Design User-Centered Web Applications

Getting users involved means creating effective applications that help people perform tasks over the Web.

By Harris Kravatz, Christian Pantel, and Tara Ehrcke

E-BUSINESS BENEFITS

- A user-centered application design methodology promotes iterative and incremental rapid prototyping, with short feedback loops, to help companies deploy higher-quality Web applications. This also helps companies avoid snags late in the development stage, which can translate into higher costs, missed deadlines, and other development problems.
- Getting users involved in the application design helps manage customer expectations and ensure that the Web application meets users' needs.
- A user-centered design methodology involves both careful preparation and iterative design before starting development. It helps the Web design team efficiently communicate application requirements, standards, and guidelines to the application developers.

Although Web site design has become a common occurrence, creating Web applications is a relatively new phenomenon. While Web sites tend to attract users who seek information, Web applications attract users who want to perform tasks such as paying bills, making travel reservations, buying and selling stocks, tracking parcels, and taking courses.

Web applications can be distinguished from Web sites by their level of interaction. Users interact with Web sites primarily by selecting links and viewing information—an experience similar to changing channels and viewing programs on television.

In contrast, Web application interaction is give and take—action and reaction. Users enter information and select choices. The application processes the information and provides immediate feedback. For every action, users expect a reaction. They expect the application to guide them in completing their tasks, and to provide a status of their tasks. When they've accomplished their goal, users expect to receive confirmation.

To design this level of interaction requires a team skilled in interaction design and a methodology that focuses on the user.

User-centered design methodology

A user-centered design methodology focuses on user's needs, includes them in the design effort, and relies on them for design feedback. The iterative and incremental methodology is characterized by rapid prototyping and short feedback loops.

A user-centered design methodology has two main phases: the preparation phase and the design phase (figure 1). The preparation phase occurs only once, but the design phase is iterative. Prior to beginning the preparation phase, you need to create a user-centered design team. Typically, this team consists of interaction designers, graphic designers, information developers, usability testers, and developers.

Preparing a user-centered design

The goal of the preparation phase is to establish a solid foundation on which a Web application can be built. Key activities in this phase include understanding the domain and users, creating the conceptual design, and documenting the design standards and guidelines. Not adequately preparing for design, such as entering the design phase

Harris Kravatz is principal of HK Interface Design, a consulting and training firm in Boca Raton, Florida. He has been mentoring and teaching the user-centered design process described in this article since 1994. kravatz@ibm.net.

Christian Pantel is a user experience architect with PeopleSoft in Pleasanton, California. He was the user-centered design team lead for the university project referred to in this article. christian_pantel@peoplesoft.com.

Tara Ehrcke is a user interface designer at IBM's Pacific Development Center in Vancouver, British Columbia. She was instrumental in creating the conceptual design and the design standards for the university project. tara@ca.ibm.com.

Phase 1: Preparation
Step 1: Understand domain and requirements
Step 2: Get to know the users
Step 3: Identify design constraints
Step 4: Develop conceptual design
Step 5: Establish standards and guidelines
Step 6: Select technologies
Phase 2: Design (Iterative)
Step 1: Determine scope of increment
Step 2: Produce initial prototype
Step 3: Conduct collaborative design session
Step 4: Produce detailed prototype
Step 5: Evaluate and update prototype
Step 6: Review with customer
Step 7: Communicate specification to developers
Step 8: Review design process

Figure 1: **User-center design methodology**—Focuses on user's needs, includes them in the design effort, and relies on them for design feedback.

without a firm navigational scheme, can lead to expensive rework during the design phase. Here are steps of the methodolgy you should follow:

Step 1: Understand the domain and requirements

The user-centered design team must become familiar with the application domain for the Web application they'll design, and must clearly understand the customer requirements and for what portion of those requirements their team is responsible. This is also important in managing customer expectations. The team reads the customer's requirements document and interviews subject matter experts.

Step 2: Get to know the users

Early in the process of designing a usable application, the design team must know who will use the applications. It's useful to determine the characteristics of the user population and identify whether the users can be categorized by the roles they'll play when using the application. User surveys are a good tool for learning about users.

Step 3: Identify design constraints

Before starting to design a Web application, the design team must understand the design constraints under which it's operating. Some of the questions to address include:

- How much time do we have?
- Do we have access to subject matter experts and users?
- Which Web browsers must be supported?
- Which platforms must be supported?
- What's the minimum network connection speed? Do we need to support text-only browsers or other client interfaces, such as handheld devices? Do we need to provide special access for the visually impaired?
- What screen resolution must be supported?
- Do the external systems impose any interaction constraints? For example, do we have a live connection to the external systems or can we only run batch jobs?

Step 4: Develop a conceptual design

The conceptual design of the application specifies the tasks and information that users access, and the mechanisms for navigation. It's based on an understanding of the users, the goals of the application, and the design constraints. A major part of creating the conceptual design is creating a design prototype that focuses on the breadth of function first, then depth.

When focusing on breadth, the goal is to create a consistent method of access to all tasks and information. Focusing on depth specifies the design of one or two key user tasks from start to finish, and helps identify potential design problems early. The team then obtains feedback on the conceptual design from users and the customer, and iterates on the design as necessary. Feedback is obtained through informal walk-throughs and usability testing.

Step 5: Establish standards and guidelines

The team writes user interface standards and guidelines based on the results of the conceptual design activities. An important purpose of standards and guidelines is to document design decisions to help ensure consistency and good design when more than one person will be prototyping during the design phase. These standards and guidelines should be as complete as possible, and should include use of fonts and colors, navigation mechanisms such as tabs, buttons and links, page heading formats, and table specifications. There's opportunity to update the standards and guidelines during each iteration of the design phase, but the cost of making changes increases as time passes because more and more re-work may be required to retrofit earlier prototypes.

Another benefit of writing standards is that it forces designers to consider design issues early in the development cycle and to articulate design decisions. The standards and guidelines are also a useful tool in helping new team members quickly become productive.

Step 6: Select technologies

This step is where the team selects technologies for prototyping. Choices include HTML, Cascading Style Sheets (CSS), JavaScript, XML, and Java. Whenever possible, the prototyping environment should be the same as the project's development environment. This lets the development team reuse the prototype during the development phase, and provides a more realistic look and feel when conducting design reviews and usability testing. The use of style sheets is recommended if your target browsers can support them. They're helpful in providing consistency across prototype functions and facilitating design changes—significant benefits for a Web application with hundreds of pages.

Designing a user-centered Web application

The goal of the design phase is to fully specify the behavior and appearance of the Web application. This phase is both iterative and incremental. It's iterative because the steps in the phase are repeated, and incremental because each repetition of the steps specifies an additional component of the Web application. An advantage of an incremental approach is that developers can start coding as soon as the first iteration of the design phase has been completed. Here are the steps for designing the application:

Step 1: Determine the scope of the increment

The first activity in the design phase is to determine the scope of the increment. Many factors may be involved in

determining the scope of each increment, including the customers' priorities and other project dependencies. Project management must be involved in this determination, and should formally accept the scope of each increment.

Step 2: Produce the initial prototype

Based on the customer's requirements and the agreed-upon conceptual design, and by applying the established user interface standards and guidelines, the design team produces an initial prototype of the use cases contained in the current increment. This prototype is reviewed internally, and serves as an input to Step 3.

Step 3: Conduct collaborative design sessions

An important goal of a collaborative design session is to clarify requirements and explore high-level design alternatives. These sessions can also provide the design team with valuable information about the customer and users' environments. The team holds a collaborative design session for each set of closely related use cases in the current increment.

An effective participant mix includes a few users, a few subject matter experts, and a customer representative empowered to make decisions. The users' contributions are important, since their needs are likely to be representative of other users. Subject matter experts provide valuable input because they clearly understand the business processes being designed in the Web application. It's also important that a decision-maker participate, in case there is disagreement on the interpretation of certain requirements.

The initial prototype should be used as a starting point. Avoid getting bogged down in discussions about design details. It's much more important, at this stage, to clearly articulate the required functionality and to explore screen flows rather than endlessly discuss the layout of individual screens. Facilitate a quality meeting to ensure that these sessions are as productive as possible.

Step 4: Produce a detailed prototype

Based on the results of the collaborative design sessions, the design team can produce a detailed prototype. Time permitting, it's most effective to fully prototype the use cases and not leave design decisions open to resolve during development.

To manage customer expectations, it's important that the prototype be technically feasible. Ensuring this may require a review with a project architect or with developers.

Step 5: Evaluate and update the prototype

The detailed prototype is brought back to the group involved in the collaborative design sessions. During this second session, the participants have the chance to verify that their requirements were clearly understood by the design team. Participants also have a chance to critique the prototype and make suggestions for improvements.

In parallel with this review, the detailed prototype should be usability tested with representatives from the user population.

Based on feedback from the review and the usability testing, the prototype is updated and re-tested to ensure that most of the usability problems were identified and fixed.

Two iterations of usability testing with six to eight users is typically sufficient. Most of the design problems can be identified during these first two iterations.

Step 6: Review with the customer

Obviously, the final prototype should be reviewed with the customer to give them a chance to see the progress and to approve the design before it's formally implemented.

Step 7: Communicate the specification to developers

A critical part of the use case design phase is to effectively communicate the specification to the developers. The specification should consist of two parts: a written document for each use case and the final prototype.

This documentation includes:

- A description of the use case and its important scenarios,
- A chart depicting the navigational flow of Web pages, and
- Details about every Web page, including where to obtain data elements from the external systems.

Step 8: Review the design process

If you're implementing this process for the first time, it's a good idea at this point to conduct a brief internal process review. This review can identify any efficiencies that could be gained in the implementation of each of the steps of the design phase. As part of the process review, updates to the standards and guidelines can also be considered, discussed, and agreed upon.

Challenges of Web application design

Designing for the Web has a unique set of challenges. Much of the available technology for designing Web applications is immature and changes at an unprecedented rate. Designers must cope with limited interaction techniques; browser incompatibility across browser vendors, versions, and platforms; inadequate performance due to network constraints; and inadequate development tools.

In addition, Web users can make designing Web applications especially challenging, because many of them are inexperienced, or have only limited experience with computers and Web browsers. Often, little training is provided because the Web application is accessed by a broad audience of geographically distant users. Also, Web application users may typically be casual users who don't spend the majority of their time using a single Web application. For example, a typical user may make travel reservations once a month, or use online banking every other week.

A Web application must therefore be easy to use and not rely on a user learning how to use it. Very few users will ever become expert users of the application.

As of yet, there are also no widely accepted user interface (UI) standards for Web applications. UI designers are challenged to make fundamental decisions regarding the UI, relying only on a limited set of tools and interaction techniques.

Meanwhile, customers familiar with applications on traditional platforms may have unrealistic expectations because they don't understand how the current state of Web technology may limit the level of interaction that can be provided by today's Web applications.

Applying the user-centered design methodology

The user-centered design methodology presented here was successfully applied in the design of a large-scale university Web

application that lets students register for classes online, thereby avoiding lengthy lines and trips all over campus. However, this methodology wasn't initially chosen for the project.

The application design was initially based on a modified object-oriented analysis and design methodology which de-emphasized collaborative design and iterative prototyping. Instead, detailed use cases were created from a requirements document written by the customer. Each use case described a task, identified the actors (users) who would perform the task, stated the pre- and post-conditions of the task, and identified any external systems with which the application would interact. They also outlined successful and exceptional task scenarios.

Decision Point Pages: A New Way to Communicate

To help designers better communicate the intended behavior of the application to developers, we developed a Web prototyping technique called decision point pages.

Decision point pages are Web pages inserted into a prototype to exercise an application's various branching points. To visually distinguish decision points from pages that are part of the delivered application, we used a different color background and different color titles. The decision point pages also contained a message stating that the pages weren't intended to be part of the application, but were available to describe a situation and list possible alternatives. The description was hyperlinked to an application page, or to another decision point page. In this way, we were able to simulate actions that would be performed by the application itself, such as checking that a set of courses had no time conflicts. It also let us prototype error handling.

Figure 2 shows a sample decision point page. Clicking on the "Add a class" button links to this decision point page. The decision point tells developers that code is required to determine whether the user is allowed to enroll. Each alternative has a hyperlink to the page that handles that alternative.

Prototyping with decision point pages can be more efficient than writing scripts. Linking a page to a decision point that describes the intended processing is much faster than writing a script that simulates the processing. Furthermore, standard Web authoring tools can manage the links of a prototype if the prototype uses decision point pages, but they can't when scripting is used. This makes prototypes with decision point pages much easier to maintain.

Figure 2: **Sample Decision Point page**—These Web pages are inserted into a prototype and can exercise an application's various branching points.

Decision Point — Enrollment appointment check
Does the student have a valid enrollment appointment or is it open enrollment time?

- Yes, continue to *Add* a class page
- No, display the page that informs students of the days and times they can enroll.

This page will not be part of the product; it's here to exercise the prototype.

From the use case document, class diagrams, object interaction diagrams, and state transition diagrams were created. The use cases and diagrams were then presented to domain experts and users to validate the work completed and to obtain feedback.

Presenting those technical documents and diagrams to people unfamiliar with them proved problematic. Since the concepts were difficult to understand, the domain experts and users became frustrated. They couldn't envision the user interface from the diagrams, and were only able to provide limited feedback. This frustration spread to the customer executives who began to doubt whether requirements were being captured and whether progress was being made.

To address the customer's concerns, a user-centered design team was created and charged with implementing the user-centered design methodology presented here. Collaborative design sessions and prototype presentations replaced use case reviews and technical diagrams. The object-oriented analysis and design methodology continued to be used successfully in developing the application, but it was done in the background and not presented to the users.

This shift to a user-centered design methodology alleviated the customer's concerns and helped to design a higher quality product. Involving users, subject matter experts, and representatives from the customer in the design and evaluation of the application provided valuable feedback. Rapid prototyping allowed for quick evaluation of design alternatives, letting customers easily see progress through the delivery and refinement of prototypes. This also gave the customer more confidence that their requirements were understood and a high quality solution could be delivered. ADVISOR