Designing User-Centered Web Applications in Web Time

Molly Hammar Cloyd, *Broadbase Software*

As designers struggle to develop Web applications "in Web time," they are under the added pressure of delivering usability. This author describes her company's successful transformation to user–driven processes for designing e–commerce applications. She also offers strategies for introducing human factors methods into a reluctant development organization.

U sability has moved from a "nice to have" to a "must have" component of e-commerce application design.[1] In the past, customers purchased desktop applications and then struggled to learn how to use them or called for technical support. Now, they shop in a try-before-you-buy model. If they can't navigate your site, they are a few clicks away from your competition. Even if you're in an industry with few competitors, users' time and attention are at a premium.[2]

The growth of e-commerce and business-to-business applications has created an unprecedented emphasis on knowing our users and designing usable applications. However, backing corporate commitments to usability with user-driven development processes is a challenge. Designers struggle to design new applications, defining Web user interface standards as they go, all the while under pressure to deliver applications faster—in "Web time." These problems are compounded in many start-ups, which have little design process infrastructure, much less human factors methodology, in place.

With little historical data about Web application user interface and usability standards, human factors engineers are searching for ways to balance three different approaches to Web-based usability engineering: transferring traditional application design techniques to the Web environment, relying on emerging Web design standards, and conducting new research into what

Web application users want and need. Developers frequently ask usability professionals, "What's the difference between a Web site and a Web application?" "Should I conform to Web site standards or Windows standards when designing Web application screens?" and "What should the Cancel button on a Web form do?"

To complicate matters, developers focusing on getting Web applications to market in Web time often means they cut back on planning and design in the development process. Overall, the use of software engineering processes is in decline.[3] The result: human factors engineers are pressured to provide unprecedented usability in a fraction of the time they need.

This article presents a case study of how Decisionism, an analytic-applications company, redefined its software development process to design usable Web applications in Web time. In the midst of these process changes, Broadbase Software acquired the company. The development process that

Decisionism pioneered is now the basis for the user-centered design group at Broadbase Software.

Our challenge

Our organization's decision to enter the B2B Web application arena shifted us from being a traditional software developer to a Web application provider. Specifically, we were faced with these challenges:

- Shifting the development organization's mindset from a feature-driven approach to a user-goal-driven one.[4] Rather than generate lists of product features (what our product would do), we wanted to set requirements based on what users would be doing with our product.
- Changing the organization's view to human factors methods. Prior to entering the Web application arena, Decisionism did not have a human factors group, so its addition represented a change in the corporate culture.
- Introducing a design process in an organization in which team members were reluctant to be bound by procedures or heavy project documentation requirements. Our challenge was to design a process comprehensive enough to be repeatable and to support introducing new team members and technologies, without being cumbersome.
- Defining an all-new product, starting with very little knowledge about potential users and no concrete information about how users would perform tasks with the new application.
- Having a limited design, development, and quality assurance staff along with a corporate goal to be first to market with a B2B analytic application.
- Designing a development process that would allow for the thorough investigation of users' characteristics and goals yet would facilitate a rapid application development life cycle.
- Making an architectural shift from a user interface that is tightly bound with functional components to a flexible one that could be changed with minimal impact to the underlying code.

Our Web application design process

Decisionism redefined its development process by placing human factors methods at the core. We stripped away our existing software development process and started over. In a matter of days, we outlined the human factors methods and deliverables that would be required to

- determine who our new application's users would be, what their goals are, and how they work;
- establish overall Web application user interface standards;
- identify usability goals for the new application;
- communicate usability architecture requirements to the developers;
- determine the application's overall flow;
- design the user interface, including site maps, prototypes, and usability tests; and
- produce a user interface specification that supports developers in programming the interface but does not take months to write.

With this process outlined, we asked the development, quality assurance, documentation, and marketing leads to add their pieces to the process. We sequentially added each functional group's tasks until all the people on the team were satisfied that the process met their needs. The resulting development process is centered on human factors methods. Every human factors deliverable is a critical input to other functional teams' work.

We defined five phases for the requirements and design process:

1. Condensed user and user-goal analysis.
2. Proof of concept (prototyping).
3. Combined site maps and storyboard content.
4. Use cases with screen mockups.
5. Hand-off of use cases and screen mockups to development.

Condensed user and user-goal analysis

Prior to the design phase, the business development group completed a market analysis of prospective customer companies. Using this information, we spent one week sketching out a preliminary picture of our prospective users, identifying such factors as their goals, skill level, and measures of job success. Everyone on the team—the lead architect, programmers, services members, the quality

> **Decisionism redefined its development process by placing human factors methods at the core. We stripped away our existing software development process and started over.**
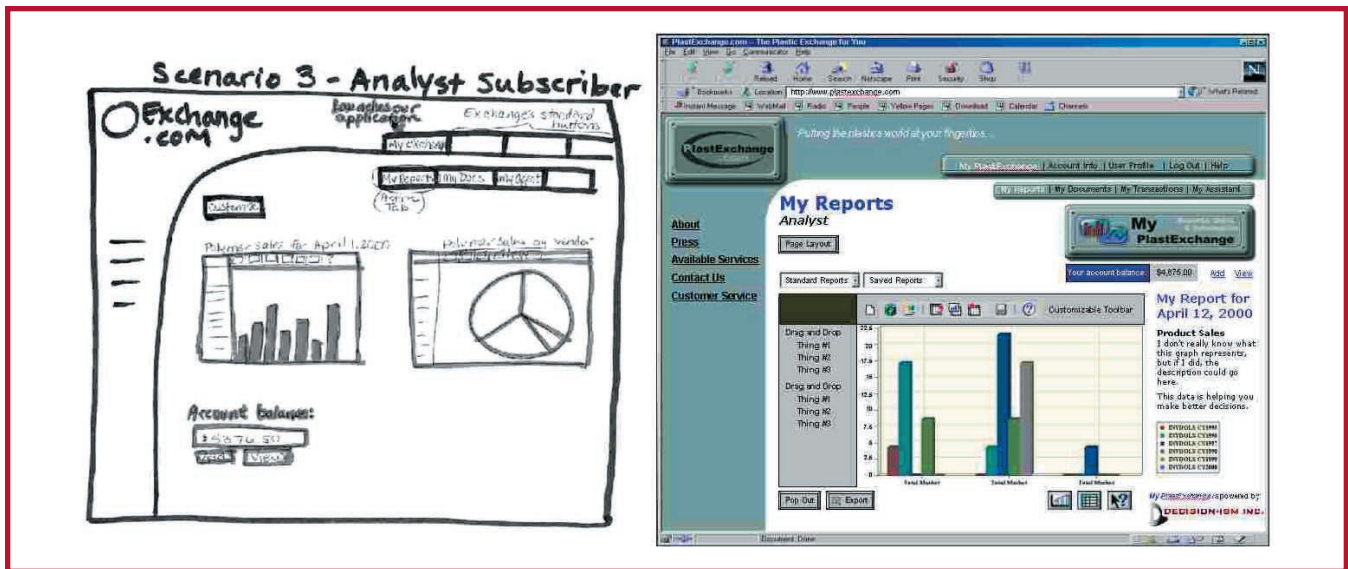
Figure 1. We used a paper prototype to convey our initial product vision to team members and from this developed a PowerPoint demo screen to test with users.

assurance and documentation lead, and the vice presidents of engineering and business development—participated in creating user profiles. This initial look at users gave the team a starting point for identifying the tasks that users would perform with the application and for creating a prototype to use in subsequent user analysis and feedback sessions.

In *Mastering the Requirements Process*, James and Suzanne Robertson describe the requirements process as determining "the business problem to be solved ... and what the product will do to contribute to a solution."[5] Unfortunately, in many companies, this process is abbreviated because of tight deadlines. The functional requirements document becomes merely a shopping list of features that engineers prioritize and identify trade-offs to determine which features can be implemented in a given release.

In taking a user-centered approach, we steered away from feature lists and focused on a handful of real-life user problems or goals that our product would accomplish. For example, instead of listing requirements such as "Display of multiple analytics on a single page," one of our requirements was "Enable users to determine the best auction starting price for a commodity." This requirement led to an Offer Optimizer software module that not only displayed multiple analytics on one page but also supported users in making smart buying and selling decisions in a B2B market.

With our preliminary user profiles and user goals in place, we started an ongoing process of meeting with potential customers, watching them work, and asking for their feedback on user interface prototypes. In addition to soliciting feedback from a variety of

B2B companies, we created a close development partnership with 20tons.com, a marketplace information provider for the plastics industry. They acted as subject matter experts and provided us with ongoing feedback and input into user profiles and use cases throughout our design and development processes.
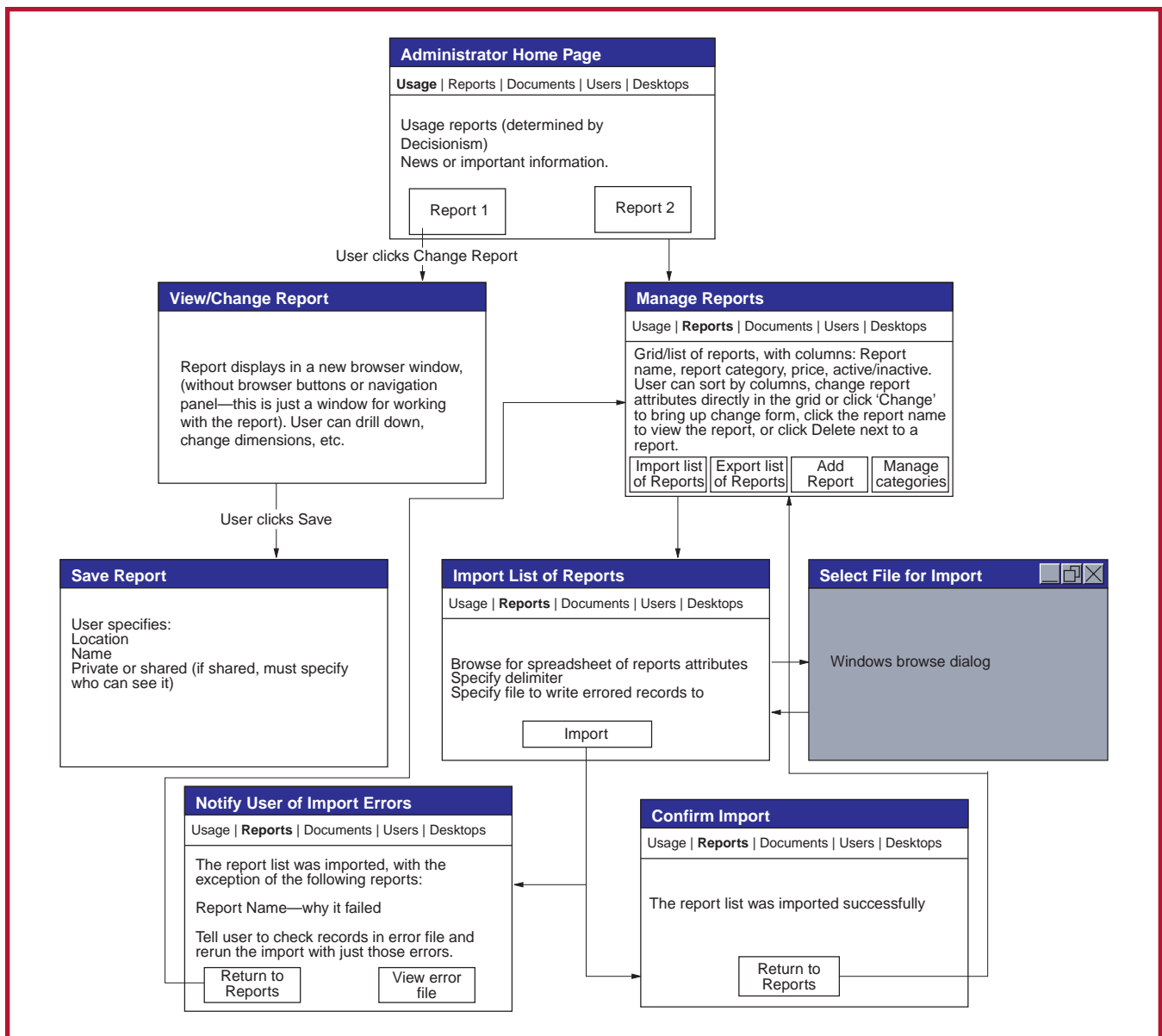
Proof of concept

Because we were introducing a brand-new product idea and starting with so little user information, we created a paper prototype to convey our initial product vision to team members. This served as a starting point for gathering requirements and usability feedback from prospective users.

We chose paper prototyping rather than functional prototyping for three reasons:

- It was faster to mock up and revise designs than coding screens.
- The designs clearly had not yet been coded, so reviewers did not hesitate to suggest changes.
- Developers were not tempted to use already written code.

Once we were satisfied with our initial paper prototypes, we created PowerPoint slides of the proposed user interface (see Figure 1). We used these to gather feedback about our overall product requirements and interface design approach. The PowerPoint prototype conveyed our overall vision for the product yet was general enough to spur design conversations with users.

Because we wanted to gather requirements as well as usability feedback, we used cognitive walkthrough to evaluate our prototype design. In a cognitive walk-

Administrator Home Page

Usage | Reports | Documents | Users | Desktops

Usage reports (determined by Decisionism)
News or important information.

Report 1   Report 2

User clicks Change Report

View/Change Report

Report displays in a new browser window, (without browser buttons or navigation panel—this is just a window for working with the report). User can drill down, change dimensions, etc.

User clicks Save

Manage Reports

Usage | **Reports** | Documents | Users | Desktops

Grid/list of reports, with columns: Report name, report category, price, active/inactive. User can sort by columns, change report attributes directly in the grid or click 'Change' to bring up change form, click the report name to view the report, or click Delete next to a report.

Import list of Reports | Export list of Reports | Add Report | Manage categories

Save Report

User specifies:
Location
Name
Private or shared (if shared, must specify who can see it)

Import List of Reports

Usage | **Reports** | Documents | Users | Desktops

Browse for spreadsheet of reports attributes
Specify delimiter
Specify file to write errored records to

Import

Select File for Import   ▢▢✕

Windows browse dialog

Notify User of Import Errors

Usage | **Reports** | Documents | Users | Desktops

The report list was imported, with the exception of the following reports:

Report Name—why it failed

Tell user to check records in error file and rerun the import with just those errors.

Return to Reports | View error file

Confirm Import

Usage | **Reports** | Documents | Users | Desktops

The report list was imported successfully

Return to Reports

through, prospective users tell the facilitator what their goals would be for using the product, and then they guess where each navigation path will take them and explain how they would expect to perform certain tasks using the prototype design.[6] These methods expose the users' goals and expectations and identify potential navigation pitfalls in user interface designs. Cognitive walkthrough proved to be a valuable technique for gathering usability data on prototypes that were not fully functional. We used it to evaluate our prototype with five users at two net market companies. Ideally, we would have gathered feedback from a larger sample of users. However, with our time constraints and the difficulties we had in finding users, we collected as much information as we could before moving on to the next phase.

Combined site maps and storyboard content

Armed with a better understanding of our users, we were ready to build a site map, an aerial view of the application showing how the user interface screens would flow from a user's perspective. To save time and make the site map easier for reviewers to conceptualize, we built storyboarding components directly into our site map. Whereas many site maps only contain representations of each screen and the navigation between screens, our site map included lists of each screen's content. By presenting user goals, navigation, and screen content in the context of the overall application flow, the site map was the converging point for user-driven and technical product requirements (see Figure 2).

We conducted a series of intensive review sessions to get input and approval from every

Figure 2. Combined site map and storyboards made it easier for reviewers to conceptualize.

member of the development, marketing, services, and executive teams. We also gathered feedback from our 20tons.com development partner. This feedback and approval process was critical to our ability to develop the application quickly. As a group, we walked through every screen of the application, considering the task flow and functional requirements from a user's perspective. The process, though tedious, ensured that everyone involved in the application's design, development, and marketing was in full agreement about its scope and flow. This process also identified and forced us to resolve contradictory visions of the application scope or flow early in the design process.[7]

After the development, marketing, and business development teams signed off on the prototype, we created detailed designs for each screen and included them in a modified use case document. Together, the site map and the modified use case document took the place of the traditional user interface specification document.

Use cases with screen mock-ups

We expanded each user goal identified in the condensed user and user-goal analysis phase to include use case information. The lead architect, lead developer, and human factors engineer jointly contributed to use case documents. With slight modifications of the Rational Unified Process use case template,[8] our use cases embodied users' goals and motivations and functioned as developers' guidelines for implementation.

Our use case document was organized by user task (for example, "Viewing a Report"). For each process, the document provided details about the look and feel, task flow, and technical requirements for implementing the use case in the application. For each user goal or task, the document included the following:

- the users' goals and, if applicable, how users would know when they met each goal;
- frequency and criticality of tasks;
- usability requirements of the user interface supporting each use case;
- a picture of the screen (this was a placeholder section in early versions of the document, later filled in with a design diagram);
- a list of data elements (such as buttons,

links, or display-only items) and how they would respond to users' actions;
- descriptions of how interactions with the data elements would be validated;
- requirements for entering and exiting each screen; and
- requirements for future releases that might affect how a use case is implemented.

Figure 3 shows an example use case and screen mock-up that we developed using the modified use case template.

Critics of use cases argue that it is a time-consuming, arduous task that can delay implementation. Others argue that there is no way of knowing when the set of use cases is complete.[9] However, our team subscribed to the view that in rapid development environments, designers should select a small number of users and use cases that represent the entire product and then develop a user interface architecture that can extend to the whole product.[10] We questioned the cost-effectiveness of creating an exhaustive set of use cases with such limited time. Our aim was to identify the users' most important goals and then develop an application that would enable users to meet those goals, meet the product requirements, and be extensible to outlying goals and tasks.

We generated use cases for each product requirement. We focused on the activities users would perform most frequently with the application and activities most critical to the users' success with the product. These use cases gave us the framework we needed to develop the application's core functionality.

Combining use cases, screen mock-ups, and screen descriptions into a single document saved time and also ensured that use cases, user interface designs, and functional requirements were kept in sync. Jointly developing use cases put human factors' influence into a context that was already familiar to developers. Using the modified use case template ensured that user goals were viewed as integral to every use case. This meant that technical requirements in use cases were driven by the flow of events from a user's perspective. Also, efficiencies were gained by creating a use case model that incorporated both developers' and users' needs instead of embarking on separate activities to define human factors requirements and development requirements.

Hand-off of use cases and screen mock-ups to development

The design and development process was iterative. Once we identified the big picture of the application flow in the site map, we created use case documents for specific areas of functionality. When a use case or group of use cases was complete, we handed those off to engineers for development. For each set of use cases, the director of development produced an architecture design for that iteration of the product.

As engineers coded one set of use cases, the design team created the next set of use cases. If engineers encountered implementation issues that required user interface changes, we responded by quickly mocking up alternative screen designs. Having the overarching site map in place made it possible for us to hand off sections of the user interface to be coded without losing continuity across incremental designs.

The entire product design process, from user analysis to hand-off of the design to developers for coding, took about 12 weeks.

Implementing human factors processes in a reluctant organization

Several members of our development team were reluctant to adopt development processes, let alone one grounded in human factors methods. Some had come from large companies where they'd had negative experiences with ISO or slow-moving waterfall processes. Others were concerned that the human factors engineer would design the application in isolation and hand down designs that the developers would have no control over. Most were concerned that following a process would prevent us from meeting our time-to-market goal. With these concerns in mind, we worked to create a process that would help, rather than hinder, developers. Strategies included the following:

■ Combining phases of traditional human factors processes to ensure that our process required minimal documentation and was not cumbersome.

Figure 3. An enhanced use case and screen mock-up.

Use case #3: View a report

Goals

Users will view reports to answer specific business questions. Their goals are to find the information they need and use the information to make buying or selling decisions on the B2B exchange.

Flow of events

All steps are required unless noted.
1. The user views a list of available report portfolios.
2. The user clicks a portfolio name.
3. The system displays the thumbnail view for the selected portfolio.
4. The user clicks a report.
5. The system displays the report.
6. (Optional) The user modifies the report (see Use case #4: Modify a report)

Usability requirements

The typical user will have little or no experience with OLAP analytical tools. This interface must support the learning curve of inexperienced users, and must also convey the full spectrum of analysis functionality at is available.

- Users should feel comfortable/safe exploring the interface and trying new analysis techniques.
- Users should be able to find the report they want to view in less than 1 minute.
- Users should be able to pop a report out to full screen view, manipulate it, and save it in under 4 minutes.

Frequency/criticality

Users will view reports every time they open the application. Users will conduct analysis in a stream-of-consciousness manner. This means they might view one report, which will trigger the need for a second report, and so on. Report accuracy is highly critical—if users get the wrong data, they may make an incorrect purchase decision. Incorrect data will also reflect poorly on the exchange and will damage the trust relationship between the user and the exchange.

Screen mockups
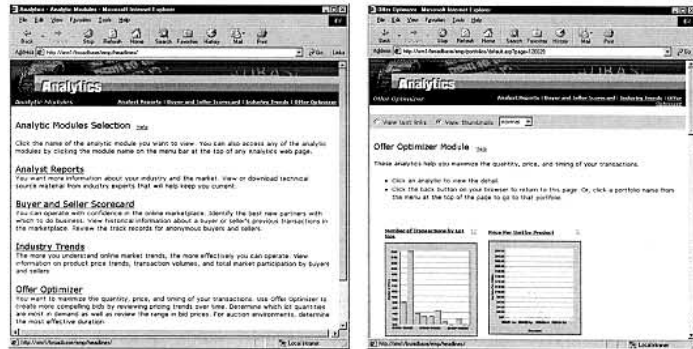
Screen descriptions

Analytic Modules Selection screen
Controls

UI Control	Description	Default Value	Valid Entries	User Input Required	Behavior
Analyst Reports	Menu Option (Text link)			No	Display Analyst Reports portfolio.
Buyer and Seller Scorecard	Menu Option (Text link)			No	Display Buyer/Seller Scorecard portfolio.
Industry Trends	Menu Option (Text link)			No	Display Industry Trends portfolio.
Browser back button	Browser button			No	Display Log In page
Browser forward button	Browser button			No	Disabled (unless the user clicked Back to get to this screen)
Help	Text link			No	Help topic: module descriptions and how to pick a module
Name of analytic module	Text link			No	Display corresponding Analytic portfolio page.

Tab order
When the user presses the Tab key, the focus advances from field to field in the following order:
1. When the page opens, focus is on Analyst Reports.
2. Buyer/Seller Scorecard
3. Industry Trends
4. Offer Optimizer
5. Help
6. Sequentially across menu options: Analyst Reports, Buyer/Seller Scorecard, Industry Trends, Offer Optimizer

Data validation
None.

Entry methods
- The user successfully logs into E-Marketplace.
- The user clicks the Browser Back button on a portfolio thumbnail page.

Exit methods
- The user clicks the Browser Back button.
- The user selects an analytic module.

Future requirements
- Create a new report from scratch
- Buy a report.

- Completing design phases in parallel and handing off designs for coding incrementally. We emphasized that our process is an iterative process, not a traditional waterfall process.[11]
- Focusing on getting the developers' buy-in to the new design process. We did this by involving them in every step of the design. In *The Elements of User Interface Design*, Theo Mandel discusses the importance of creating a multidisciplinary product design team.[10] Involving a wide range of people not only provides the full spectrum of skills needed for good design, but it also increases the team's buy-in to the design.
- Creating a shared vision among all team members.[7] Our proof-of-concept prototype produced early in the design process conveyed the product vision to the entire company. This gave every team member a vision of what the product would do to help B2B users. It took the product from seeming like something too large and impossible to produce to something we could actually design and build within our time constraints.
- Distributing articles and Web site information to developers pertaining to Web application usability and design. This increased the developers' awareness of the need for usability in Web-based products.

Accruing benefits from the process

Involving the entire development team in the design phases had a number of benefits. It gave developers a say in what they would be developing, and it showed them the volume of work that had to be done before coding could begin. It gave us more complete requirements and designs because of collaborative input from multiple disciplines. It also shortened the calendar time spent on each design phase. This enabled us to do user analysis and detailed design while staying on schedule.

We were also able to demonstrate to the whole team the importance of identifying our users and understanding their experiences. This was the beginning of a user-centered culture at Decisionism.

Our process helped dissolve communication barriers between human factors and development personnel. Because team members were involved in developing user profiles and task analyses from the start, we lost no time communicating user research findings and convincing developers of what users needed. Team members stopped viewing user requirements as something imposed on them and started viewing them as the purpose for the project. Design meetings emphasized how the application should work from a user's perspective.

Finally, by distributing the human factors workload, we were able to accomplish human factors activities in the time permitted. Instead of the classic problem of not enough human factors people to do the work, one human factors engineer was able to oversee all human factors activities and keep a big-picture perspective of working toward usable design.

> **Released in December 2000, our application, called E-Marketplace, was the first B2B analytic application of its kind in its market.**

Released in December 2000, our application, called E-Marketplace, was the first B2B analytic application of its kind in its market. When Broadbase Software approached Decisionism about acquisition, Decisionism illustrated the viability of getting a B2B product to market using the proof-of-concept's prototypes, user profiles, site maps, and enhanced use cases. Since Decisionism had not yet released a B2B analytic product, this demonstration enhanced our appeal as an acquisition candidate. The human factors and user interface design team, now part of Broadbase Software, is implementing the processes described in this article for Broadbase, along with the three other software companies Broadbase recently acquired.

Beyond the business benefits, the development team reported several positive results from this process. Team commitment improved in getting the product to market. The marketing, human factors, and development teams worked closely together to create a product vision and design. Creating the project plan was greatly simplified.

Developers saw the product as a whole instead of focusing only on the individual features or components they were coding. They also understood the interdependencies between features and worked together to make a cohesive product.

Developers had time to focus on solving implementation issues and coding the product. Having clear site maps and screen designs meant that they didn't have to spend time deciphering requirements documents or worrying about details of screen flow and layout. The application flow in the site map expedited identifying and resolving business and presentation logic issues in the technical architecture and made it easy for them to identify dependencies among features.

Identifying navigation and application flow problems at the site map phase minimized the number of defects that our QA engineer found during final testing. The QA engineer used the site map as a reference in planning test cases. Moreover, the detailed site map and use case documents controlled scope creep by clearly outlining what needed to be developed.

Developers accepted human factors as a key part of the design process and began seeking out human factors and user interface design team members for design guidance.

At the time of our acquisition by Broadbase, the development team was required to completely change the underlying technologies, development language, and third-party components. The technology-independent nature of the site maps and use cases made this change possible. In fact, the development team was able to make the required changes and still deliver the product three weeks before the deadline.

Most importantly, E-Marketplace hit the mark with B2B net markets and their customers. While we have not yet completed formal usability tests, we gathered subjective feedback and cognitive walkthrough data throughout our design and development process. We responded to customer problems and suggestions, and customers successfully navigated our user interface during cognitive walkthroughs. Ultimately, we provided customers with a targeted analytic application for doing business in online markets.

The first release of E-Marketplace was a stake in the ground that redefined our design processes and development culture. Future plans for E-Marketplace include formal usability testing and integration into the Broadbase analytic application suite. Like the user interface, our Web application development process will be iterative. We also plan to inte-

About the Author

Molly Hammar Cloyd is the human factors engineer at Broadbase Software. She joined the company after starting the human factors function at Decisionism, an analytic applications company recently acquired by Broadbase. She is now building a user-centered design group at Broadbase. Contact her at Broadbase Software, 4775 Walnut St., Ste. 2D, Boulder, CO 80301; molly@broadbase.com; www.broadbase.com.

grate a few more techniques into our process for future product releases:

- While group design and storyboarding sessions helped us generate a broad range of design ideas, we plan to experiment with parallel design, in which designers sketch screens separately before coming together to combine efforts. We hope this will expedite the initial screen mock-ups and facilitate generating more design options for the team to choose from.
- We plan to conduct formal usability testing at multiple points along the design process. We have received funding for usability testing equipment and resources so we can gather quantitative usability data, identify specific areas for design improvements, and measure improvements against baseline usability results.
- We will iterate user interface designs based on usability test results, user feedback, market requirements, and new Web application technologies.[10] 🄪

References

1. S. Ward and P. Kroll, "Building Web Solutions with the Rational Unified Process: Unifying the Creative Design Process and the Software Engineering Process," www.rational.com/products/whitepapers/101057.jsp (current 2 Jan. 2001).
2. J. Nielsen, *Designing Web Usability: The Practice of Simplicity*, New Riders, Indianapolis, Ind., 2000.
3. R. Reddy, "Building the Unbreakable Chain," *Intelligent Enterprise Magazine*, vol. 3, no. 3, www.intelligententerprise.com/000209/feat3.shtml (current 2 Jan. 2001).
4. A. Cooper, *The Inmates Are Running the Asylum*, Macmillan USA, Indianapolis, Ind., 1999.
5. S. Robertson and J. Robertson, *Mastering the Requirements Process*, ACM Press, New York, 1999.
6. J.S. Dumas and J.C. Redish, *A Practical Guide to Usability Testing*, Ablex, Norwood, N.J., 1993.
7. J. McCarthy, *Dynamics of Software Development*, Microsoft Press, Redmond, Wash., 1995.
8. G. Schneider and J.P. Winters, *Applying Use Cases: A Practical Guide*, Addison-Wesley Longman, Reading, Mass., 1998.
9. B.L. Kovitz, *Practical Software Requirements*, Manning, Greenwich, Conn., 1999.
10. D. Mayhew, *The Usability Engineering Lifecycle*, Morgan Kaufmann, San Francisco, 1999.
11. T. Mandel, *The Elements of User Interface Design*, John Wiley & Sons, New York, 1997.