trust each other.

During the redesign period, members tried out innovations in their development (not always well accepted by management). Some design issues were put aside as "sacred cows," too politically sensitive to change. With time, the whole development group recognized the interface committee as part of the design acceptance process and as a forum andarbiter. The process encouraged communication, consulting, recognition of expertise, and group coherence, and it led to the formation of other development processes. The committee had to defend its work and process to local and upper management; the members' multiple viewpoints were more persuasive than one person alone. Finally, local management requested committee participation in presentations to upper management.

Expanding Processes: Interface Guidelines with User Participation

Although it had approved formation of the interface committee and the redesign and guidelines goals, management subsequently asked for interface guidelines within 1.5 months of the committee's formation, still believing existing practices to be clear and the task "short term." Collectively, the committee requested to continue work on redesign and do an interim set of guidelines, and management agreed. The initial guidelines were based on an existing, innovative product influenced by field research data, along with some features from other products. (Management chose which product; some felt slighted that "their" software product was not chosen as representative). These first guidelines were an historical artifact within a very short time and scarcely referred to, as the redesign effort evolved rapidly. In retrospect, the interim guidelines were a useful exercise in understanding issues to be addressed and in getting user (developer) participation.

Before drafting the final guidelines, the committee promoted the new design in several ways: Several members (with committee help) made a demo of the new interface look in order to receive acceptance from management and coworkers. After some negotiation, management gave approval. Members also found ways to implement code (not scheduled) for the new standards, even on their own time, so that the new standards would be viable. The committee evolved a common interface "semantics" for communication, so that everyone could understand terms in a similar way.

To promote co-optation and ownership, developers were asked to contribute examples or even to write sections of the guidelines. The HCI consultant was in charge of the guidelines. With the first public draft, developers mostly expressed satisfaction with content and presentation. Some wanted detailed code or in-depth HCI know-how, vs. others who wanted minimal information, believing that "common sense" should prevail. Upper management and a client group had different expectations about content than developers, and their previous approval of the new design (from the demo) turned out to be shaky, causing renegotiation on colors, terms, layout, etc. With the second draft, developers were satisfied with the guidelines' usability; this had been a goal of the interface committee. However, management still believed that developers needed further information which the developers had not expressed they needed (or expressly said they did not need). The committee again met with management to resolve the issues. With the final version, everyone from development to management "owned" the guidelines, a positive ending to the story, despite the difficulties in reaching consensus

Conclusion

User-centered processes developed "bottom up" in the software development group. Even with

a mandate for usability, processes needed to be built to make the products usable. The participatory processes challenged the status quo of traditional software practices, managerial authority, and access to users. Management's approval of the processes (although tentative) was necessary. The professionalism and values of the participants enabled the processes to be built. Their perception of the need to produce user-centered software, bolstered by user data from field research, was a defense against those who tried to "muscle" their own viewpoints (management or group members). End-user participation was a required basis, but not sufficient in order to achieve feedback into design and the user-centered redesign and guidelines efforts. It was necessary to build processes whereby designers and developers were involved, leading to their buy-in and ownership.

A Consultant's View of the Process of Introducing Usability Engineering into Development Organizations

During my 11 years experience in the field of software usability, including 6 years as a consultant, I have conducted over 60 projects in over 40 different companies, including both hardware and software vendors and companies in other industries with internal development organizations. On the basis of this experienced, I have devised an idealized methodology for integrating usability engineering into the development process. I have helped and watched many development organizations—each with their own unique development process—take initial steps to move from their existing development process towards this integrated methodology. This section describes what I have experienced to be the main motivations for change, and the key factors in success or failure of initial attempts to incorporate usability methods into development organizations.

About the Authors:
DEBORAH J. MAYHEW *is principle and owner of Deborah J. Mayhew & Associates, a small consulting firm offering a variety of services related to computer-human interaction. Prior to establishing her company she held a faculty position at Northeastern University in Boston and has also worked for several hardware and software vendors' companies as a usability engineer.*

Motivations for Change

A variety of things seem to inspire a development organization's first attempts to integrate usability methods into their overall development process. Sometimes a single individual plays the role of change agent. This individual may be at any management level, from a project leader who decides to hire a usability expert onto their project team, to an R&D Vice President who decides to make usability a part of their organizational territory. In this case, it is the vision of a single, powerful individual that drives change.

In other cases, a clear high visibility disaster is the change agent that motivates an organization. Perhaps a high profile, very expensive development effort fails dramatically, and users clearly state that they reject a system because it is unusable. Or, a product fails in the market place and customers point to its user interface as the reason for their rejection.

Sometimes, in the case of vendor companies, the market place clearly provides motivation for change. Marketers hear a clear request from customers for improved usability, and bring pressure to bear on development organizations, or they perceive that they are losing market share and attribute this to competitive companies doing a better job of usability.

In the case of internal development organizations building systems for in-house users, a general motivation to address typical business problems such as low user productivity, high user training costs, and the desire to grow without increasing costs may become focused on usability issues.

Sometimes the motivation to first bring in a usability engineer, often on a consulting basis, arises out of a need for an objective, neutral means for resolving internal conflicts over design issues. Opposing parties agree to resolve a design issue through objective usability testing, but do not have the skills in-house, or can only agree to abide by the findings of an outsider.

Finally, sometimes education, in the form of short presentations to management, or seminars for developers, can begin the process of change.

Factors in Success or Failure

In my experience, unless one of the motivators listed above is present, trying to influence an organization to change to integrate usability methods into its development process is a thankless task. But when one of these motivators is present, an opportunity arises for professional usability engineers, either internal or external, to have an impact. Then there seem to be a set of factors which influence the success or failure of these individuals in their role as change agents. Keys to success include:

Communicate effectively

Prototype a design standard, rather than just document it. Write designer/developer-friendly design documents.

Get "buy-in"

Get others involved in design and testing. Cast yourself as a team member, not a critic, etc.

Be an engineer, not an artist

Make it clear you are applying systematic methodologies, data and principles, rather than just offering one more opinion.

Produce well-defined work products

Don't allow yourself to be cast as one more opinion in the design meeting. Define short information gathering, design, and testing projects with clear scope, schedule, deliverables, and impact.

Manage expectations

Make the limits of prototype testing clear. Make the limits of design principles and guidelines clear. Don't be afraid to state what you don't know, haven't done, and aren't good at.

Clarify value added

Always cost-justify your plans and efforts. Choose initial projects that will quickly and dramatically demonstrate value, such as testing vs. style guide development.

Test

Do this whenever possible, as data are better than expert opinion.

In contrast, the factors which seem to lead to failure include poor communication, failure to work at getting "buy-in", failure to find ways to establish credibility, failure to define clear work products by which the professional and the field can be evaluated, failure to manage unrealistic expectations, failure to clearly demonstrate value, and failure to provide objective data.

The usability engineer who wishes to succeed in the role of change agent needs to first ask him/herself if the proper motivators are present in the organization. If none are present, the creative and ambitious professional can create one, either by educating appropriately powerful managers, or by making problems visible and clarifying value added. For example, a usability test on a high visibility project which clearly demonstrates dramatic usability problems can be such a powerful motivator. Until such a motivating force is present, most efforts, no matter how professional, will fail.

Once the motivator is present, then the usability specialist must take care to operate strategically to achieve the success factors listed above. It is entirely possible to fail even in a receptive environment. Being a change agent requires political skills as well as technical skills, and choosing and conducting one's projects strategically is the key to moving a development organization in the direction of usability engineering, and to establishing integrated roles for usability engineering and engineers.

References

[1] Kim, S. (1990). Interdisciplinary Collaboration. In B. Laurel (Ed.), The Art of Human-Computer Interface Design. Reading, MA: Addison-Wesley Publishing, Co., pp. 31-44.

[2] Cypher, A. (1991). Eager: Programming Repetitive Tasks By Example. In S. P. Robertson, G. M. Olson, and J. S. Olson (Eds.), Reaching Through Technology: CHI'91 Conference Proceedings. New York: ACM Press, pp. 33-39. See pp. 445-446 for a video presentation description of Eager.

[3] For a detailed description, see K. H. Kvavik, D. Fafchamps, S. Jones, and S. Karimi, Field Research in Product Development, SIGCHI Bulletin, 24, 2, 2-27.