by Barry Raybould

Performance-Centered Design

COMPUTERS. They mediate our work more each day, especially in the area of customer service. To stay competitive, companies have had to move decision making and the access to information much closer to employees who deal directly with customers. It follows that more computer usage requires more software development.

Much of this development involves "Performance-Centered Design," which can help improve employees' work performance and reduce training time. For example, software designed using the PCD approach can lay out the key steps in a specific work process and embed tips on making decisions (called knowledge support) at the appropriate steps. PCD's goal is to generate "day one" performance, meaning that a new employee attains an acceptable level of performance within hours rather than weeks or months.

In the computer-software market, effective PCD is critical for garnering positive reviews in PC magazines. Such reviews can be worth an additional $10 million in sales. An example is Quicken, a personal-finances software application, for which the developer recently received a $2.1 billion bid from Microsoft.

Another reason for the rising interest in PCD is that business consumers are beginning to expect the same quality of design that they get in the PC software they use at home. Companies such as American Express Financial Advisors have shown that dramatic improvements in workplace performance can be achieved by using the PCD approach—for example, an 85 percent reduction in training time and a 90 percent reduction in errors.

In another example involving a hotel chain, a traditional software application required two weeks of training to learn 80 percent of front-desk procedures. But a performance-centered interface reduced training time by 93 percent and guest check-out time from 40 to 80 percent.

So, what are the pitfalls? One, if the organization isn't performance-oriented, department goals may conflict with a performance-centered-design approach. For example, design work may be seen as a threat to software-delivery dates. Instead, a project's target dates should be expressed as "when performance is achieved by employees" and not "when the software arrives." In fact, the lifecycle of a performance-centered project can be shorter than the typical project lifecycle. One reason is that you don't have to wait for lengthy training to be completed in order to attain effective performance on the job.

Performance-centered design can boost employees' work performance through on-the-job training—as long as PCD isn't an afterthought to the software development

To cut project-cycle time, it's recommended to create a business-performance model that depicts graphically how employees achieve effective performance in their work—and how that relates to work processes and organizational goals. A typical BPM identifies potential roadblocks to effective performance, such as conflicting procedures and processes, as well as inadequate knowledge support for decision making that should be embedded in the software.

Highlighting potential roadblocks to meeting business goals in a project makes it easier to get top management involved—and to refocus the project on performance rather than on meeting software-delivery dates.

Of course, just making the software easier to use doesn't guarantee day-one performance. Knowing how to use the application's menus isn't the same as knowing how to accomplish a work task, such as taking a customer's order on the telephone. A help system and tutorial—no matter how excellent—aren't sufficient. Microsoft has recently recognized that when designing its software.

For example, Microsoft Publisher has incorporated a performance-support capability called "Wizards" that guides users through the process of creating a brochure, for example, instead of just showing how to use the commands. And Microsoft's Powerpoint helps users structure key building blocks in a sales presentation, not just how to insert a graphic on the slide.

One can't achieve the desired results just by waiting for the software to be developed and adding performance support as an afterthought. PCD is software-interface design, not the design of separate components bolted onto a computer system. It's necessary to address the interface design at the beginning of a project. Most HR practitioners and trainers already possess the appropriate skills, such as needs analysis and knowledge presentation. With some instruction in interface design and PCD, they can become valuable members of a software-development team.

Overall, for a successful interface design it's crucial to work with senior management to promote a performance-oriented view within the organization, to communicate the difference between day-one performance and ease-of-use, and to ensure that people with expertise in performance-centered design get involved with the system design from the start of any project. ■

Barry Raybould *is president of Ariel Performance Centered Systems, 5215 N. O'Conner, Suite 760, Irving, TX 75039. Phone: 214/869-8113.*