About This Free Report

This report is a gift for our loyal audience of UX enthusiasts. Thank you for your support over the years. We hope this information will aid your efforts to improve user experiences for everyone.

The research for this report was done in 2012, but the majority of the advice may still be applicable today, because people and principles of good design change much more slowly than computer technology does. We sometimes make older report editions available to our audience at no cost, because they still provide interesting insights. Even though these reports discuss older designs, it’s still worth remembering the lessons from mistakes made in the past. If you don’t remember history, you’ll be doomed to repeat it.

We regularly publish new research reports that span a variety of web and UX related topics. These reports include thousands of actionable, illustrated user experience guidelines for creating and improving your web, mobile, and intranet sites. We sell our new reports to fund independent, unbiased usability research; we do not have investors, government funding or research grants that pay for this work. Visit our reports page at https://www.nngroup.com/reports/ to see a complete list of these reports.

HOW TO SHARE

Do not link directly to the PDF file (the hosted address could change). Instead, we encourage you to distribute the following link to the report’s page on our website to allow people to decide whether to download it themselves:

http://www.nngroup.com/reports/best-applications-2/

REPORT AUTHORS

Kathryn Whitenton, Kara Pernice, Patty Caya, and Jakob Nielsen
Copyright Notice

Please do not post this document to the internet or to publicly available file-sharing services.

This report is free, but it is still copyrighted information that may be updated from time to time, so please don't distribute this file or host it elsewhere.

Even when people post documents with a private URL to share only with a few colleagues or clients, search engines often index the copy anyway. Indexing means that thousands of people will find the secret copy through searches.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About This Free Report</td>
<td>2</td>
</tr>
<tr>
<td>Executive Summary</td>
<td>12</td>
</tr>
<tr>
<td>Complex Applications</td>
<td>12</td>
</tr>
<tr>
<td>Lightweight Applications</td>
<td>12</td>
</tr>
<tr>
<td>Workflow Applications</td>
<td>12</td>
</tr>
<tr>
<td>Multi-Platform Applications</td>
<td>12</td>
</tr>
<tr>
<td>Augmenting Human Capabilities</td>
<td>13</td>
</tr>
<tr>
<td>Linear Workflow and Wizards</td>
<td>14</td>
</tr>
<tr>
<td>Living in an Ecosystem</td>
<td>15</td>
</tr>
<tr>
<td>Super-Tooltips: Interaction Design Technique of the Year</td>
<td>16</td>
</tr>
<tr>
<td>Usability Methods</td>
<td>17</td>
</tr>
<tr>
<td>Overview of the Winners</td>
<td>18</td>
</tr>
<tr>
<td>Made in America but Used Around the World</td>
<td>18</td>
</tr>
<tr>
<td>It Doesn’t Take a Village to Build a Great App</td>
<td>18</td>
</tr>
<tr>
<td>In-House Teams Lead the Way</td>
<td>18</td>
</tr>
<tr>
<td>Screenshots are Just a Moment in Time</td>
<td>20</td>
</tr>
<tr>
<td>Design Trends</td>
<td>21</td>
</tr>
<tr>
<td>Designing for different Types of Users</td>
<td>21</td>
</tr>
<tr>
<td>• Presenting Various Options and Views to Meet Needs</td>
<td>21</td>
</tr>
<tr>
<td>• Preferences and Customization:</td>
<td>21</td>
</tr>
<tr>
<td>Making Top Tasks Obvious for All</td>
<td>21</td>
</tr>
<tr>
<td>Presenting Various Options and Views to Meet Needs</td>
<td>22</td>
</tr>
<tr>
<td>Multiple Paths</td>
<td>23</td>
</tr>
<tr>
<td>Preferences and Customization:</td>
<td>23</td>
</tr>
<tr>
<td>Wizards</td>
<td>23</td>
</tr>
</tbody>
</table>
Linked Controls ................................................................. 24
Downloading or Sending Data to Another App to Review and Manipulate .................................. 24
Making the Most of Space .................................................... 25
Safety Nets .................................................................. 25
Help (inline) .................................................................. 26
Complex Applications ........................................................ 27
BondWorks™ Advisor Platform ............................................ 28
  Overview .................................................................. 28
  Users ..................................................................... 28
  The Application ....................................................... 29
  Design Process .......................................................... 41
    Usability Methods .................................................. 41
  Lessons Learned ....................................................... 47
  Team ..................................................................... 49
GEN2i ........................................................................ 51
  Overview .................................................................. 51
  Users ..................................................................... 52
  The Application ....................................................... 52
  Design Process .......................................................... 61
    Usability Methods .................................................. 61
    Development Approach ......................................... 65
    Timeline .................................................................. 66
  Lessons Learned ....................................................... 67
  Team ..................................................................... 68
OEConnection ................................................................. 69
  Overview .................................................................. 69
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executives</td>
<td>70</td>
</tr>
<tr>
<td>The Application</td>
<td>70</td>
</tr>
<tr>
<td>Design Process</td>
<td>82</td>
</tr>
<tr>
<td>Tracking the Design Evolution</td>
<td>83</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>87</td>
</tr>
<tr>
<td>Project Team</td>
<td>88</td>
</tr>
<tr>
<td>Lightweight Applications</td>
<td>91</td>
</tr>
<tr>
<td>CycleStreets</td>
<td>92</td>
</tr>
<tr>
<td>Why They Were Chosen</td>
<td>92</td>
</tr>
<tr>
<td>Overview</td>
<td>92</td>
</tr>
<tr>
<td>Who Are the Users?</td>
<td>93</td>
</tr>
<tr>
<td>The Application</td>
<td>93</td>
</tr>
<tr>
<td>Innovation</td>
<td>95</td>
</tr>
<tr>
<td>Consistency</td>
<td>97</td>
</tr>
<tr>
<td>Flexibility</td>
<td>99</td>
</tr>
<tr>
<td>Information Display</td>
<td>104</td>
</tr>
<tr>
<td>Design Process</td>
<td>105</td>
</tr>
<tr>
<td>Implementation</td>
<td>105</td>
</tr>
<tr>
<td>JQuery Mobile Framework</td>
<td>106</td>
</tr>
<tr>
<td>Standalone Code</td>
<td>106</td>
</tr>
<tr>
<td>Lessons Learned</td>
<td>106</td>
</tr>
<tr>
<td>Team</td>
<td>107</td>
</tr>
<tr>
<td>VitreaTeach</td>
<td>108</td>
</tr>
<tr>
<td>Overview</td>
<td>108</td>
</tr>
<tr>
<td>Who are the Users?</td>
<td>108</td>
</tr>
<tr>
<td>The Application</td>
<td>109</td>
</tr>
</tbody>
</table>
1. Steps in a task or process presented out of order on a page or screen ................................................................. 301
2. Missing Simple Paths ............................................................................................................................ 301
3. Invisible Controls and Cryptic Icons .................................................................................................... 301
4. Hideous Forms ................................................................................................................................. 302
5. Unclear Differences When Comparing Choices .............................................................................. 302
6. indistinguishable Current Selection Status ......................................................................................... 302
7. Branding Features or Menu Commands ............................................................................................ 303
8. Only the Promise of Content .............................................................................................................. 303
9. Views and Modes That Trap People .................................................................................................. 303
10. Bad Dialog ..................................................................................................................................... 303

About the Authors .......................................................................................................................... 304
Acknowledgements ......................................................................................................................... 306
Executive Summary

The winners for best application user interfaces of the year are:

Complex Applications
- The web-based **BondWorks** wealth management platform (BondDesk Group LLC) provides financial advisors with a streamlined system to trade fixed-income assets.
- The portable **GEN2i** (Hottinger Baldwin Messtechnik GmbH, Germany) lets users record engineering data in the field.
- **OEConnection** (OEConnection LLC) lets parts department employees at auto dealerships and repair facilities process more than 6 million original equipment parts transactions monthly.

Lightweight Applications
- **CycleStreets Mobile** (CycleStreets Ltd. and Anna Powell-Smith, UK) helps cyclists plan safe, fast routes for cycle journeys using crowd-sourced map data.
- **VitreaTeach** (Vital Images, Inc.) is an educational tool that radiology residents and practicing radiologists can use to save, manage, and share medical images.
- **Wakelicious** (Utopian Army, The Netherlands) is an alarm clock that wakes you up by playing YouTube videos sent by your friends.

Workflow Applications
- The **Climate Action Planning Tool** (National Renewable Energy Laboratory) helps organizations reduce their carbon emissions.
- **eReview** (International Monetary Fund, IMF) integrates complex business functions into an organization’s intranet.
- The self-service **Eventbrite** (Eventbrite Inc.) software tool helps event hosts organize, sell, and manage tickets for events. The **Create Event** workflow, which won the award, helps users set up a new event within Eventbrite’s online system.
- The **Hobsons CRM** (Hobsons, Inc.) product-suggestion portal tool lets customers make comments and suggestions.

Multi-Platform Applications
- **Dixio** (Semantix Group SL, Spain) offers users multiplatform lookup capabilities.
- **Highlight** (Cohdoo, LLC) lets users with iOS devices make audio recordings, mark interesting points, and share recordings with others.
- The **Mobile Unified Communication** client (Verizon Wireless) lets business customers link their mobile and business lines.
In contrast to our other design awards, we decided to split the application award into categories. It doesn’t make sense to judge small consumer apps that do a single thing together with huge enterprise applications that manipulate immense sets of convoluted data—such as the 4 million different bonds that traders manage through BondWorks. Workflow applications that integrate many steps or coordinate several user roles have still different requirements, as do multi-platform apps that work across several devices.

Solving a big problem is more difficult than solving a small problem. However, if your app does only one thing, it had better to do it really, really well. So, the criteria for polish are stricter for small applications than for enterprise applications. With thousands of screens in some applications, they can’t all be perfect, but the flow between screens and the ability to select the correct feature for any given situation must be great. So, instead of comparing apples and oranges, we decided to judge each application style in relation to its peers and select the best designs in each category.

That we would ever classify a radiology application as “lightweight” defies the imagination, but it’s appropriate in the case of VitreaTeach because it focuses on one thing: allowing residents and experienced radiologists to manage images for learning purposes. We might even call it the Pinterest of medical science. But, of course, Pinterest itself wouldn’t work for radiologists — partly because of the privacy concerns for any medical information and partly because of the benefits that domain-specific features provide.

For example, VitreaTeach both offers special access to cases from the user’s own hospital and knows something about anatomy and pathology. Interestingly, one of the things that make the design cleaner than other medical systems is the simple tagging that provides much of the taxonomy support for the images. Designers simplified tagging after user research found that full, formal anatomy and pathology taxonomy hierarchies added too much overhead.

**AUGMENTING HUMAN CAPABILITIES**

Software’s real goal should not be to simply process transactions in a system where users are nothing more than data operators who click required buttons to make things happen. Rather, software should work to augment human capabilities, helping us to overcome weaknesses and emphasize our strengths.

A vital way for computers to help users is to direct their attention to a smaller number of important issues instead of overwhelming them with all possible options. For example, BondWorks originally presented 85 attributes of tradable bonds on the search results screen. However, search log analysis showed that nearly three quarters of users were searching on only 10 of the attributes, so the screen was redesigned to focus on those attributes.

OEConnection proactively takes the next step when a spare part is backordered from the manufacturer: the software automatically proceeds to search other dealers’ inventory for the desired part. Traditionally, users would have had to issue each type of search separately from separate application areas. But that’s exactly the approach that turns humans into robotic button-ushers; better to turn the computer into a problem-solving tool that understands the domain and lets users expend their brainpower on higher-level issues.
As another example, the Highlight application’s basic feature is to make audio recordings searchable by letting users add a marker when something interesting occurs. For real-world tasks — such as conducting interviews or sitting in a foreign-language class — users wouldn’t actually know that “something interesting” was being recorded until after it happened. To accommodate this common scenario, the application has an offset feature that automatically places the marker into the recording a certain number of seconds before the button is pushed. Instant time travel.

More to the point, Highlight uses the computer’s flexibility to augment human skills by taking over the tedious job of rewinding the recording when users realize that something interesting should have been marked. This reduces users’ cognitive load and frees them to concentrate on listening to the event they’re recording instead of attending to the device.

Another simple way Highlight augments human capabilities is in its ability to associate a picture with each recording. This is particularly simple when doing interviews, where users can snap a photo of the interviewee with the same phone that’s recording the audio. These photos come in handy later when retrieving recordings from long lists, because it’s often easier for our poor brains to associate an interview with the person’s photo instead of relying on a textual description.

We also saw a trend to focus screen real-estate allocations on the information that’s most important in the current context, while showing less UI chrome. More content, less application overhead.

Some winning applications had special small control panels for direct access to important operations that would usually have required a switch in context. This doesn’t quite equate to a modeless design, because complex applications do tend to have several distinct stages or areas of workflow and data. But surfacing crucial features to live above the regular UI architecture is yet another way of empowering users to act on their insights instead of delaying them by the rigid demands of the way the application is structured.

**LINEAR WORKFLOW AND WIZARDS**

We want to empower users to be creative and accomplish advanced things with our software. But we should also recognize that users sometimes just want to get their tasks done without having to explore numerous options and new ideas.

To speed users through infrequent or complicated tasks, it’s often good to present a linear workflow with minimal disruptions or alternatives. Yes, the lack of flexibility can feel constraining, but it can be faster to just power through all the steps instead of having to ponder which steps are needed. Also, the cost of too much freedom is that users have to decide the order of the steps — something that they’re often happy enough to delegate to the computer.

Wizards are the classic approach to linear workflows, and several winning applications include nicely designed wizards. The Climate Action Planning Tool uses the simpler approach of offering numbered steps to guide users. An early design included a *Before You Start* step, but user studies showed that people ignored it. In the final design, the Before activities appear under *Step 1: Gather Baseline Energy Consumption Data*. Users just want to get going, so you need to be somewhat heavy-handed if setup work is required before starting the real activity.
Wizards don’t work for everything. In user testing, Eventbrite developers discovered problems with a wizard for creating new events in the system. Because users’ mental models of setting up a new event didn’t include all the necessary steps, they often dropped off and didn’t complete the wizard. So, the team decided instead to go with a 2-step process in which users first entered all the information and then customized their pages.

When creating a linear workflow, take care not to signal to users that they can ignore subsequent steps — unless it’s true. For example, after the first 6 steps, Eventbrite’s workflow was as follows:

- Step 7: Change the color scheme on the registration page.
- Step 8: Choose privacy settings.

It might have seemed logical to first fully create the thing that the privacy would apply to, and then to choose the privacy settings. To some users, however, Step 7 signaled that all the “real” set-up was done and that the remaining steps would be frivolous decorating. If they didn’t care about the decor or were satisfied with the default colors, some people simply stopped at Step 7 and never got to the privacy options.

**LIVING IN AN ECOSYSTEM**

A major trend among the winning applications is a modularized approach to the total user experience, with major features outsourced to other applications. Designers accept that their product will live in an ecosystem of other applications (and websites) and that users will often prefer to do certain things elsewhere in that ecosystem.

Relying on other applications to perform work for your app is obviously not a new idea. Having applications work together goes back at least to the pipes in Unix. But we see this done more often now than in years past, when a more monolithic approach was common.

Unless you’re IBM or Google, you’ll never build a spreadsheet that rivals Excel. Furthermore, most business users already own Excel and know how to use it. Thus, instead of building spreadsheet-like features into your application, it might be better to find a smooth way to integrate with Excel through data export and import. Many data-rich applications among our winners chose this course.

As another example, Wakelicious didn’t implement a complete system for uploading and storing videos for Internet playback. Instead, it offers a simple integration with YouTube, which not only saved the team tremendous development effort but also gives users access to a pre-populated content library from which to choose wakeup calls to send their friends.

One caveat about relying on other software is that it’s often best to modify or simplify the user interfaces to fit the needs of your users; because of its broader purview, 3rd-party software often embodies unneeded complexity that it’s best to hide. eReview, for example, built on top of SharePoint but designed a more streamlined process than the one provided out-of-the-box.

Although it’s great to leverage existing applications and Internet resources, applications must also provide their own value-add in their core domain. This was most clearly shown by the Climate Action Planning Tool, which actually started out as a regular Excel spreadsheet that could perform the necessary calculations.
Developing it into a real application to guide facilities managers through the workflow — and explain and illustrate the complicated concepts involved in reducing CO2 — made the experience much more successful.

**SUPER-TOOLTIPS: INTERACTION DESIGN TECHNIQUE OF THE YEAR**

In *Application Design Showcase #1*, we declared the lightbox to be the interaction design technique of the year. This time, we saw improvements in a broad range of interaction techniques but no single new widget that was widely used.

Several applications, however, enhanced an old idea: the tooltip. These “super” tooltips stretch far beyond their original intent and thus earned the spot as the year’s most improved interaction design technique. Emerging in the 1990s, tooltips (or “bubble help”) started off by providing small explanatory texts that appeared as contextual help when the user pointed to a command button or other interface element.

The new super-tooltip isn’t actually a “tip,” but we’ll stick to this well-established name with its long history in GUI development. A super-tooltip is actually an explanatory pop-up that appears as a rollover effect when users point to an item in the user interface. This “item” might be a command or a button — as with traditional tooltips — but it can also be a user-created object, such as a message or product order. The following screenshot shows a typical super-tooltip visual design:

![Example of a super-tooltip from Eventbrite.](image)

In addition to static help text, super-tooltips usually contain dynamic status information about the object that users are pointing to. This lets users see the information in the context of a larger overview screen, instead of having to click through to a separate screen with detailed object properties and editing options. Super-tooltips are thus particularly suited to complex applications with many data objects that users must consider together.

---

1 Available at [http://www.nngroup.com/reports/applications/showcase-1/](http://www.nngroup.com/reports/applications/showcase-1/)
The super-tooltip shouldn’t include editing options; it should be for information only, since it appears on hover and disappears when users move the mouse. Users should also retain the option to click through to a specific screen that’s optimized for viewing and revising an object’s properties in greater detail.

The entire Dixio application could be viewed as an even more souped-up super-tooltip. However, that wouldn’t be entirely accurate because Dixio explanations appear when users double-tap (or long-tap) a word. Thus, it’s not a rollover/hover effect and has to be explicitly dismissed.

**USABILITY METHODS**

Most winners extensively employed a wide variety of usability methods, which is certainly one reason they ended up with such good user interfaces. Iterative design and user testing were particularly common and frequently used to grind down complexity relative to early design ideas that seemed good but were too difficult. Another common trend was the steady reduction in documentation and help text throughout the iterations, as user testing repeatedly revealed the old lesson that users don’t read very much.

Simple user testing with a handful of users can often substantially improve product success, and was used by 85% of the winning projects. For example, an early design for the Mobile Unified Communication client used the command *Add Call* for placing a new phone call while putting the current party on hold. Although this seems logical enough, it was confusing to users who thought the command was for conference calls. Changing the command label to *New Line* did the trick.

One-third of the winners used field studies (also known as ethnography), venturing onsite to observe users’ natural behaviors in their own environments. From BondWorks sitting with brokers as they made deals, to OEConnection hanging out in dealership parts departments, to VitreaTeach watching radiologists at hospitals, domain-specific applications require domain-specific user research.

Although it seems obvious that domain-specific applications need field studies, most design projects in the past typically did without them. When we analyzed the first decade of Intranet Design Annual winners, we found that field studies were used by only 10% of the winners from 2001–2003 and just 23% of winners from 2007–2009. Although this represented more than a doubling within a single decade, it’s still sad that less than a quarter of good intranet teams ran field studies at decade’s end. Going from a quarter of projects to a third in a few years shows continued progress for this important method.

In addition to field studies, we also saw substantial growth in a usability method that was often overlooked in the past: sitting in on support calls. Knowing why customers call with problems is a prime way to discover pain points — as long as you recognize that a customer’s description of the problem might be far afield from the underlying root cause that the redesign must address.
Overview of the Winners

In this installment of the Applications Showcase we increased the number of selected winners from 10 to 13 (a clear reflection of the quality of the submissions we received,) and among the applications selected for inclusion in this year’s report we’ve identified a few commonalities across teams, including:

- **US dominance:** A majority of winners this year, 2012, (69% compared with 50% in the 2008 edition of this report) are from US-based companies.

- **Small teams—big impact:** Teams large and small built the applications featured in this report, but small teams are proving that it doesn’t take an army to build a great app. Nearly a third of the winning teams had only two members.

- **Made in-house:** Fewer teams this year share their accolades with outside agencies as every app in this report was designed and built, at least in part, by an in-house team.

**MADE IN AMERICA BUT USED AROUND THE WORLD**

The United States leads the pack among the winning organizations. US-based companies produced nine (of the 13) winning applications featured in this report, while companies from across Europe round out the winner’s circle with one application each coming from the UK, The Netherlands, Germany, and Spain.

So while the companies represented in this report may not be all over the map geographically, what users can do with these winning applications certainly is.

The thirteen teams featured in this report built applications with uses ranging from interpreting advanced diagnostic measurements (GEN2i) to using a YouTube video sent by a friend as an alarm clock (Wakelicious.) They provide tools for industry sectors as different as fixed-asset (bond) investing (BondWorks) and a non-profit organization dedicated to cycling. The teams featured here serve many different sectors, including: scientific engineering, financial services, retail automotive, telecommunications, education, government, and NGO and general use.

This year’s winning teams may be mostly based in the US but the reach of their efforts is felt by users across the globe.

**IT DOESN’T TAKE A VILLAGE TO BUILD A GREAT APP**

Team size, as usual (and as we find with much of our research,) spans the spectrum from large (15 members) to small (a single user experience designer,) but a noticeable trend emerged this year as winning duos proved that two heads really are better than one (or more.) Two-person teams designed four of the winning applications (Vitrea Teach, Cycle Streets, Wakelicious and Cohdoo Highlight.) And if you only have two resources to throw at a project it seems a pairing between a designer and a developer can yield winning results. It did for these teams.

**IN-HOUSE TEAMS LEAD THE WAY**

The votes are in and so is the work, in-house that is. Far more teams forged ahead with the team they have on hand to design their winning application rather than
hiring from outside. All but one of this year’s winners did all or most of the work on their applications in-house. And even the teams that brought in reinforcements in the form of outside agencies and consultants did so for only a small component of the work.

The talented teams featured on these pages prove that it really doesn’t matter what size team you have or what language you speak. What really matters is that you put your resources toward good work: serve the needs of the end-user regardless of whether your technology is mission-critical or surprising and whimsical.
Screenshots are Just a Moment in Time

We appreciate the designers and organizations with winning applications in this report. They were generous enough to share their screenshots and stories with us. We would like to acknowledge that the screenshots and descriptions are from the version of the applications that we reviewed, and the design of these applications is ever-changing. We included as many design updates as possible up until the time of this report’s publishing, but please know that these apps may have already changed by the time you are reading their case studies. Regardless, we feel that the discussions about why the designs are so good and lessons about the teams’ processes will carry on for a very long period of time.
Design Trends

We can’t help but notice some design trends and commonalities across the excellent, though widely different, applications in this report. This section summarizes some of those patterns.

DESIGNING FOR DIFFERENT TYPES OF USERS

Application designers who have one clearly defined type of user are very lucky. Most app designers are faced with the challenge of designing for numerous different user types who have varying needs, tasks, approaches, and knowledge. What the great app designers in this report demonstrate is the ability to determine and contemplate the particular user types and the mindset of each at any given point in the UI.

Designers made decisions about how to cater to the different user types in different situations. Some ways the designers helped the different user groups include:

- **Making Top Tasks Obvious for All**: Identify essential starting points—the ones people actually use, not just the ones you want them to use
- **Presenting Various Options and Views to Meet Needs**: See and manipulate content in multiple ways, such as: text and images views, categories, and filters
- **Multiple Paths**: Offer multiple, but not too many, paths to do the same or a similar task (This means different methods, not listing the same command or link to access a feature multiple times)
- **Preferences and Customization**: Customize the UI and edit preferences to affect content pushed to the users, and arrangement and look of UI elements

MAKING TOP TASKS OBVIOUS FOR ALL

One good way to cater to different user types in the same app is to ensure that the main tasks that any users will want to do are incredibly easy and accessible. **Wakelicious** (a social video alarm clock) designers researched and contemplated before they came up with two buttons for their most important tasks in their app: **Set alarm** and **Wake a friend**. These are explanatory and cut to the heart of the app in the initial screen.

**Cohdoo’s Highlight** (recording) focuses intensely on the core action of marking a highlight. The majority of the screen is devoted to the **Highlight** button so that users can perform the action without even looking at the device.

**Hobson’s Product Suggestions** app designers knew that two of the app’s top tasks are 1) dealing with suggestions that have been made, and 2) writing new suggestions. Thus the top two tabs in the UI are simply **View Suggestions** and **Submit Suggestion**. These are easy to see, click, and the text is easy to understand.

On a similar note, the **Eventbrite** app displays two rows of options (tickets) by default in order to introduce the idea of creating more than one type of ticket.

Sometime users actually need to do something **before** they can use the app’s main function. The **Climate Action Planning Tool** app, which supports users with varying degrees of knowledge about energy use and goals for greenhouse gas
emissions, offers an incredibly clear call to action at the start: the *Start: Gather baseline data* button.

Verizon’s **Mobile Unified Communications** app caters to different users by acknowledging that some people mostly make phone calls from their business phone, some mostly make mobile calls, and some make roughly an equal number of mobile and business phone calls. The UI makes it possible for the user to quickly and easily choose between these (and they can also set up default quick access to one or the other.)

**PRESENTING VARIOUS OPTIONS AND VIEWS TO MEET NEEDS**

The IMF’s eReview (for managing the review and approval of report drafts) has separate tabs and workflows for different types of users; one area is set up for the department that creates a report, and other tabs are designed for departments that review and comment on reports. Similarly, **Dixio** (lookup technology) offers the option to search for a word’s definition as either text or images. So people who are doing more of a visual task, or those who tend to respond better to pictures than to words may be able to work in a more helpful and meaningful way because of this option. Dixio also translates any foreign language words into the language most of the article is written in.

Sometimes offering the same information sliced in different ways can help users find what they want. This same app, Dixio, presents search results defaulted to *all* results, but offers tabs for each individual dictionary (content source) which the user has searched in. So either people who are questioning a particular definition or people who respect one source more than another (even for a particular type of word) can find their answer. But the users who do not care for one particular source over another may stick with the default state, all sorted.

Within the view area of Hobson’s **Product Suggestions**, users may sort by sub-tabs—*All Suggestions, Suggested by Me,* and *Suggested by Co-Workers*. These choices enable people to easily review and compare suggestions by source. Additionally, designers included filtering options and a few different ways to refine results, including searching for a keyword, filtering checkboxes, and sorting the list by other criteria.

The **VitreaTeach** app (for viewing and sharing medical images) offers hierarchy, filters, and the possibility for user-generated tags; all of which help radiology residents to review pertinent, interesting images from old cases.

Messages in the lower left on the **BondWorks** app’s homepage allow brokers to keep abreast of news, alerts, and announcements. Advisors may sort this list and see, for example, all orders for a particular account, or all pending orders, which can help them keep abreast of current work and plan for the future.

This same app offers a bond comparison tool which allows advisors to choose some of the bonds they are most interested in and compare them. Designers even anticipate the user’s next steps and make it possible to buy a bond or print the comparison table.

The **Climate Action Planning Tool** app presents the final report after users enter data, but also presents the users carbon emissions reduction goal as a drop-down list so they may easily test different goals and see the specific results in a graph.
MULTIPLE PATHS

**GEN2i**, app for recording engineering data in the field, offers a fun way to scroll through data values for the standard sequencing using a rotary knob on a touchscreen. An added touch of whimsy, the knob makes a clicking sound as it spins. But sometimes users are not in a whimsical mindset: People who just want to enter the value quickly may skip this feature and instead use a different UI, a more traditional button encompassing the range of numbers to choose from. For full flexibility, the app also allows people to enter their own values outside the offered range.

**BondWorks** is designed for financial advisors and brokers who are working with bonds either 1) all day every day, or 2) rarely. An Advanced Search link leads to involved search options which help cater to the different user types, but the simpler default search doesn’t get in the way for any users.

The **OEConnection** app for ordering and tracking automotive parts always displays the current part number at the top of the list, but users may also manually type a new part number if they know it.

The **Highlight** app allows editing elements on the Details screen by tapping and holding them, but this screen also includes an explicit Edit button in the top right corner for users who have not yet learned the gestures or who prefer using clickable buttons to gestures.

PREFERENCES AND CUSTOMIZATION

**Dixio** allows the user to edit preferences to see information in the format he chooses, and allows people to customize their own glossaries.

Verizon’s **Mobile Unified Communications** app allows users to customize the defaults for the main buttons (or widgets) setting a button to the type of most employed phone call, either Business, Mobile, or Prompt me [to choose business or mobile] every time.

**CycleStreets** designers understand that some riders may be cautious and prefer to always take quiet streets even though a busier street might be quicker. In the Preferences area of the app, user may easily customize the journey planning process and set their default route type preference, as well as their normal speed. They may also turn off the GPS location feature in order to conserve battery life.

The **Eventbrite** app (for online ticketing platform) allows users to change the visual appearance (background color, text color, images) so it suits their preferences.

WIZARDS

Wizards and linear processes are here to stay, especially for complicated and infrequent tasks.

Verizon’s **Mobile Unified Communications** helps the user get his app set up in the way he needs it. A wizard process, planned for various user scenarios, makes customizing this very productive and not overly constraining.

---

The **Climate Action Planning Tool** app offers numbered steps to lead people through their process. Interestingly in research studies designers learned that the *Before You Start* step was ignored as users wanted to just get into the work. In the current design the important steps that used to be categorized under *Before You Start* are now under a section called *Step 1: Gather Baseline Energy Consumption Data*. This tested far better and is a very good lesson for other designers creating stepped UI’s such as wizards: Be cautious with information-gathering screens or pages as users often think they are information-only pages and not necessary to their process.

**LINKED CONTROLS**

Linked controls, a form of progressive disclosure, hide choices users don’t need and show only those that apply to the current situation or questions the system is asking. It’s easier for designers, often, to just present everything to the user at once, but winning app designers do extra work to create categories and give users information and questions just in time.

The *OEConnection* app for ordering and tracking automotive parts makes it easy to choose the right part number. Specifically, designers implemented a linked control relating the *Part #* field with the *New Part #* field. The *New Part #* field considers the number in the *Part #* field as well as other information in the order, and based on database and inventory knowledge suggests like or better parts when the *New Part #* field is clicked. A drop-down displays part numbers the user may possibly want.

IMF’s *eReview* app changes the display of buttons depending on both which step the user is in in the process (you can do certain things at certain points) and the user’s access (based on her role.)

The second button, *Quick Call*, in Verizon’s *Mobile Unified Communications* app changes depending on whether users chose *Business*, *Mobile*, or *Prompt me [to choose business or mobile]* every time as the first button choice. The second button is a linked control and only presents choices related to the first button’s selection.

**DOWNLOADING OR SENDING DATA TO ANOTHER APP TO REVIEW AND MANIPULATE**

Great designers know they shouldn’t reinvent the wheel, nor should they code in grand, complicated features when their users already have other apps which support them. When designers do make these imposing attempts the resulting implementation of the complicated features are often a half-measure as compared to the other robust app. Or the new features are fine but add too much complexity to the app as a whole. Better is to make an export possible, simple, and reliable so people may take advantage of another jam-packed app.

The *Hobsons* app does this, enabling users to copy the data (suggestions) to a spreadsheet file by just clicking the *Download Suggestions* button. Users may then use a spreadsheet app to manipulate the information.

**GEN2i** designers know that their users typically use their app to do measurement tasks, maybe some viewing and basic measurements, but engineers look for the real power in another large software package. Thus, the *Export Recording* option is very easy to see and use, and offers options to export the full or partial in various formats such as Word, Excel, or PDF.
A related but somewhat different feature from downloading data from an app appears in the **Climate Action Planning Tool** app. Designers derived a downloadable worksheet to help users gather information that is essential to getting their task done, and getting the most value out of the application. Users must consult various other people about the different energy sources on campus. Thus, designers created a worksheet that is optimized to fit the way users perform the task, looks like the online form in the app which makes it easy to transfer the data into the application later, and is a PDF so the look remains intact when it is printed. This same app does allow saving and printing the final report results, based on all the data the user input in the app, as a PDF document as well.

**MAKING THE MOST OF SPACE**

Using screen real estate effectively is a skill that the winning designers in this report all have. These designers prioritize the tasks along with what needs to be communicated in the allowable space.

The **Highlight** (recording) app designers chose to not include a bottom navigation bar in their mobile app in order to keep a strong focus on the primary task of each screen.

**CycleStreets** also relegates the global navigation options (such as switching between different types of tasks) to a *Home* screen. This leaves more space on the map display screen for actually viewing the map, and for controls that let users quickly flip between different versions of a route.

**VitreaTeach** designers knew that students were very interested in the text accompanying medical images, not just the images themselves, so the screen real estate is balanced between images and their accompanying descriptive text.

Replacing content in the same area is one way to save space in a dashboard, or any UI. **GEN2i** changes the content in the lower right rail depending on what is selected.

Sometime layout is critical to understanding functionality. **Eventbrite** presents the steps in its *Create* workflow into a vertical line, to make the sequence of steps obvious and help simplify a complicated process.

**SAFETY NETS**

The **GEN2i** app automatically saves data it collects, as does **Highlight**. Designers of these apps are of the mind that if the user takes time to set up a collection or a recording then they meant for it to be saved. So why make people do additional work to keep their data? In these particular cases, unless the system would slow down or run out of space there is no reason to force the user to save.

**Highlight** also anticipates that many users will realize they want to mark a highlight only after they experience the action, so the app allows users to automatically mark all highlights for a few seconds prior to the important event.

**Wakelicious** has a backup alarm clock built into the application. It is turned on by default, so that even if the app is disabled or loses an internet connection, users will still be woken up on time.

**eReview** manages collaboration among different departments, and everyone needs to be informed of the project status. The system manages this not just by making the status and task assignments visible, but by proactively sending out notifications when tasks are assigned or completed.
OEConnection designers realized that users may not always remember to check for conversion opportunities, or parts within an order that they aren’t specifically requested to fill, but that they could take the initiative to fill. The app places this information front and center within the order details screen, so users won’t overlook opportunities.

**HELP (INLINE)**

Most applications focus on ensuring that controls and labels are self-explanatory, and that users don’t need to turn to online help. Wakelicious designers believe this, but also make help accessible for users who seek out the help icon. Initially the app offered an abundance of help, but testing showed (a very common usability finding) that users didn’t read the prominent instructions, so they were shifted into an auxiliary role. The few areas of the app where users are likely to be interested in learning how it works rather than just using it—like the Settings screen—have brief explanations right next to the relevant controls. Many great applications give bite-sized, accommodating assistance in-line in this manner rather than as huge help files.

BondWorks includes hover-activated tool-tips next to search fields, to quickly explain the purpose of each field.

The Climate Action Planning Tool app offers a *What’s this?* link after various questions in the app, such as *Buildings_% compounded over 40 years.* The link opens a pop-up box describing the *Buildings* topic as well as the typical growth rate per year for that item.
Complex Applications

Applications in this Complex category are those which are usually meant to support involved processes or complicated problems. Some tasks and content are technical, mechanical, or financial in nature.

The users of complex applications are often highly skilled in a particular domain, such as engineering or finance, and may be willing to learn some of the nuances and intricacies of a deductive application design because of the luxuries available with more advanced and powerful features. This doesn’t mean, however, that the app should not still follow good design principles and be as simple to use as possible.

One of their great common problems in complex applications, no matter what the domain, is the need for seeing or having the ability to access large amounts of data all at the same time.

UI’s that often help support complex activities include:

- Dashboards and portals
- Using the same space to rotate through different content
- Involved, advanced search capabilities
- Possibilities for viewing data in different ways
- Exporting data to other applications
- Graphical displays of data.

Winning applications in this category include:

- The BondWorks™ wealth management platform, a web-based application providing financial advisors a streamlined system to trade fixed income assets.
- GEN2i, a portable device for recording engineering data in the field.
- OEConnection allows parts department employees at auto dealerships to quickly process parts orders sent by collision body shops, repair facilities, consumers and other auto dealerships.
BondWorks™ Advisor Platform

**Organization:**
BondDesk Group LLC is a leading fixed-income technology firm.

**Application:**
The BondWorks™ wealth management platform is a web-based application providing financial advisors a streamlined system to trade fixed income assets.

**Headquarters:** Mill Valley, CA (USA)

www.bonddeskgroup.com

**Design team:**
All design and usability efforts were handled internally. Two agencies were consulted with at the beginning stages of the product redesign: User Think and Momentum Design Lab.

**Members:**
In-house: Kamaryn Tanner, Product Manager; Ken Hoffman, Product Manager; Kelly Arkles, Team Lead, User Experience Team; Brad Taylor, Senior Designer/Frontend Developer; Joe Dakroub, Designer/Frontend Developer; Victoria Whitehorne; Technical Writer and Christina Matsoukis, Designer/Frontend Developer.

**OVERVIEW**

**Main Function:** The BondWorks™ Wealth Management Platform gives financial advisors the tools to carefully evaluate the best investment choices for each client they serve, efficiently execute trades and manage clients’ fixed income planning.

**Why it’s Unique:** BondWorks enables advisors to find suitable bonds and provide trade suggestions clients. Intuitive design and wizard tools makes it easy for infrequent users to buy or sell bonds confidently, and sophisticated features appeal to those advisors who are focused on managing large, complex fixed income portfolios for high net worth clients.

**The Organization:** BondDesk Group LLC is a leading financial technology company, providing innovative, enterprise-wide fixed income solutions to a community of over 2,000 investment firms including many of the top broker-dealers and buy-side institutional accounts in North America. BondDesk’s flagship application, the BondWorks™ Wealth Management Platform, enables over 25,000 financial advisors and asset managers to trade bonds with confidence and gather new assets.

**Design Philosophy:** A look, listen and learn approach is how the BondWorks team took their bond trading application from functional to fabulous. By taking a meticulous and sensitive look at how their users do the work of trading bonds, the BondWorks design team was able to streamline and optimize functions big and small across the bond trading lifecycle and across their application. In helping the users perform daily tasks with ease they ultimately helped their own bottom line, paving the way for increased trade volume and increased profits.

**USERS**
Approximately 25,000 financial advisors, who trade bonds on behalf of their clients, use the BondWorks application. The application is optimized to help make bond trading accessible for financial advisors who don’t necessarily trade bonds every day as well as to serve advisors whose primary business is built around fixed income.

The application has been designed to streamline processes for power users and give confidence-building tools to advisors for who trade less frequently. To accomplish this, BondWorks offers a variety of tools to facilitate research, workflow and trading, including the following:
• **BondWorks Ladder Wizard**: A guided workflow automates this historically complex, manual process.

• **Smart Search**: Advisors can use a powerful basic query screen, an expanded detailed query screen, access to the last 25 recent searches, and a natural language search engine.

• **Bond Comparison Tool**: Advisors can perform side-by-side comparisons of up to five bonds in an easy-to-read format.

• **Monitoring & Alerts**: BondWorks tracks the status of key events and allows advisors to create watch lists to monitor availability of bonds in the system.

• **Transparency into available offerings**: Advisors have maximum transparency into their available inventory with tools such as a scatter plot; access to the last 25 recent searches; rollovers with more bond detail; toggling between table and graph views; and grouping based on rating, asset class, and optionality.

• **Customizations**: Advisors can select their own default homepage; customize the alerts they receive; and add, delete, or remove columns from their search results.

• **Order and Bid/Offer Wanted Summary**: BondWorks provides users with a trade blotter that shows all open and recently placed orders. Color-coding makes it easy to check the status of each order.

### THE APPLICATION

BondWorks takes what could be an extremely complicated and dry application—an advanced retail fixed income trading system—and turns it into a helpful, focused tool that truly aids financial brokers and advisors.

The homepage focuses on what is undoubtedly the most important and common task for financial brokers and advisors: searching for bonds. Ancillary tasks are possible too via the top horizontal and left-side menus. While there is a lot on the homepage, designers subtly made the visual focus on the main task by adding a large amount of white space to each side of and below the search fields.

The bond market is a relatively complicated one; there are less than 10,000 companies listed on the stock exchanges, but about 4 million bonds available for trading. Choosing the best bonds thus requires detailed information and thorough research. Knowing this, designers created the **Search for Bonds** area on the homepage, which includes more specific search types, housed in the following tabs: **Criteria Search, Yield Search, State Search, Quick Picks**, and **Saved Searches**.

The default tab, **Criteria Search**, displays ten fields, many of which offer drop-down selection when applicable to expedite the task of searching. The remaining fields allow brokers to enter freeform text values for quantity, price, yield to worst, name, maturity, and coupon, as these are far less predictable choices. Each field question has a tool-tip link, which can easily be ignored or displayed, depending on the user’s needs.

An **Advanced Search** link leads to even more involved search options. Given the app’s user types, financial advisors and brokers who are working with bonds 1) all
day every day, and 2) rarely, this UI helps cater to both of these but doesn’t get in the way for either.

In addition to searching, the homepage provides messages in the lower left, so brokers can keep abreast of news, alerts, and announcements. The section below search criteria on the homepage summarizes the most recent orders, presenting the information of interest in the table, such as: order number, status, transaction, quantity, issue description, and account number. Advisors can sort this list and see, for example, all orders for a particular account, or all pending orders. This feature helps them keep abreast of current work and plan for the future.

The homepage is the entry screen for financial advisors that enables them to view and access key decision making elements for searching, messages, orders and tools.
BondWorks provides advisors with a powerful basic query screen, an expanded detailed query screen, access to the last 25 recent searches, and a natural language search engine.

The search results page displays very detailed content in a sortable table. Users can select the number of results to see on the page and use pagination buttons at the bottom to see more results pages.

In the upper right, users can toggle the data view from a table to a graph.

Additional buttons in the upper right allow users to access their 25 most recent searches, construct a new search, modify the current search, or save the search.
BondWorks provides advisors with maximum transparency into their available inventory with tools such as access to the last 25 recent searches; rollovers with more bond detail; toggling between table and graph views; and grouping based on rating, asset class, and optionality.

Clicking the graph button allows users to view the search results in a scatter plot, which is often easier to consider than tabular format. The graph gives brokers a better sense for how a bond stands in the market in relation to other bonds.

BondWorks provides a scatter plot view for analysis.

In the chart, users may choose a particularly interesting section, draw a box around it with the cursor, and add bonds within that section to a reminder list. They are then offered options regarding the watch list. This is just one more feature designers added to help advisors analyze bond activity.
BondWorks tracks the status of key events and allows advisors to create watch lists to monitor availability of bonds in the system.

The user can switch back from the scatter plot to the tabular view by clicking the Table button in the upper right. Advisors may then choose bonds they are most interested in from the table and compare them. Clicking the checkbox to the left of the result enables the Compare button in the upper left. This opens a dialog box that lists the selected bonds and compares them, displaying green up and red down arrows to indicate relative performance. Designers anticipated the user’s next steps by enabling the advisor to buy a bond from here, or print the comparison table. Closing the dialog box returns the user to the search results page.

Advisors may perform side-by-side comparisons of up to five bonds in an easy-to-read format.
After buying a bond, an advisor may choose to check its status to ensure the order is in the system. The *Order & Bid/ Offer Wanted Summary* (sometimes called the “Blotter”) page gives details about the transactions which have posted or are in progress. The row just above the table gives a snapshot of activity, including the total number of orders, numbers pending, filled, rejected, cancelled, and expired. This table affords action as well as passive reading. It’s not just for sorting as advisors can cancel pending orders listed in the table.

BondWorks provides users a trade blotter that shows all open and recently placed orders. Color-coding makes it easy to highlight the status of each order.

Designers wanted users to be able to access what they need from anywhere in the UI without disrupting search results, charts, or currently displayed data. So they implemented the EasyNav Toolbar, which appears in the upper right corner of every page in the system. The toolbar includes icons for tasks such as: reading system messages, reviewing pending orders, checking for watch list alerts, opening a list of favorite screens, and browsing the list of system-defined short cuts. Designers chose to invoke the action on click instead of on hover because they were concerned about data disappearing when the mouse was moved. Advisors deal with a large amount of data so an error here could be a disaster. The EasyNav Toolbar provides access to important system information and, more importantly, allows users to see that information without leaving the screen they are currently using.
BondWorks provides users with the EasyNav Toolbar in the upper right-hand corner of every screen in the system. With a single mouse click advisors can read system messages, review pending orders, check for watch list alerts, open a list of favorite screens, or browse the list of system-defined short cuts. Not only does the EasyNav Toolbar provide access to important system information, it allows users to see that information without leaving the screen they are currently using.

Another task brokers need to do is build a bond ladder. Historically this was a very complex and manual process. Thus, BondDesk designed a wizard to ease advisors into building ladders for clients. The wizard takes users through the process, step-by-step, asking questions about tax, bond type, credit rating, and ladder structure. Originally designers created the wizard to give confidence to users who don’t consider themselves bond experts. It worked, but the “big bond guys” use the wizard as well. It makes sense because the wizard consolidates several pieces of disparate information easily and quickly.

In the wizard, each step follows important guidelines about good wizard design, including the following:

- **Heading** at the top naming the reason for the wizard itself, in this case, *Ladder Builder*.

---

3 A bond ladder, according to investopedia.com is “A portfolio of fixed-income securities in which each security has a significantly different maturity date. The purpose of purchasing several smaller bonds with different maturity dates rather than one large bond with a single maturity date is to minimize interest-rate risk and to increase liquidity. In a bond ladder, the bonds’ maturity dates are evenly spaced across several months or several years so that the bonds are maturing and the proceeds are being reinvested at regular intervals. The more liquidity an investor needs, the closer together his bond maturities should be.” BondDesk designed a wizard to ease advisors into building ladders for clients lessening the need for liaison traders to perform this function manually. The wizard approach provides an easy to use automated tool in which a user can find bonds to include in a ladder based upon specific client and bond characteristics.

http://www.investopedia.com/terms/b/bondladder.asp#ixzz1qWstiOjC
• **Clear status bar** at the top which includes how long the process is; the actual steps involved; where the user has been in the process, is currently, and where else he will go. In this case visited steps are green, current is blue, and future are light grey.

• **Page/task name** and heading, noting the main things the user will do on that particular screen.

• **Clear questions**, labels, and fields.

• **Clear ‘advance’ and ‘back’ options**, in this case *Previous Step* and *Next Step* buttons in the lower right.

• Plus, of course, a way to **cancel out** of the process in case the user entered the wizard by mistake.

![Image of a wizard screen](image)

**Step 1. Tax selections**
Step 2. Bond Type selections.

Step 3. Credit rating selections.
Step 4. Ladder structure selections.

Once the wizard questions are completed, the system selects bonds that meet the entered criteria and organizes them by year. At this point, the advisor can get a sense of the cash flow the client might receive, and generate a PDF report to give to the client.

(Note that even though BondWorks is a winner in the complex applications category, the need for an advisor — the primary user — to interface with clients means that it also has advanced workflow elements. The clients can be considered secondary users in some ways and BondWorks needs to support this multi-person workflow. For more on workflow, see the category of workflow application winners, starting on page 143.)
Step 5. Results give brokers a sense of what the client’s cash flow could be.

On the results page, brokers may buy bonds by clicking the Buy button to the left of each item.

Step 6. At the end, brokers can move forward and buy bonds, using the BUY buttons on the left in the table. The Ladder Builder wizard lessens the need for liaison traders to perform this function manually. The wizard approach provides an easy-to-use automated tool through which a user can find bonds to include in a ladder based upon specific client and bond characteristics.
Another feature that helps new and advanced users is the *Did You Know* carousel. This carousel provides an easy to use and easy to read help option. The beauty of this interface element is it makes people aware of new functionality in the app. It also provides quick access to documentation on more complex workflow tools. Context-sensitive online help is also available to assist users.

*Did You Know Carousel*: In addition to standard online help and context sensitive help, BondWorks uses the *Did You Know* carousel to quickly highlight new features added during major releases and to highlight more advanced workflow tools.

Any user can be aided by good customization features; BondWorks provides the ability for users to change their profile in *Settings*, select a default homepage, choose a default search tab, and select the number of search results to display per page. There are several other configuration options which help expedite work and meet the needs of individual advisors, such as the ability to choose which alerts to receive and which columns display in the search results.
Users can change profile settings to select a default homepage and display a particular search tab by default, among other things.

With a good mixture of inductive and deductive UI elements, BondWorks leads experienced and less experienced advisors and brokers through the processes of researching, buying, and communicating about bonds. An ingenious dashboard and a thorough wizard, among other features, increase user confidence and productivity. Finally, anticipating the user’s next step, focusing page priority, and embedding needed albeit unexpected features makes this app a world-class winner.

**DESIGN PROCESS**

**Usability Methods**

The most recent product redesign resulted from two primary drivers:

- The application was looking a bit dated and it was time for a visual refresh, and
- The company wanted to streamline many of the bond trading processes to help traders increase their volume, and thus increase the company’s bottom line

The overall goal was to redesign the application from the ground up. “We really wanted to do a re-design from the ground level,” says Product Manager Kamaryn Tanner, “To stop and think about what tasks brokers are trying to accomplish we wanted to go back to fundamentals. We didn’t want to just put a pretty face on what was already there. We wanted to re-do the information architecture throughout the application so we started at the task level.”

This required a number of research methods so the team could get a complete picture of what they needed to do to greatly improve and enhance the application. These included:
Watching users work: Because it’s difficult to get in front of actual brokers to do task-based testing, the design team did the next-best thing, they spent time sitting with brokers as they went about their daily work. That exercise proved pivotal in learning where the roadblocks were in the application and where the design could support streamlining of frequently used or difficult processes. “We actually talked these guys into letting us sit at their desk with them,” says Tanner. “And as they took client calls they let us listen in. We also watched what they did to sell these bonds and use the software.” These sessions helped the team identify where and when the advisors turned to paper and pen as part of their regular process. Those paper moments highlighted opportunities for improvement in the application, to provide functionality to replace manual processes. “For example, they would write on Post-it Notes and stick them on their monitors,” she says. “We got a good flavor for what their day was like and what they were having trouble accomplishing using their current software.”

Another key finding from these observation sessions was the realization that financial advisors work in an environment where interruption is the only constant. The team took this reality to heart when designing the new version of the application, building in support mechanisms so brokers can be interrupted and still be productive. Observing users also helped the team map out important processes, such as the steps needed to build a bond ladder wizard.

The process flow for the bond ladder wizard enabled the design team to identify all elements needed before coding took place.
• **Focus groups:** The team enlisted the help of an outside firm to conduct traditional two-way mirror focus groups. These sessions did not focus on how the clients use BondWorks to do their work, but rather helped inform the team about the overall challenges of what it’s like to work with clients to sell investments and where bonds fit into that investment big picture.

• **Sitting in on support calls:** Because Tanner had led many of the trainings for the BondWorks application prior to the redesign, the team was able to glean user pain points from the things she heard in those sessions. “It became clear that when I had to explain a little bit too much about something, we had a usability issue,” she says.

• **Checking user stats:** Doing an analysis of the user logs was an ongoing part of the design process. The BondWorks team is fortunate in this area as they can literally see everything their users do in the application by studying user logs. Analyzing this data truly informed some of the decisions the team made about specific functionality.

  The design of the Smart Search is a good example of this. Searching is one of the most common things users do in BondWorks. “In the month of November, for example, 180,000 searches were performed,” says Tanner. “It’s by far and away the most common activity that’s done so we spent a lot of time looking at the logs to determine how users search so we could help them in that area.” It’s important that the application deliver finely tuned search results, otherwise the user will be overwhelmed with results. “Someone who searches for ‘bonds’ and gets 5,000 results is overwhelmed and is quickly out of there [the application].” So the team focused on optimizing the search functionality to serve the less frequent user.

  Bonds have many attributes that are important when choosing which one to purchase. The original application presented approximately 85 of those attributes on the search results screen. After studying the search logs, the team realized that nearly three quarters of all users were only searching on some combination of ten attributes. The team simplified the search screen in the new interface to focus on these ten key attributes. “That helps us present a search function that is a lot less intimidating,” says Tanner.

• **Surveys:** Because the team employed beta users and pilot users while they were designing and building the new application, they had a pool of users at the ready who had knowledge of how the new application compared to the old. At the end of the trial period, these users completed surveys, rating the new product against the old. The new application scored high, with over 90% of respondents saying the new app was better. The respondents were also asked about specific features, ease of use, etc., to gauge specific details to support their overall opinions.
• **Paper prototype testing:** The first round of low-fidelity wireframes were simple paper prototypes where the design team asked people to be potential users and try to accomplish tasks with the prototypes. As they made decisions and clicked on the screens, they were encouraged to think out loud and share their thought process. These early wireframes served as a good starting point for the team to determine user flows and information architecture. The team also used these early sketches to identify how important features would work.

• **Wireframes:** Higher-fidelity wireframes were developed later in the process to solidify interface design elements and properly allocate screen real estate. “We learned a lot, especially during that beta phase,” says Tanner. A good example was an early wireframe comp that had a lot of additional space in the header that didn’t need to be there. In place of the white space, the team decided to pull other elements up the page, to give them more visual prominence and to distinguish the header space in the application from the various tool bars and browser elements that are often present when the BondWorks application is presented inside a company’s intranet window.

Another finding realized during wireframe testing was that users did not see the Smart Search box. By maneuvering elements around the page in later designs, the team was able to give the feature more prominence and make it more noticeable.
Used mainly as a guidance model, the team’s design process was based on achieving as many of the deliverables seen in this screen as possible. This model outlines what was needed to achieve the new design.
This is an example of a low-fidelity wireframe used during the early task analysis phase.

This is a high-fidelity design comp created during the design process to fine tune a feature.
• **High-fidelity prototypes:** During the beta phase, the design team also introduced high-fidelity, interactive prototypes (or simulations) for the order ticket flow, primarily using iRise (a sophisticated prototyping tool). “This was an important step,” says Kelly Arkles, User Interface Team Leader. “It allowed for us to better visualize—for both developers and users—the overall experience and process flow. It helped us find potential holes in the design and improve the experience prior to any code being created. For all new functionality and customized order ticket flow, simulations are now required as it has proven to enhance the design process, communications, and overall user experience while also eliminating unnecessary rework.”

**LESSONS LEARNED**

BondWorks team members share a few of the lessons learned from the redesign effort:

• **Know your audience.** The product team made a technology decision upfront during the redesign: to provide browser support for Internet Explorer 7 and above. That single choice presented a world of challenges and a significant amount of work when they found themselves forced to go back and retrofit the app for IE6. “We made a decision that we were only going to support IE7 and IE8,” says Tanner. “That actually affected a lot of things we did and the way we did them. That was all fine and good until we needed to bring on a very large, new client that was using IE6. In retrospect, I think we were maybe just a little too aggressive in assuming that everybody would have IE7.” The simple truth is that clients in the financial space are slower than other industries to adopt new technology; assuming they could uniformly eliminate an antiquated browser technology from their product was unwise and costly.

• **Take a product approach.** Though a long-time coming, the team decided that together with the redesign they would move BondWorks from a client-based application to a product-based application. This decision has saved the company both time and money and will pay off long into the future. “What we had before was several custom versions of our applications,” says Tanner, “The amount of custom code we had was very hard to maintain and was definitely a testing issue for us. We had to expend an awful lot of resources every time we did any work because many clients’ code was different.” Going in a different, more unified direction made sense, but was not universally supported — at least at first. “Many believed our value was in customizing systems so we spent a number of years demonstrating that we could provide even greater value through innovation in the marketplace and a deeper understanding of our clients’ business challenges.” A product approach has not only saved the team countless cycles of reconciling code changes specific to custom client versions, but it has resulted in a far better product. “We designed a better overall application than any of the customized versions our clients had,” she says.
• **Know your design landscape.** When designing a screen environment that (often) lives inside of another screen wrapper (as is the case with intranet apps such as BondWorks) it’s important to really, truly understand how the window-inside-a-window affects design size and the use of screen elements. “What we had planned and designed for was 1024 x 768. But because BondWorks can be pulled into a browser through various intranets and applications, we couldn’t always accurately determine how much toolbar space or screen real estate is taken up by that portal,” says Tanner. “And that wasn’t something that we found until into the beta stages when a lot of the design was determined. We had to determine how the elements would fit in the screens at a specific size.” The solution they found to accommodate the various unknowns of client environments (and still maintain the integrity of their own application design) was to establish a viewport size and make that a required condition of incorporating BondWorks into a client’s portal environment. “What it came down to is that we require the viewport size to be 1000 pixels wide by 625 pixels for our application and that doesn’t really affect the functionality. It just affects the display,” she says.

Arkles adds, “After we made this discovery, simulations for customized client order tickets have improved. We ensure that each simulation reflects the viewport size so that clients can visualize how their requests will affect the user’s view and interaction with the order ticket flow prior to coding and the UAT process. This also helps us with new functionality development as we can see how the worst case scenario with viewport size affects overall layout of screen elements.”

• **Choose technology wisely.** Sometimes a simple technology choice can lead to unplanned dependencies, as was the case with the JavaScript library the application is built upon. “We chose a JavaScript library that, at that time, was not at a mature level,” says Arkles. “We were closely tied to the library releases and upgrades. The upgrade process wasn’t as smooth as it could have been. Now it’s a very mature library. It’s definitely gotten better but taking the approach proved to be a very big challenge. Also, our developers had little experience using the JavaScript library. There was a definite learning curve to get to a more expert programming level.”
TEAM

“Hunting around for information really annoys advisers.” This quote identifies the underlying principle for the application—large amounts of information need to be easy to find and use. This basic idea combined with key usability concepts drove the product and design teams to create an application that helps users get their job done rather than get in their way. When suggesting ideas, all design contributors ask the question, “Does this make the user’s job easier or harder?”

Pictured (left to right): BondDesk Product Management Team members Kamaryn Tanner and Ken Hoffman.
Pictured (left to right): BondDesk UX Team members Kelly Arkles, Brad Taylor, Joe Dakroub, Victoria Whitehorne and Christina Matsoukis.
GEN2i

Organization:
Hottinger Baldwin Messtechnik GmbH (HBM)

Application:
Portable device for recording engineering data in the field

Headquarters:
Darmstadt, Germany

www.hbm.com

Design team:
The GEN2i application was designed in-house by the company’s software group, in the Netherlands. The team was comprised of six engineers and one interaction designer.

Members:
In-house: Klaus Lang, Product Manager GEN2i; Edwin Schild, Software Development Group Leader; Marco Dijkmans, Software Engineer; Coen Van Dongen, Software Project Lead Engineer; Panagiotis Georgiadis, Software Engineer and Cees de Vries, Interaction Designer

OVERVIEW

Main Function: The goal of the GEN2i project was to create a touch screen operation for a portable measurement system (comparable to a digital storage oscilloscope.) The primary function of the application is to control the system, acquire data, store data and provide basic analysis and report generation, both in the field and in the lab. The measurement activities of this instrument can range from airplanes to space shuttles to cars to heating: whatever you can imagine, as long as it produces physical signals. It’s designed to record all kinds of physical phenomena, such as temperature, pressure, power, force, etc. to help engineers diagnose problems. If something produces a physical signal GEN2i can measure it. So if an engineer in the field can register all that information at the same time, he can likely see what’s going on, what the relationship is between the various components, and deduce what’s wrong.

Why it’s unique: The GEN2i application interface is remarkable for its flat structure. The engineers have achieved a tight focus on core features, all accessible from one screen and have made effective use of space in the interface. Though they had to design for many controls, they still managed to reserve most of the screen real estate for the results display. And to top it all off they’ve made it possible to completely control the interface with one hand because the apps’ users are often in situations where they’re using their hands for other things. In fact, users can control GEN2i with just one finger.

The organization: The Hottinger Baldwin Messtechnik (HBM) mission is to help its customers optimize the product life cycle. HBM is a business-to-business company with a product range that covers sensors, transducers, strain gages, amplifiers, and data acquisition systems as well as sophisticated software for testing and analysis of equipment. The potential fields of application for its products can be found in every branch of engineering and industry in both virtual and physical test and measurement.

HBM has approximately 1,500 employees worldwide and is a wholly owned subsidiary of Spectris plc, a group specialized in precision instrumentation and electronic controls. HBM has 27 subsidiaries and sales offices in Europe, America, and Asia. HBM also has representatives in another 40 countries around the world. In addition to headquarters in Darmstadt (Germany), other HBM production facilities are located in Marlborough (USA) and Suzhou (China.)
The group that built GEN2i, HBM Netherlands BV, is a small (approximately 20 employees) engineering group, part of the Germany-based HBM company (www.hbm.com.)

**Design Philosophy:** The team behind GEN2i have achieved a nice, tight interface solution to solve a very complex problem. When designing for expert users it’s sometimes difficult to simultaneously accommodate new users as well. The application has done both well, while keeping the focus firmly on the needs of the experts. New users will have a steeper learning curve with GEN2i but experienced users will appreciate the quick access to controls without deep menus to wade through.

**USERS**

The GEN2i instrument has been sold to approximately 500 users worldwide since its debut in the market a year and-a-half ago.

The interface was designed specifically for engineers who do troubleshooting. These users come from two primary groups: technicians in the field and engineers in the laboratory:

- **The technicians** have a measurement job to do, for example trouble shooting an engine (ranging from a simple motor to a space shuttle) or any other device. The technicians need to acquire data easily so they can repair something that is broken.

- **Engineers** use the instrument to measure, analyze, archive and report.

The users are often working in a cramped, hectic environment with a lot of noise so the GEN2i interface is critical to these users who need to get to what they need without taking the time to read the user manual.

**THE APPLICATION**

The GEN2i instrument is designed for particular types of users: smart people who know what they want. They care more about the data than they do the app it is encased in. Designers recognize this and present a UI that gets out of the user’s way. A dashboard is in order given the user type and what they need for this app: status information and the ability to record and watch data interactions. Understanding that the main section with the data meters is very important, designers dedicated most of the UI to this, but still offer other toolbars and panels so users may manipulate data. They also, when possible, economize on space by making areas multipurpose, replacing content with new and different content to make the most use of the dashboard real estate.

Just after startup the main application shows all functionality, and the dashboard UI groups the information so the users may scan and pick out what they want. The data begins showing and collecting right away so users can start collecting data. Each section in the dashboard uses backgrounds and lines to set themselves off from the main content area and from each other. In general, the action buttons are accessible but take little space away from the main content area. Given that even on a this small, portable measurement system you may view up to 64 channels, real estate in the content area is at a premium. The data the users care about takes up the bulk of the dashboard, the center section. This area is used to play live data or play back recorded data.
The top horizontal bar with the labels Max and mV presents the real-time values of the channels. These meters correlate with the channels presented in the main content area in three ways, including:

1) name, for example, A1 is A1 in both areas,
2) color, A1 is yellow both in the top horizontal area and on the left, and
3) position, A1 is first on the left in the horizontal bar and first from the top in the vertical bar.

These three criteria can truly help the users orient themselves in a busy dashboard environment.

Users may add and remove, and drag and drop channels within the main content area, and tap on a channel or meter to configure it. When an object is dragged an information balloon pops up to give user feedback. Once the user has the configuration she desires she may save it via the Save setup button, which is third from the top in the left-most toolbar.

Dragging an object opens a yellow information balloon for feedback.

If a user is sitting in front of the instrument and is left-handed, then she is always obscuring the more important parts of the instrument. Switching to left-handed reverses the order of the vertical panels—everything switches around.
In addition to small customization, the app allows users to switch from righty to lefty view.
Users may switch to the lefty view, which swaps the panels, the right for the left.

Many times engineers want to review the activity that has been inspected, but since data consumes space and recording can make the system lag, users may choose to record as they see fit. Record and playback options appear in the upper right of the right rail. The red arrow button with the word *Record* under it makes it easy to start a recording, and the green square *Stop* button to stop it. The *trigger* icon allows users to generate an event that finalizes specific types of recordings. This enables people to capture the events they care most about without having to record, or review recordings from, 24/7.

And since recording is one of the most important tasks and are used to review the recorded data, the standard play and replay controls appear in the bottom, horizontal bar of the dashboard.

The blue rotary knob in the upper right area was originally a physical knob designed to scroll through the data and set the sample rate. But in the final version the user can swipe through the data and use the rotary knob icon to scroll through the sample rate values. As it does it makes a clicking sound to pull the metaphor fully through. This, the designer says, makes people smile when he demos it. The values above the knob change as it scrolls. Those who don’t want to use the knob may click a button, or select from preset values.
The lower right section displays a blue bar with arrows. This shows the selected channel, and the arrows allow users to scroll through and choose a different channel. The set of mostly blue-banded buttons displays settings for that channel, for example: `mode`, `coupling`, `input`, `span`, `offset`, `scaling`, `filter`, and more.

The main application after startup makes all functionality readily available (except for some commands on the left-hand side.) The top bar is status feedback. The main content area houses a selection of available channels and can be configured by drag and drop.

The section in the lower right rail changes depending on what is selected. Replacing content in the same area is one way to save space in a dashboard, or any environment. The user may select one of the settings buttons for the selected channel, at which point the blue square buttons representing settings go away and are replaced with information related to that selected setting and channel.
When a user taps a settings button in the settings panel (lower-right group of the main window) the panel rotates to show the options for the selected setting. The same is true for the acquisition control group. To keep operation straightforward and minimize the user’s cognitive load you can modify only one setting at a time.

Some of the commands in the left-hand toolbar (or menu) offer are a top-level item segueing to more options. For items that have more options available, the commands fold out. This progressive disclosure is one more way the designers refrained from taking space away from the content area. To ensure that people can close the fold-out item, designers added a red “x” in the upper right.
When the user taps a command which has more options on the left-hand side, an options list will fold out, judiciously hiding commands until users ask for them. (This toolbar is the only location that may require two taps to get to the information.)

A user may want to look at a lot of data at once in numeric rather than graphical meter format (sheet-based rather than a channel-based.) They may view all settings (organized on the x-axis) by all channels (organized on the y-axis.) The tabular format allows this channel-based viewing, and users may set and compare items. With so many of each they cannot all fit on the small display, so users may scroll vertically as well as horizontally when needed. The unconventional scroll control on the right and bottom offers arrows to move, and clickable dots which also convey status.

Users may read and compare these items, and it is also possible to easily copy the settings in one channel to other channels via this matrix. Buttons below the table, *copy to*, *copy*, and *paste* make this a snap.
This matrix allows users to see an overview of all settings and channels (up to 64,) and/or modify settings. Users may easily copy settings from one channel to other channels.

The underlying software behind the app is called Perception, which is a Windows program, and users may swap between it and the app. Using Perception users may access more functions to manipulate their data.
The Perception program (an underlying software for the app) allows for much more functionality for the users.

At first glance this application appears to be quite complicated, but after a short time the target users become accustomed to it. Each section has a compartment that people can easily scan to, and controls for manipulating it are handy. Data is shown real-time or can be recorded and edited. Recordings may also be exported to ensure that data can be best analyzed and shared, no matter the computer.
The users may export the data in a variety of formats.

**DESIGN PROCESS**

**Usability Methods**

The first thing you have to know about the design process for GEN2i is that the design team was designing for a conservative market so the target audience was looking for evolution not revolution from the existing product.

“It’s a very conservative market where we are operating in,” says Interaction Designer Cees de Vries, “and that means that you can’t revolutionize, you can only make an evolution of what you’re doing. So we had to put most of the legacy stuff already in the new instrument, but also find a way of making it easier for people to use.”

The previous version of the instrument was a result of the knowledge the team had gained over the years from designing other software programs. This collective
knowledge heavily influenced the mindset of the engineers when they embarked on this project. “The more you create these kind of software packages, the more questions you get over the years,” says de Vries. “And then you already have a kind of image of what it should look like and what the feature set would be, in a minimalistic way.”

One of the clear mandates in moving from the existing instrument to the new design was to improve the information architecture and optimize the operation flow.

For example, “When using the knobs in the old version we had to go through a multi-level menus, so if you search for a certain function which that could take you three, four, or five steps, basically,” he says.

With a poor IA structure, users often ended up in the wrong menu and that was something the team especially wanted to improve. The team accomplished their goals through a number of design activities, focusing on optimizing the operation flow. These included the following preliminary activities:

- **Heuristic evaluations**: The team conducts heuristic evaluations on a regular basis with Product Marketing and other stakeholders. Information from these regular meetings informed many of the decisions the team made with the new design.

- **Books, reports, and other resources**: Learning is a continuous activity for the design team. They frequently consult books, report and other user-focused research to improve their knowledge and apply that learning to the products they design. And in this case, books and research played a particularly important part in their decision-making. “What really contributed to the decisions was what I read in books,” says de Vries. “That seems a little bit simplistic but I really like what Tufte is doing. I came up with the flat interface, and the flat interface was basically given by the iPhone, but even more so by the books from Tufte where he explains why you should have flat interface and what the possibilities are. And that influenced the whole design from start to finish.”

- **Sitting in on support training**: During the pre-release period the team had an opportunity to listen in on support training and gather user feedback.

- **User Comments**: The company regularly collects user comments and incorporates that feedback into product releases. These comments are collected through a database called TestTrack.

During the design project the team used wireframes and beta testing to ensure that the engineers had a solid design direction to work from and that the product was well tested prior to launch.

- **Wireframes**: de Vries created prototypes in Photoshop to give the software engineers a guide for designing the instrument. The wireframes ensured that the engineers didn’t have to figure out the basics.
• **Updating the IA:** The team used the existing information architecture as a jumping off point for their work. “What I did was, I took the old instruments, we are replacing not only the one instrument but also two other ones,” says de Vries. “And then I went through their information architecture and just by comparing those two to three models, to see what were the typical overlays, I figured out what would fit better.”

• **Beta testing:** Beta testing was conducted with a small group of sales and marketing staff. These participants had never used touch interface instruments before and they weren’t familiar with this new product. The design team thought it would be beneficial for them to have firsthand knowledge of the product before it was launched so they were a good group to test with. The testing was done a little earlier than traditional beta testing, with approximately 50 percent of the software working when the tests were conducted. And the testing was very informal. Participants just given the freedom to use the device and provide feedback with very little direction or instruction.

• **Usability testing of design prototypes:** The team tested with internal users (normally not recommended) but their testing had an interesting twist. They used the hardware engineers to test the software features. And for these engineers testing software was “a whole new world” that they were unfamiliar with. “It’s a good idea,” says de Vries, “to have them come out of their hardware knowledge and go into the software.”

The testers were presented with different feature sets and one group would test one version and the other group an alternate version. Then the design team could see what went wrong and where. These sessions were informal, more of a discussion so the design team could not only observe how the participants interacted with the controls, but also hear how they were thinking and how the system did/didn’t match their own ideas of how it should work.
The illustrations below show how the interface design progressed from concept to execution:

The settings panel. Initially there was no panel.

Later, the design evolved from general to detailed.
The acquisition panel also evolved from general to detailed. For this part of the design, the team borrowed a number of components from the underlying Perception software, such as the rotary knob, pixel display and gauges (colored progress bars.) Note: All controls are “handmade”.

Marketing and sales team beta testing the GEN2i.

Development Approach

The team employed a modified Agile development method, where the team introduced new features every few weeks (rather than every week in a traditional
Agile environment.) The team sat together, set goals for a one three week period, identified roadblocks, etc. and kept up this schedule throughout.

They worked at a very fast pace, with something happening every day—daily builds—so the team had a new functional version of the software to review and test nearly every day. And that’s a big plus for improving the product.

“You can look at the new features, the new implementations and decide right on if it’s going on the right direction or the wrong direction,” says de Vries. “So you have much faster turnaround with respect to implementation of features, the usability of features, and you just try it out as soon as you get a new version and you see what’s going on, and then try it out, if it works, if it doesn’t work.”

But that pace can be exhausting as it is productive.

“It’s a little bit more what I would say hectic,” he says. It’s very, very fast programming.”

It may have been a lot of work, but that work paid off.

**Timeline**

The project took approximately one year from feasibility study to release. They started with a lot of background information because the legacy product was used as a foundational feature set. Then, that feature set was prioritized to accommodate the tight timeline. Features were added and tested nearly every week until the release date. Thought the project took a full year, the development cycle was only three months and is outlined as follows:
## PROJECT DEVELOPMENT TIMELINE

<table>
<thead>
<tr>
<th>Milestone Date</th>
<th>Hardware</th>
<th>Software</th>
<th>UX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q2 2009</td>
<td>• Feasibility study</td>
<td></td>
<td>• Design strategy</td>
</tr>
<tr>
<td></td>
<td>design document</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q3 2009</td>
<td>• Box prototype 0</td>
<td></td>
<td>• Profile</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Task analysis</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Information architecture</td>
</tr>
<tr>
<td>Q4 2009</td>
<td>• Box prototype 1</td>
<td>Implementation</td>
<td>• GUI Design</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• GUI Implementation</td>
</tr>
<tr>
<td>Q1 2010</td>
<td>• Box 0-series</td>
<td>• Release candidate</td>
<td>• Finalize</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Release</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Shipment</td>
<td></td>
</tr>
</tbody>
</table>

## LESSONS LEARNED

De Vries shares a few lessons learned from the redesign effort:

- **Do IA up front.** “I’m not completely satisfied with the IA, because we did not do any research on that part. For other projects I did research and then you see how relatively easy it can be. You can’t get everything in perfect but you can get 80 percent.”

- **Do user testing early.** “As long as it’s everybody’s personal opinion you will never get a verdict, not for all features, but for features that are important or that you can’t compromise I suggest user testing. I always find it very surprising to see what people struggle with.”

- **Test with the people you have.** “I use in-house [testers] depending on the situation, because it’s hard to find people in this specific niche area. They [the in-house testers] always like it; it’s only an hour they don’t need to work.”

- **Get buy-in from marketing and sales.** “With this instrument, which is relatively new, why should they try to sell this thing which is a little bit risky. It’s their bonus on the line. If you have buy-in from these guys and they like it you’re all set. These are the guys that need to prove that my design is right and not a shipwreck. These are the guys that need to sell the instrument and get a bonus out of it. It is like features: a feature not found is a feature that’s not there. When I design a perfect instrument and it doesn’t get sold it’s worthless.”
• **Don’t skip the testing.** “Do your card sorting or other technique to get the information architecture right. I did not do this for this project and that lead to numerous discussions without a defined outcome. Do user testing as soon as you have wireframes or prototypes. This is also to avoid long discussions with stakeholders about the implementation of a feature.”

**TEAM**

The software group that designed the application is based in the Netherlands is comprised of six engineers and one interaction designer. The team’s direct manager, Peter Ackermans also played a pivotal part in the project by giving the design team “room to breathe.”

Cees de Vries is currently a Usability Manager within HBM; during the development of the GEN2i interface, he was HBM’s Interaction Designer. de Vries has a bachelor’s degree in Computer Science. He originally started as a hardware/software development engineer, and worked his way through various job roles. Five years ago he started to develop himself in the field of design and usability. His personal ‘hero’ is Edward Tufte, who inspired him to develop the interface as a flat interface where information becomes the interface. His favorite quote is “Don’t make me think,” by Steven Krug.

Pictured: Members of the application design team (left to right) Marco Dijkmans, Edwin Schild, Cees de Vries and Coen van Dongen.
OEConnection

Organization:
OEConnection LLC is a strategic venture chartered by Ford, GM, and Snap-On Business Solutions in 2001 to leverage and implement technology solutions that help automotive dealers manage their parts inventory and sell more OEM parts.

Application:
OEConnection provides a user-centric workflow that allows parts department employees at auto dealership quickly process parts orders sent by collision body shops, repair facilities, consumers and other auto dealerships.

Headquarters: Richfield, OH (USA)

www.oeconnection.com

OVERVIEW

Main Function: As the automotive industry’s largest original equipment (OE) parts marketplace, OEConnection connects more than 32,000 North American buyers and sellers to seamlessly market, manage and move an average of over 6 million original equipment parts transactions monthly. OEConnection is not just a single application, but also a gateway to five integrated technology solutions that facilitate various aspects of the parts procurement and inventory management workflow for automotive repair shops and parts dealers.

Why it’s Unique: When serving a user population that will move back and forth between related and linked applications it’s important to create a frictionless experience. That level of seamless integration between products is precisely what OEConnection achieves. Using the suite of products that combine to create the OEConnection experience is fast and simple, and from a business perspective the technology is also scalable.

The Organization: OEConnection is both the technology and the organization behind the platform. It is an award-winning technology leader and innovator of original equipment (OE) replacement parts solutions. OEConnection serves the automotive and heavy-duty truck markets and beyond. OEConnection equips some of the world’s largest equipment manufacturers, their franchised dealers/distributors, and their customers with timely and accurate online parts marketing, procurement, and wholesale management solutions.

Design Philosophy: Workers in parts departments work in chaotic environments filled with competing priorities and a world of distractions. One of the core goals of OEConnection is to make sure these users can seamlessly attend to all the things competing for their attention while still allow them to complete the task at hand: procuring parts.
**USERS**

OEConnection serves 32,000 users who primarily work in the parts departments of automotive dealerships and collision and repair facilities representing 18 automotive manufacturers across North America.

What users *do* with the tool depends somewhat on which application they are working with at any given time as they move back and forth across the platform, but the primary tasks are looking for and ordering parts and fulfilling parts orders.

**THE APPLICATION**

It’s utterly unbelievable how helpful OEConnection is. It is testament to how studying user tasks and mental models can inform an amazing application, in this case, for automotive dealers managing their parts business. These professionals, if nothing else, need to keep a thumb on the status of their parts and parts orders. Help with this begins on the OEConnection home screen, which enables users to track, review, and search easily and quickly. Each order is listed in a clear table with various important criteria as column headers, including: status, order type, submitted by, vehicle description, salesperson, and last modified. Pagination under the table allows users to see more orders, and the search box above the table enables users to search for something specific they recall.

Users may show or hide an additional column for an order’s Transaction ID, a unique ID assigned to the order by the software that the Customer Care Department uses to assist end users with troubleshooting an order.

The common order overview screen is the first screen the user would see upon log in.
On an order details page, the user may review many different fine points regarding a specific order, and he may also perform a number of actions on the order. The VIN number, car year and model, and shop are all displayed at the top of the page, clearly denoting which car the information on the rest of the page concerns. The horizontal list of links below these headings allows the user to take any number of actions on the particular order, including: save, print (which displays a list or reports to print along with a quick print of the most commonly used reports, as well as an advanced print option that mimics the print dialog found in a previous version of the app,) add part, locate, view returns, cancel order, respond to buyer (which transmits any edits performed by the dealer back to the buyer,) and submit/commit (which completes the order by walking the dealer through a series of verification steps depending on the program promotions he is eligible for.) Having these tasks all in one place greatly expedites the dealer’s work and increases his confidence level.

The content area of the page is basically divided into three sections—Conversion Opportunities, Buyer Requests, and Response to Buyer—and are expandable, but collapsing by default smartly makes good use of screen real estate. The plus sign buttons to the right are obviously for expanding the section. Within the three sections in the content area the information is about the part, program, shop cost, dealer cost, and part catalog information, among other things. Designers packed the area with content, but laid it out in such a way that it is understandable and manageable.

For each of the three sections a number appears next to the group heading. This assists the user in understanding the work queue remaining on this order, and he may quickly assess whether a part needs to be fulfilled.

The Conversion Opportunities section shows which parts on the order were not specified as Original Equipment Manufacturing (OEM) and were intended for a third party to fulfill. It’s important to keep the OEM information up to date, as the parts represent upselling opportunities; that is if the parts are converted to the OEM version of the part.

The Buyer Requests section shows the parts that were specifically requested by the buyer to be filled by the dealer with genuine parts.

The last section, Response to Buyer, comprises the list of parts that the dealer has verified to be correct for the vehicle being serviced, and presents the required information to price the part (as the seller wouldn’t want to spend time pricing a part they won’t actually be selling.) The Buyer Request and Response to Buyer sections of this workflow are also common to other transaction based workflows.

Users may access notes and photos that were sent with the order, as well as another piece of important content, the RO number, which the shop provides. If a customer calls with this number it makes the order easier to locate.
The order details page displays the pertinent information about a particular order, and enables the dealer to take several important and common actions.

The table and page layouts in OEConnection are well prioritized and limit clutter. Headings and sections help with this, as does consistency. Designers understand the importance of knowledge transference and flow throughout the app. Knowing that users acclimate more quickly and successfully in UIs that have some semblance of consistency, the team created a persistent menu across the top, consistent headings across the top, and repurposed tables with varying content within each of the transactions workflow processes.
Menus and layout remain consistent across the transactional workflows. Any nuances in data display exist due to special considerations based on the source of the transaction, such as the requirements for transaction financial totals.

The Print command offers several options including Quick Print and Advanced Print. Advanced Print offers the options to select multiple reports, and to hide pricing in the printout. In the previous version of CollisionLink (v.4) these options were available under a command call Print. This was confusing and the new name has alleviated the misperceptions. The new command name maintains the ability to toggle pricing options, support legacy users, and also prevents any semantic confusion about what the difference may possibly be between a Print and a Quick Print command.
INFO@NNGROUP.COM  Executive Summary

The print commands offer the ability to print without prices, among other things. The **Quick Print** and **Advanced Print** commands certainly appease legacy users.

Part numbers are extremely important when ordering automotive parts. If a part number is right then everything falls into place. But if the user is unsure of the part number then it can laden the order process with issues. Often times an order comes in and the part requested is incorrect, for example, a shop requests a hood, but the hood part number is for the wrong model. The system avoids these issues by predicting them and aiming to reduce them. Specifically, designers implemented a linked control relating the **Part #** field with the **New Part #** field. The **New Part #** field considers the number in the **Part #** field as well as other information in the order, and based on database and inventory knowledge, suggests better parts. When the user clicks the **New Part #** field a drop-down displays part numbers the user may possibly want. As an added convenience, the current part number displays at the top of the list. Users may also manually type the new part number if they know it, and the system will add it to the order.
A linked control offers a more correct part number based on the requested part number as well as other information in the order.

In many cases the user will want more information about a recommended part, or even to compare the recommended part with the current one. Selecting the Additional Information link from the bottom of the drop-down opens a matrix with more details about each part, and the possibility to select one.

This information should cover most peoples’ needs, but in the event that the user needs more information than what is in the table, clicking the part number in either the Buyer Requests or Response to Buyer section will open a dialog that provides all the information that is available but that was hidden by default. This type of convention has been extended throughout the app so the user can always expect to click on a part number for detailed information. And the approach itself, hyperlinking to information, allows OEConnection to provide information that they may infrequently reference, and it allows room in the UI to expand with new features without cluttering the main details page.
In the event that the user needs more information than what’s in the table, clicking the part number in the *Buyer Requests* or *Response to Buyer* section will open a dialog that provides all the information that is available for them but that is also hidden by default.

In another attempt to save on screen real estate, designers created the *Info Center* which shows information (that used to appear constantly in the main content data table in a past version of the app.) The categories are separated into tabs—*VIN Details, Order Info, Notes, Photos,* and *Event Log*—which are linked to from the menu on the details screen. This allows the user to select the information he wants and be taken directly there.
The Info Center helps reduce clutter in the main content area.

OEConnection, in this case the D2D Express app, thinks of everything and doesn’t just drop the user when it faces any backend issues. Namely, when a part is out of stock or backordered from the OEM, the system automatically searches other dealers’ inventories for that part.

Given the ever-moving inventory, this system poses a level of urgency to the ordering process: These offers are first-come, first-serve and have limited time duration. The Overview screen provides enough information to the dealer to allow him to quickly and confidently decide whether he can commit to selling the part in question. And once the dealer commits, the system provides the necessary shipping details so he may quickly box up the part and ship it to the appropriate destination.

Leaving no stone unturned, the omnipresent notification system, along with e-mail, and text alerts, helps support this process. So users will never miss an offer.
When a part is backordered the system queries all the dealers until it finds the part and allows them to sell it on behalf of the OEM to a dealer in need.

The most common user task is simply to locate parts, so other features may be lost on them or get in their way. Given this, designers devised the D2DLink console. The Locate task is the main area of the screen, surrounded by Opportunities, Discounted Parts, and Inventory Updating.
The D2DLink console makes *Locate [parts]* the main area of the screen, surrounded by *Opportunities, Discounted Parts, and Inventory Updating.*

When the user finds a part he wants the system predicts and expedites his next step, like a busy, smart elf. Knowing what information the dealer needs in order to move forward, the app presents the following items to him in a neat little package:

- A view of his own inventory to verify that he doesn’t already have the part in stock,
- Primary Distribution Center results, (PDC,) which shows the current stocking levels at one or more OEM warehouses,
- Parts that are available in the dealer community, sorted by the dealership’s favorite trading partners, then the remaining dealers sorted by proximity.

After this appears a list of parts that the dealership has decided to buy electronically. To boot, the system can order multiple parts both in quantity and part numbers or SKUs from multiple dealerships and send those to our system in bulk. And, multiple part locate results can remain accessible simultaneously, which reinforces the notion of having multiple transactions open in other workflows. This is important because often times dealers are looking up a part for a walk-up customer at a counter, while looking up a second part for a customer on the phone, while looking up a third part on behalf of their own service department. The ability to maintain these queries without having to start and stop is vital to the parts department’s efficiency.
The system predicts and expedites the user’s next step, like a busy, smart elf. Knowing what the dealer needs to know before he can move forward, the app presents important information to him in a neat little package.

Continuing with momentum, the dealer may select *Create Order* from any set of part locate results, then verify their billing and shipping information, verify the correct parts and part quantities, then submit the order. Once submitted, a new transaction is created and sent to the selling dealer.
Create Order leads to verifying billing and shipping information, verifying the correct parts and part quantities, then submitting the order.

To construct a search, users may access the locator, which is available in all workflows in the same location in the upper right area of the screen. Once a part has been entered the user is redirected to the D2DLink workflow and a new subtab with results. The ability to search for a part at any time no matter what the current focus, is required for the users’ success. Thus, designers made it possible to initiate a search to locate parts from CollisionLink. This automatically redirects the user to D2DLink and creates a subtab for the part number entered. This type of accommodation shows a true understanding of user workflows, and does not limit the user based on the app or view he happens to be in when he wants to initiate a task. Additionally, designers implemented the safety net of auto-save as a background task.
To construct a search, users may access the locator, which is available in all workflows, in the same location near the upper right hand corner of the screen.

OEConnection designers were disciplined and thoughtful when considering screen real estate. They frequently prioritized pages and workflows based on users’ tasks and mental models. They made mindful tradeoffs in many cases where they chose to hide information by default, eliminating clutter on the screen, but making the deeper information available and easily accessible with the look of an eye and the click of a mouse.

As for workflow and considering users’ next steps, OEConnection is like the fairy godmother of apps, making suggestions and offers, and doing all the backend work, coming close to perfectly automating everything for the user.

**DESIGN PROCESS**

Prior to the most recent redesign the OEConnection products were “really hard to use”. “They were actually kind of brutal,” says Shawn Kenney, User Experience Designer. So, it was clear that in order to really fix what was broken would require the company to re-build from the ground up.

One of the first steps in the process was to achieve a level of integration across the products that had not been successful before.

- **Site Visits: Hanging Out in the Parts Department**: On site visits played a major role in the product redesign. And these site visits happened at various stages of the design process.
Prior to Kenney joining the design team, the company’s development manager, business lead, and product specialists went on the road to visit dealerships around the country to “hang out” and watch how the parts departments work. Through these visits they were able to get a really good understanding of the use cases, decision tree, what’s important to users, and what steps a typical user needs to take to perform a task. These initial insights contributed to the CollisionLink tool.

Site visits also contributed to shattering the team’s misconceptions of how the users work. “Observational learnings are the best,” says Kenney. “We had a misconception that everybody had a unique process but what we found that they’re mostly the same, with maybe only a few geographic variations.”

This was helpful because it contributed to the task analysis for the tool, but also because finding homogeneity across the dealerships meant the team could use almost any dealership’s staff as a proxy for the greater user base. “For the most part we can find very specific dealers, observe them, and validate against a broader set if we need to,” he says.

- **Relying on Experts.** Kenney was brought to the team as much for his expertise as for his ability to execute on design concepts. That expertise played heavily in the design process. “One of the things I brought was an understanding of the basics of what works from an interaction standpoint,” he says. “That we’d been able to layer that on top [of the application] but we don’t spend a lot of time with taking a design, putting it in front of a user, setting up cameras, nodes on their forehead watching them, etc.” The team relied instead on equal parts expertise and iterative prototyping. For as much as Kenney was a usability expert, the team felt the users were the real experts on whose advice they should rely.

  “We can for the most part take paper prototypes of the design and set it on a table in front of them [the users] and actually have them explain back to us what they are seeing. And a lot of times we don’t even have to have to explain it to them, their eyes just light up and they ask when they can have it because you can see that the light bulb went off when you talk with them. So we try to spend as much time as possible truly understanding how to think about their job.”

- **Using Prototypes:** Kenney used Flash prototypes to demonstrate any radical design changes, such as those in CollisionLink. “We actually made a Flash prototype demo to seven dealerships and had them play with it to make sure we weren’t going too far astray,” he says.

  Paper prototypes were used throughout the process as a tool they could put in front of users to validate design choices. The prototypes were used to help both the internal team and users understand how the pieces of the unified product fit together. And they really worked. “They allowed us to take inventory of concepts we may have overlooked, and discover new feature that we could introduce.”

**Tracking the Design Evolution**

The following screens illustrate the products as they progressed through various design states as the project progressed:
This is a view of a legacy CollisionLink order screen.
This screen shows an example of a wireframe created to show the transactions details in CollisionLink. At this point in the design, the team was still using a tab paradigm to isolate the state the transaction came in as, and then a worksheet view that contained the changes being made. During the prototype phase they learned there was no need for the dealer to see the state the view came in as, so that was eventually dropped. Eliminating unneeded features is one of the best ways to increase usability — and also obviously saves on development costs.
This screen shows the results of a design study conducted by an external agency, prior to Kenney’s arrival at the company. It was aimed at the D2DLink parts locator. The company had been looking for a design framework to use for the new CollisionLink and used this as a starting point. Some basic elements were carried over from this design while some concepts such as “Find it Fast” and “My Dashboard” were shelved, and the tabs, icons and other visual elements were created new.
This screen shows an early attempt to merge together the third party design study with the new workflows Kenney and his colleagues developed. When this screenshot was captured they had yet to integrate application specific branding or the idea of multiple products running side by side, the ability to have multiple orders open at the same time or what would eventually become the organizational strategy around the left nav.

**LESSONS LEARNED**

Kenney shares his experience as the "new guy" joining a team within an organization that had well-established ways of getting things done:

- **First unify - then act.** When Kenney came on board the design philosophy of the organization was in transition. Kenney’s addition to the team was not merely the addition of a new skill set, but the addition of a person who would have to facilitate a change in the way people were used to doing things before any real design change could happen. “We had everyone with their own ad hoc processes and it was based on whatever experiences they’ve had,” he says, “whether it was in use here or at other companies and no one to get them into alignment. Not only to just be consistent in their own process, but understanding the decisions that they make impact other teams.” So, that responsibility fell to him and he learned a lot about the politics of change through this process.
“When a company is going through cultural change - before you do anything understand the level of communication and training required and get all your pieces in alignment before you begin to design and develop because that change isn’t going to happen immediately and it’s going to take time on a person to person level. You can do all the design/usability work you want but you’ll essentially be fighting against people who really should be your allies and you probably won’t get too far.”

His advice: slow down. “Slow the process down just to get people on the same page.”

- **A rush to market doesn’t save time.** “So, there’s always going to be this desire to get to the market as fast as possible but if speed is your primary driver to market you are going to cut corners,” says Kenney. “You are going to miss steps, things are going to slip through the cracks. And if you don’t do all the right steps up front, you’re going to have to do them at some point, typically those things come up in the development process because developers will see the holes that the product teams have missed and they are either going to go to the product team for answers or they are going to come up with them themselves. You’re introducing re-work, burned bridges and hurt feelings, so there’s that whole human dynamic too, that you can avoid by just taking your time up front.”

- **Your place in the org chart affects your ability to affect change.** Kenney learned the hard way that a lot of decision-making hinges on the perception of who’s driving the change. When he joined the company he was placed in the technology department, which is okay except at OEConnection there is a perception that the business units say what the need is and the technology department determines the “how”. “It proved to be very inefficient,” he says. “One, because our business units didn’t really like someone in the technology unit basically changing the requirements on them and introducing a new level of requirements around users. And then also I was in a position where I was actually trying to hold my own management accountable for the quality of the product going out the door which they didn’t really appreciate either.”

Currently he reports into the company’s Chief Commercial officer, which he says, allows him to do two things. “One, it puts me on an even playing field with our business leads,” he says. “That way I can actually work with them as a peer and have in-department discussions when I need to have those. Also since I report directly to their boss so, in the event that we have walls that we need to have removed I now have someone who’s in a position to remove those walls for me and I’m not trying to negotiate or twist arms or bribe or throw my hands in the air and call it a day because I can’t remove those on my own.”

**Project Team**

Kenney was the primary designer and user experience lead for the OEConnection redesign project, but other team members performed specific functions throughout the process. Here is a list of the team members and their roles:
• **Shawn Kenney, User Experience Designer:** Responsible for disseminating team feedback, participating in onsite user research, identifying user mental models, creation of wireframes and workflows, facilitating design sessions, creation of mockups and functional prototypes, participating in validation sessions, creating usability & design standards, and ensuring compliance with existing company usability and design standards.

• **Chris Fuss, Project Manager, FordParts:** Responsible for project goals and timelines, communicating business objectives, identifying test markets for workflow validation and organizing validation sessions.

• **Erik Enright, Project Manager, CollisionLink:** Responsible for providing feedback gathered from users in field, maintaining project timelines, scheduling design sessions, communicating design directions to executive leadership.

• **Jeff Toom, Product Specialist, CollisionLink:** Responsible for providing feedback gathered from users in field, contributing OEM requirements to project

• **Elizabeth Fontes, Product Specialist, CollisionLink:** Responsible for providing feedback gathered from users in field, contributing reporting requirements to project

• **John Haluch, Account Services Representative, CollisionLink:** Responsible for contributing dealer feedback provided by users of legacy applications, understanding of existing user pain points.

• **Carol Mandato — Product Manager, D2DLink/D2D Express:** Responsible for project goals and timelines, communicating business objectives, participating in onsite user research, participating in design sessions, communicating design philosophies and direction to executive leadership, identifying user groups for workflow validation and organizing validation sessions.

• **Erica Holvey, Product Specialist, D2DLink/D2D Express:** Responsible for onsite dealer user research, documenting existing user pain points in Legacy applications, participating in and contributing industry knowledge to design sessions, validating designs with users.

• **Sara Roubenes, Product Specialist, D2DLink/D2D Express:** Responsible for onsite dealer user research, participating in and contributing industry knowledge to design sessions, validating designs with users.
Pictured: OE Connection team members (front row — left to right) Carol Mandato, Elizabeth Woolf, Jeff Toom and John Haluch; (back row — left to right) Erica Holvey, Shawn Kenney, Chris Fuss, and Erik Enright.
Lightweight Applications

Applications in the Lightweight area typically have a deliberately limited feature set and/or may deal with a jocular, fun topic or task. The biggest challenge in creating a lightweight application is not in deciding what to include, but in deciding what to leave out.

These types of applications are often optional, that is, users can take them or leave them. And users are all too likely to leave an app that doesn’t instantly explain itself and immediately enrich the user’s experience or make the users’ activity easier.

Given these criteria, lightweight apps must be thoughtfully designed, responsive, and easy to use, which is no short order, and these applications are not easy to design. In fact, designers of applications in this arena need great discipline, clear goals, and a deep understanding of what their users want and how they think. Designers must also have the authority to strip away the superfluous and focus on making only the necessary and important features. (It’s not surprising that some of the best lightweight apps come from small, nimble organizations, that don’t have to worry about too many cooks in the kitchen.)

Some UI elements that work well in this type of application include:

- Can’t-miss affordances for primary actions
- Minimizing navigation elements or creating linear flows to focus on current actions
- Ruthless pruning of clutter and unnecessary features

Winning applications in this category include:

- CycleStreets Mobile, helps cyclists plan safe, fast routes for cycle journeys in the United Kingdom (and some other areas) using crowd-sourced map data.
- VitreaTeach, an educational tool that radiology residents and practicing radiologists can use to save, manage, and share medical images.
- Wakelicious, alarm clock that wakes you up by playing YouTube videos sent by your friends.
WHY THEY WERE CHOSEN

CycleStreets Mobile is an excellent lightweight application because of its:

- Innovative solution to the widespread problem of small targets on touch interfaces
- Elegant design, based on a deep understanding of the problem space
- Tight focus on core features, which makes the app flexible enough to work on multiple platforms.

OVERVIEW

CycleStreets Mobile is designed to help cyclists plan the safe, fast routes for cycle journeys in the United Kingdom (and some other areas) using crowd-sourced map data.

Why it’s unique: Many applications offer maps and route planning, but CycleStreets Mobile is uniquely optimized for a specific task and audience. This lightweight application is designed for cyclists on the move, and thoughtfully leverages the characteristics of cycle journeys to create a fast and direct interface. CycleStreets Mobile is delivered as an HTML5 website in order to make the tool accessible to all mobile users, regardless of device or platform. Designers took advantage of the constraints to create a simple and elegant interaction for the users’ core tasks.

The organization: CycleStreets is a not-for-profit organization created by cyclists for cyclists. With just two employees, CycleStreets has built a sophisticated route-mapping service and contracted with external developers and volunteers to build four user interfaces: a full website, an iPhone application, an Android application, and a mobile website application. Since its launch in September 2011, the CycleStreets mobile application has been used to plan 322,000 kilometers of cycle journeys.

Design philosophy: Throughout the design of the mobile interfaces, CycleStreets has been guided by the idea of quick and direct interaction: that a user should be able to plan a route simply by marking the start and end points on a map. The designers call this the “three-taps” interface: Start, Finish, Plan!
The CycleStreets Mobile route planning page allows users to interact directly with a map to set the start and end points for a journey.

WHO ARE THE USERS?

CycleStreets aims to serve people who are already active cyclists as well as those thinking of taking up cycling, but the mobile site is primarily intended for current cyclists who are about to begin or are in the midst of a journey. There are two main types of mobile users:

- Beginner cyclists
- More experienced cyclists.

Martin Lucas-Smith, the Director of CycleStreets, explains, “From the feedback we get, there are these really two distinct groups of people: those for whom we’ve removed a barrier to starting cycling; and those who say, ‘Gosh, I’ve been cycling around here for years and never found that cut-through and it saves me two minutes of my journey away from this horrible road.’”

CycleStreets Mobile also addresses access gaps left by Android and iPhone applications. While these native applications currently offer richer interaction, the universal access offered by the HTML5 mobile web application is a significant benefit, especially for a not-for-profit organization with limited development resources. At the time of this writing, 322,000 kilometers of cycle journeys planned.

THE APPLICATION

CycleStreets Mobile is an HTML5 web application, optimized for mobile devices. The routing engine helps cyclists plan the fastest, safest bicycle routes using map data from the OpenStreetMap project.
Users can plan a journey by selecting start and finish points directly on a map, or by searching for an address or place name.

The routing algorithm finds journeys that avoid highways and hills, and prefers cycle paths and quiet roads, quite different from car-oriented routing. Cycle routing in the UK (where CycleStreets is based) is particularly challenging, as there are few grid-based city layouts.

In addition to distance traveled, the journey plan also displays estimates of the time to complete the trip, calories burned, and CO₂ saved by cycling rather than driving. CycleStreets can display the route against three map backgrounds: OpenStreetMap, OpenCycleMap, or Ordnance Survey. Ordnance Survey open data maps are official government maps, while OpenStreetMap is a more data-rich, crowd-sourced map. OpenCycleMap displays OpenStreetMap data, but highlights of official “cycle routes” and other points of interest such as cycle parking, while de-emphasizing major car roads.

CycleStreets Mobile displays route information such as estimated time, calories burned, and CO₂ saved; as well as giving users the option to switch between faster or quieter routes, and to display the official government maps (OS) or the crowd-sourced OpenStreetMap or OpenCycleMap versions. Turn-by-turn directions are also available.

In addition to the journey planner, CycleStreets includes a “Photomap” area which lets users upload geotagged photos. The Photomap is intended primarily to document cycling-related infrastructure issues, which can then be brought to the attention of authorities.
**INNOVATION**

Most mapping applications, including previous versions of CycleStreets, let users select any point on a map by tapping the desired location with a finger, or tapping and holding to drop a pin. These methods lead to the “fat finger” problem: users have trouble selecting small targets, because their fingers obscure the entire target they are attempting to select, and they end up selecting the wrong item or not selecting anything. This is bad enough when dealing with small buttons or text, but for mapping applications the situation is even worse since the entire screen is selectable and tapping just a few pixels away from the intended target can generate a completely unintended route and set of directions. To make an accurate selection, users are forced to either tap several times until they get the right point, or use another gesture to zoom in on the map and make their target location larger.

The CycleStreets mobile application provides an innovative solution to this interaction problem: The user is presented with a crosshairs indicator overlaid on top of the map image. They can touch any part of the background map and drag it to move their desired location into the center of the crosshairs, and then tap a large button at the bottom of the screen to select that location. This interface neatly avoids the necessity of blindly selecting a screen location that is hidden behind your finger.

This unique selection method is especially appropriate for mapping cycle journeys because they are generally shorter than car journeys. Users planning car journey need to zoom in enough to see details such as street names, and then zoom out to a larger scale to see the entire journey and endpoint. With shorter bicycle journeys where both the full route and detailed street names can be seen in a single screen, the only reason for zooming is to make it easier to select endpoints accurately. Thus, the alternative selection method that does not require zooming significantly simplifies the interaction for bicycle journey planning.

Although this interaction method is unfamiliar for many users, they quickly learn how it works because as soon as the map loads the crosshair selector and button are visible. These interface elements act as a cue to immediately let users know what they should do. In contrast, applications which rely on a tap and hold gesture to select a location usually just display a map with no controls, and assume that all users already know how to make a selection. CycleStreets found that on their iPhone and Android applications, which did not provide a crosshairs interface, some people didn’t immediately understand how to use the map, as Lucas-Smith explains, “Because there’s nothing much on the map that says ‘We’re waiting for you to actually do something.’ People often would just accidentally tap on it, and then a mark appears, and then they realize, ‘Oh I’m supposed to click on the map and set my point.’ The crosshairs just seems to remove that problem.”
Left: The beta version of CycleStreets Mobile required users to tap and hold a point on a map in order to select that point. Users sometimes accidentally selected the wrong location while they were trying to move the map.

Right: The final launched version of CycleStreets Mobile makes selecting an exact location easy by letting users move the background map so that the appropriate location is beneath a central crosshairs interface, then tap a button to make the selection.
Because cycle journeys are generally shorter than car journeys, both the route overview and the map details can often be seen on a single screen, without moving the map or zooming.

**CONSISTENCY**

While the map selection interface is different, the controls and labels for navigating around the application itself are ones that most users will already be familiar with, including the following:

- The *Home* button in the top left corner of all pages (except the homepage itself)
- The settings screen, linked from the homepage with a gear icon and the label *Prefs*
- A blue dot which appears on the map to indicate the current location, as determined by GPS data.

These standard controls are instantly understandable, provide persistent access to core features, and take up very little screen real estate.
The home button located in the top left, and the blue dot which indicates the current location, are consistent with many other interfaces.

CycleStreets Mobile also provides excellent visual feedback to users throughout the journey planning process. Once a start point is set, a distinct flag appears and the interface labels change to indicate the next step, setting a journey end point.

These small details make a big difference in users’ ability to discover and take advantage of application features. Ideally, all applications should be easy for new users to understand and learn. This principle is even more important for web-based applications such as this which are aimed at a broad audience, as they can expect many first-time and casual users. Clear task labels rather than cryptic icons, and persistent, understandable navigation controls are essential for such tools.
Before (left) and after (right): Clear visual indicators let users know when they have selected a location.

**FLEXIBILITY**

Because mobile devices are generally used by just one person who carries the device around almost constantly, they lend themselves to more personal, customized experiences. CycleStreets supports this dimension of the mobile experience by letting users easily control and customize the journey planning process.

For example, some riders may be quite cautious and prefer to always take quiet streets, even though a busier street might be quicker. CycleStreets mobile allows users to set their default route type preference as well as their normal speed in the Preferences area. They also have the option to turn off the GPS location feature, in order to conserve battery life.
Users can set defaults for route type, map type, and estimated speed in the Preferences area.

Preferences for route and map types can vary not just by individual but also by situation. A single person may prefer busy, direct routes when she’s in a hurry and quieter trips when journey time is less important. CycleStreets mobile accommodates the need to adjust these preferences on the fly by providing buttons that let users quickly switch between quiet, fast, or balanced journey plans. The controls for adjusting the route type appear directly on the journey map, and because there are only three controls, it is obvious which option is currently selected.

The route switching controls on the CycleStreets Mobile application are a significant improvement over the CycleStreets iPhone and Android native applications. On the iPhone application, users can only select a route type before planning a journey; the Android app does allow users to switch between route types, but they must use the device’s physical menu button (which users often overlook) in order to access the route type menu.
The CycleStreets Mobile website displays three tabs on the journey display screen which let users quickly switch between quiet, fast, and balanced routes to compare their options and select the best plan.
The iPhone application currently requires users to select a route type in the application settings before planning a journey, which many users may never discover as the (dialog box) modal explaining this quirk appears only once, after planning your second journey. Comparing route types for the same start and finish points is therefore not easy.
On the CycleStreets native Android application, users must use the device’s physical Menu button in order to change the route type.

CycleStreets Mobile also gives users a choice of different input methods when planning a trip. The crosshairs interface makes it easy to quickly set start and endpoints on a map, but users can also search for a specific address or return to a previously saved route.

Any route planned by a user is automatically saved without requiring any login or explicit save command by the user. This model (dialog box) works well for an application intended for personal mobile devices generally used by just one person. Since the application is delivered through a website, it is subject to interruptions in internet access; automatic saving helps users recover from network interruptions by quickly recreating routes.
Users can choose between planning methods: by map selection, by address search, or by using previously saved routes. Users may also select their current location as a starting point.

INFORMATION DISPLAY

The CycleStreets Mobile application offers users a wealth of geographic information in the form of three different maps:

- OpenStreetMap (OSM) is the “Wikipedia of maps” and includes crowdsourced information about routes and nearby businesses
- OpenCycleMap (OCM) uses the same underlying data as OpenStreetMap, but is optimized for cyclists including highlights of officially designated cycle routes and hills
- Ordnance Survey (OS) includes open data maps, produced by the official United Kingdom government mapping agency.

Each of these three options offers a unique value to the user. The crowd-sourced geographic information on the OpenStreetMap and OpenCycleMap displays are invaluable to cyclists interested in finding shortcuts that don’t appear on official maps. However these maps can sometimes be quite cluttered with information and official routes, making it more difficult to see the basic geography. CycleStreets’ designers found it challenging to come up with a single route display that works well against all three backgrounds. Many mapping applications display a map in the background and highlight the route plan on top, but the OpenStreetMap and OpenCycleMap displays already make extensive use of different colored highlights to indicate major roadways and cycle paths.
Instead of adding more highlighting to an already busy visual field, CycleStreets Mobile marks the planned route with a thin but bright, saturated violet line. In another concession to the complexity of the OpenCycleMap display, CycleStreets shows only one route plan at a time, rather than attempting to display all three types simultaneously. Lucas-Smith explains, “If we had drawn all three routes on the screen, you have the two highlight colors as well as the general map base, and you then have to have three very prominent colors that stick out, for balanced, quietest, and fastest. It was very clear on the mobile that that really wasn’t going to work very well.”

The OpenStreetMap (left), OpenCycleMap (center), and Ordnance Survey open data (right) cartography designs offer different degrees of detail and cycling-specific information.

**DESIGN PROCESS**

The CycleStreets Mobile designers conducted informal usability evaluations, simply by asking friends to plan a journey using the mobile website. The underlying UI design was based on collected user feedback about the other CycleStreets interfaces, which provide the same functionality on different platforms. “… We’d already learned a lot of the problems in the previous applications,” says Lucas-Smith. For example, iPhone users find it difficult to switch between fast and quiet routes, since the iPhone application requires you to start the journey planning process over at the beginning in order to change the route type. “The HTML app is actually the easiest of all four in terms of route switching. We made sure that this app dealt with all of that learning over what’s been about three years of iterations of products,” says Lucas-Smith.

**Implementation**

The need to make CycleStreets Mobile operate on many different platforms was one of the biggest challenges of this project. Performance testing and bug fixing for different platforms added complexity to the development process, especially since CycleStreets as a non-profit did not have the resources to purchase different testing devices. The need to deliver the application via the web also significantly influenced the design.
For example, since additional features multiply the device compatibility issues, the designers kept a tight focus on implementing just the core features essential to plan a cycle journey. For example, advanced functionality such as the ability set the starting point via map but use search to find the endpoint is not supported. Future iterations may add this feature, but the time needed to support this complex feature through the web (and make sure it works on all devices) put it out of reach for the first version.

The addition of the crosshairs for map location selection actually simplified the application code as well as the user experience. The beta version, which allowed users to select any point on the screen as a starting point, required the application to determine exactly where the user was tapping on the screen, but the crosshairs interface only needs one preset screen location: the exact center, where the crosshairs are focused. “Suddenly all the code to determine where exactly the person is tapping on the screen disappears, because now when you click the button that says Click to set the web browser knows exactly where it is because it’s just the centerpoint,” says Lucas-Smith.

JQuery Mobile Framework

The use of HTML and JQuery Mobile rather than a native application toolkit influenced the design in other ways as well. Although both the iPhone and Android CycleStreets applications include a persistent menu bar, the JQuery Mobile framework was not optimized to support this structure. Instead, the mobile website features a homepage with five main navigation options, and all other pages on the mobile website include a Home button rather than a full menu bar. This structure offers less support for switching between tasks, but also leaves more screen space free to display maps and routes.

Standalone Code

For the time being, the mobile website is a standalone site, and does not rely on any of the main website code. “We took a very deliberate decision to do it as a standalone thing because the source code for the main website was not in a state where you could easily repurpose it for mobile. Also it was very clear from our experience with the Android and the iPhone applications that there’s just too much extraneous detail that you’d have to work quite hard to get rid of,” says Lucas-Smith.

LESSONS LEARNED

The CycleStreets Mobile design team benefited from a few key practices during this project. The resulting advice is described here:

Limit your focus
The CycleStreets Mobile designers quickly realized that it was essential to focus on core tasks and eliminate the nice-to-have type features. “We found that we had to cut it down quite a bit compared to native apps: keep it as simple as possible and make sure that some of the really irritating things with native apps were dealt with, particularly the switching between routes and the problem with the thumb,” explains Lucas-Smith.
Track and learn from your mistakes
Documentation of issues with previous applications proved to be an invaluable resource during the design of this version. Lucas-Smith says, “I think we were very lucky in this as it was the fourth time we’ve gone through making an interface to do this job, so it wasn’t like traditional web application testing where you’ve got a completely new thing and have people testing it out to see where the problems are. We kind of knew what the problems were because we’d already had a lot of feedback.”

TEAM
The design team included:

- Project Management: Martin Lucas-Smith
- Developer: Anna Powell-Smith

CycleStreets Ltd. was created and is managed by Simon Nuttall, the lead developer, and by Martin Lucas-Smith, who serves as the organization’s Director (in addition to his full-time employment as a web developer at the University of Cambridge.) Both are regular cyclists based in Cambridge, who have been involved with the Cambridge Cycling Campaign, a local bicycle advocacy group, for many years. They plan to release the CycleStreets code as an Open Source project and enable others to get involved. The code of all the mobile apps has already been released.

Anna Powell-Smith is a Cambridge alumna and freelance front-end developer now based in London.

With the team members based in different cities, most collaboration was done remotely, except for a kick-off meeting which took place on a train between Cambridge and London, and a curry dinner in Cambridge where the crosshairs concept was conceived by Tom Steinberg, an acquaintance of the CycleStreets designers.

So never turn down the opportunity to partake in a meal, especially a curry, with great thinkers as wonderful things may come from the experience.
VitraTeach

**Organization:** Vital Images, Inc.

**Application:** View and share medical images

**Headquarters:** Minneapolis, MN

**Design team:**
- Navid Sadikali, Principal User Experience Designer
- Thomas Myers, Senior Manager, Strategic Technical Marketing

**OVERVIEW**

**Main function:** VitreaTeach is an educational tool that radiology residents and practicing radiologists can use to save, view, annotate, share, and present medical images and case descriptions in order to support learning and teaching.

**Why it’s unique:** Normally, “knowledge management” and “lightweight” are not terms you hear in the same sentence. VitreaTeach breaks the mold by approaching its complex subject matter with a simple interface that focuses on how users actually behave, getting out of their way to make the content more visible and usable. The application emphasizes the personal and social nature of information management, especially in an educational context. It’s especially refreshing to find this approach in the healthcare field, which is often dominated by confusing, inefficient interfaces.

**The organization:** Vital Images, now part of Toshiba, builds medical imaging systems. The new in-house design team is focusing on driving product strategy through great user research and design.

**Design philosophy:** Applications intended for physicians must offer significant benefits and very low barriers to entry in order to be adopted. The healthcare field in general, and Vital Images in particular are undergoing a transformative realization: advances in medical practice have been limited not by technology but by poorly designed tools that are ill-suited to the way medical practitioners understand and share information.

**WHO ARE THE USERS?**

VitreaTeach has two target audiences with distinct goals and behaviors:

- **Radiology residents:** physicians in training who are highly focused on learning from the dozens of cases they encounter each week
- **Radiologists:** experienced physicians who may play a significant role in educating residents.  

Two primary personas represent these user groups and helped guide the design of VitreaTeach: Amanda, the resident; and Mack, the Chief of Radiology.

The Amanda persona was created specifically for this project, but Vital Images had already been using the Mack persona for other products. Although VitreaTeach is primarily intended for Amanda, it’s also a useful tool for Mack.

The design team initially thought that the learning process in radiology would follow a student-teacher model, with supervisors assigning sets of cases to residents to study. After spending time with users, they discovered that, in fact, most teaching is informal and rapid, not explicit or managed.
Residents work with real patients and cases, but their focus is on learning. They may encounter dozens of cases each week, and participate in discussions or give presentations about cases. Existing tools are optimized for maintaining patient case files and interacting with different departments in the hospital, but offer little support for managing that content for learning purposes. When Amanda sees an interesting case, she needs a way to save it so she can access it again later, take some notes about it, and share it with her colleagues or supervisors. Currently, this task of saving interesting images relies on residents writing down the patient ID numbers of interesting images, then later retrieving the images and perhaps burning them to a CD.

Residents are also accustomed to highly-interactive, lightweight applications such as instant messaging and Twitter. They want intuitive, flexible interfaces that give them the freedom to use content in a variety of ways; for example, Amanda might like the ability to quickly create groups and invite others to join, so she can easily share cases with the other residents in her rotation.

The educator persona is very different from the resident; not only is he already an expert in radiology, but he may not even be practicing much on a day-to-day basis. Instead, he focuses on managing the department, which includes helping the residents progress, and sometimes actually quizzing them. He may also be interested in publishing an institutional library of interesting cases, such as “10 Common ER Trauma Cases.”

The organization is awaiting pending FDA approval.

THE APPLICATION

The early plans for VitreaTeach focused on the content—interesting radiology images—as a valuable resource. The idea was to encourage users to upload such images into a cloud-based application, and then later the team could determine how to monetize that content.
This content-centric focus, however, led to the development of an application that was too hierarchical and complex for physicians to easily incorporate into their normal workflow. Vital Images works with medical professionals to test and evaluate software. During these interactions the team learned invaluable lessons about their users. For example, a senior physician who had a longstanding relationship with the company and was one of the early users of VitreaTeach commented, “The way it is I don’t really want to use it…It’s too cumbersome. It’s not what I need.”

User-centered design methods such as observational research helped guide VitreaTeach in a new direction: towards an interface focused on supporting social interaction and organizing content in a way that is personally meaningful to users, rather than by a universal hierarchy.

VitreaTeach integrates with existing hospital software where the target users do most of their day-to-day work. When they come across an interesting image, they can save it to VitreaTeach by right-clicking on the image.

Once in the VitreaTeach system, users can

- comment on images
- create unique tags for images
- share images with others
- present cases to others.

With this type of content privacy is a major concern. To protect patient privacy, VitreaTeach changes patient names and blocks other identifying information, rendering the content anonymous. In this way it’s possible for practitioners to view images from other institutions without violating patient privacy rights.
VitreaTeach lets users save, annotate, and share interesting medical images.

INNOVATION

Extremely sophisticated classification systems have developed over time to describe medical information. For example, medical images can be categorized according to the part of the body they depict or by the pathology they portray. While these are both valid taxonomies, they introduce a level of complexity that is simply not necessary for most of this application’s target audience. Navid Sadikali, the principal user experience designer of VitreaTeach, explains, “The existing process, with a huge taxonomy was an important thing to get rid of. The medical dictionary is so big, people don’t even use it.”

Instead of this top-heavy approach, VitreaTeach employs a flexible system that is much more related to users’ actual tasks. The Anatomy and Pathology hierarchies are still present, but are deemphasized in order to focus on classifying images according to social or personal dimensions.

For example, the dominant hierarchy is set up to match the radiology resident’s mental model of his own activities and the people he interacts with closely and tangentially. Three levels display the distribution of images:

- **Individual**: *My Cases* includes only cases saved by the user
- **Institution**: Cases saved by everyone within the teaching hospital and subgroups of the institution
- **Public**: *Public Cases* are those saved by anyone at any institution into VitreaTeach.
Using observational research we realized that teaching meant tracking interesting images from old cases for self-learning, more than formal top-down case libraries for teaching.

This structure corresponds closely to the primary users’ activities. The design team observed radiology residents and found that rather than using formal top-down case libraries, residents engaged in self-education by reviewing interesting images from old cases. Thus, cases a user has personally saved for her own use are the most important types of content. The next most frequent use would be to share images with small groups of colleagues, such as the other residents in her current rotation. Finally, an important but less common usage would be to review an institution wide library of cases, or cases from another institution.

VitreaTeach also includes a system of tags and filters which provide a complementary way of browsing through images. Users can create their own tags to describe content according to its personal relevance. For example, a user could create a Present tag to describe images he plans to use in a presentation. This flexibility is a game-changer, because suddenly the application has become a tool customized to serve the user’s tasks, rather than a tedious, dense hierarchy he must navigate.

So, for example, a user could browse to My Cases in order to see just the cases she personally saved, then click on a tag she had previously created such as Discrepant Findings to filter the content to only those bearing that tag.

Both the hierarchy and the user-generated tags remain visible in the left navigation pane, making it easy to see which subset of content is currently visible. “Users loved this screen and couldn’t believe it was a web-based application,” says Sadikali.
Cases can be filtered by their associated groups or by custom user-generated tags. Here, the images are filtered to display only My Cases which have been tagged with Present.

An early prototype of VitreaTeach planned to combine anatomy and pathology in a traditional hierarchy. This concept was discarded after users loudly rejected it (although it is the same model of a few existing similar products.)
The first version of VitreaTeach used folders as a visual metaphor and devoted a great deal of space to listing other institutions.

**FLEXIBILITY**

VitreaTeach offers users several different modes of use, each tailored to support a specific type of task. The default mode includes persistent navigation that lets users efficiently browse through cases, and a Case View screen that displays the full case description and comments on a single page that can be consumed at a glance. This mode works well for individual use, but another important task for residents is to present a case to supervisors and colleagues in a collaborative learning setting.

Before VitreaTeach, users relied on PowerPoint for their presentations, cutting and pasting images into a presentation file. The initial version of VitreaTeach maintained this metaphor of creating a presentation, as Sadikali explains, “Initially, the presentation was something you made, but one of the reasons for building this application was that PowerPoint was too much work. Users didn’t have time to copy and paste images, they needed to be able to present right now. So we needed something much quicker. That’s why Present Mode was so important.”

The current version of VitreaTeach treats a presentation as a mode you enter, rather than an object you create. From the list view screen you can click a button to enter present mode. Present Mode displays a different format of the case information...
which removes the application navigation controls and adjusts the content layout to divide the case description into a series of screens with larger text, more suitable for displaying to an audience. Users can configure the present mode to focus on the most important elements.

The *Present* mode makes it easy to share cases with others in meetings. Each segment of the case is presented on a separate screen, and text is enlarged for easy reading.

Users can customize their presentations by selecting which elements to include.
VitreaTeach also provides a Quiz Mode which temporarily hides the textual description of a case to let users test their memory. Sadikali explains, "If you have a case image and don’t remember exactly what the diagnosis was, you can go into Quiz Mode and the description and comments are hidden until you click to show them."

The first version of VitreaTeach included specific areas for Presentations and Modules in the main application navigation. Now, these areas have been entirely eliminated and instead users can instantly switch any case into either Present Mode or Quiz Mode. This shift helps keep the overall application experience lightweight and simple.

*Quiz Mode* hides the case details until the user clicks to reveal them.

VitreaTeach also gives users the flexibility to share content with others in a variety of ways. In addition to sending a message through the VitreaTeach system, users can simply copy a URL and share it through any tool they choose, such as Twitter. This option was important because visits with users showed that they already use systems such as Twitter, instant messaging, and web conferencing to exchange information. The VitreaTeach design team considered incorporating a live sharing feature, but ultimately realized that the audience would most likely already have established online collaboration channels, and it would be more effective to simply make it easy to integrate VitreaTeach content into those existing channels by providing URLs for each case.
Users can share cases saved in VitreaTeach either through the system, or by copying a URL.

**INFORMATION DISPLAY**

Each screen of VitreaTeach is carefully optimized to make the most important information easy find. The two main principles guiding the information display design are simple, but overlooked by many application designers:

- Devote enough space to the most important content
- Don’t show items that aren’t necessary to the current task.

For example, the *Case View* screen displays one large medical image, thumbnails of related images, and text describing the image contents. Initially, the majority of the screen space was devoted to the primary image, but user research revealed that this actually wasn’t the ideal layout for users’ tasks. Sadikali explains, “The image area was originally larger, but through talking to users we realized that text is far more important than we anticipated. Often they know the image but the meaning of it is what the teaching system is about...in some cases the history and details are actually just as important, or even more important than the text. That was part of our learning, to realize that for this particular system, images are not necessarily the focus.”

Accordingly, the final *Case View* screen reprioritized the screen real estate, devoting more space to the details and comments about the image.
The VitreaTeach listing screen is another example of great information display—not because of what it fits in, but because of what it leaves out. “We cleaned up a lot of things...to emphasize the right information. When you’re searching, the case description and this history is really important, but everything else is not important. The modality, date, and age columns are really critical, but people aren’t really interested in a whole lot of other information. We have a lot more information that we could show, but it would just clutter the list,” explains Sadikali.

This early version of the Case View screen devotes the majority of the screen to images, leaving only a small area for the descriptions and annotations, which wasn’t sufficient for users.
The final Case view screen layout reflects the importance of descriptions and annotations for learning, and condenses the images into a smaller area.
The case listing screen shows just a subset of the available information, focusing in the key parameters: History, Tags, Modality, Date, and Age.

**INTERFACE ELEMENTS**

The VitreaTeach interface was built with both traditional GUI and touch device users in mind, an important consideration because visits with radiology residents revealed them to be extremely technology-savvy. One of the partner hospitals was even considering providing iPads to all incoming residents, so tablets are a very realistic platform for use of this application. VitreaTeach supports tablet users by providing explicit, visible controls—rather than relying on context menus—for tasks such as switching into Present Mode. The Case View screen also supports touch events, so users can zoom in on an image using gestures.
The Present Mode and Quiz Mode buttons are easily accessible for users on touch devices.

**DESIGN PROCESS**

Once users tried the first version of VitreaTeach, it became clear that it needed more work. Vital Images has ongoing relationships with medical professionals who provide user feedback on various products, and one influential user actually specifically suggested involving a user experience designer to improve VitreaTeach.

While the initial feedback suggested that the visual design needed improvement, but after a closer look at the structural issues and potential outcomes, Vital Images realized the issues with VitreaTeach ran deeper than just the appearance, and decided to pursue a full redesign. Sadikali convinced stakeholders of the value of a full user-centered design process by using familiar terminology and concrete examples, “I use visuals, like showing the wireframes and then the visual design that came later, to express the difference between visual and interaction design. The visual approach tends to work pretty well because people can see the difference. I also used terms that people understand, like “workflow,” which has some cachet in our industry.

**Background Knowledge**

Sadikali relied heavily on his past knowledge about radiologists (acquired over the course of fifteen years working in field) to inform the design for VitreaTeach.

Because medical software can directly impact patient outcomes, its development in the United States is supervised by the Food and Drug Administration (FDA.) Software companies who build products to directly support the practice of medicine are basically required by the FDA to do user centered design—that is, to have users evaluate the products and ensure their effectiveness before the products can be sold. Because of these requirements, Vital Images has ongoing relationships with several teaching hospitals which provide product feedback. Design researchers regularly interact with medical practitioners, as Sadikali explains, “We have a lot of background user research that carries over. We actually film users in hospitals, and have a lot of footage of radiologists. But we still do user research because things change, and we need to understand what’s going on in the field, and what the new procedures are.”

The accumulated wealth of information helps create a rich picture of how radiologists think, what their goals are, and how they are likely to behave. “We empathically understand what they’re trying to do...that’s my role as the designer,” explains Sadikali. “We know they don’t have to use the product and if it’s not easy, they
Targeted Research

To guide the design of VitreaTeach, Sadikali supplemented his general background knowledge about radiology practice with specific research targeted at the learning process for radiology residents. Activities included:

- Brief email surveys to assess the importance of competing goals and design elements
- Telephone interviews to gather specific details about how residents study and learn
- On-site observation of residents practicing and studying.

This additional work was crucial to the design of VitreaTeach because it led to some surprising insights, such as the fact that most learning was informal and self-directed, rather than official assignments created by instructors.

This target research also revealed that managing content was the most important user need for VitreaTeach to address, even more than viewing the content. Free competitor products support the viewing of images but the ability to easily save, find, and share them is an area where VitreaTeach can provide unique value to users.

Conversations with senior practitioners revealed key information about the business model as well, as Sadikali explains, “We were thinking everyone would want to share, but the reality is institutions aren’t necessarily interested in cases from other institutions. And some sites have told us that right now, they don’t want to publish to the public web, in part because they believe it is a lot of work to publish to the web, and valuable teaching cases are seen as a proprietary work that serves the institution.”

“This design process reinforced the importance of observational research to find out how people really behave. We broke through the communication that users typically have which is, ‘You need to include this feature and that feature,’ and got them back to, ‘What are you really trying to do, what is radiology?’ I started by asking these basic questions to work up to the more complex questions, to get to their mental models. Even if you have been given a very explicit set of requirements, you don’t necessarily know what problem you really need to solve. They may not be the real needs, or it may be a mapping of the real needs, but it’s the wrong mapping, so you need to do something different from what the requirement says to achieve the right result,” says Sadikali.

Expert Review

It’s essential to find out whether users can understand and use an interface, but difficult to get busy medical professionals to come into the lab for a usability test. Sadikali addressed this by conducting “review” sessions with practitioners to get feedback about his prototypes.

Via WebEx online meetings, Sadikali showed practitioners screens built in Fireworks and linked together. These sessions weren’t traditional usability tests, but aimed to

4 The Picture Archiving and Communications System—PACS—are the basic systems radiologists use for their core tasks.
evaluate users’ understanding rather than simply explain the product, “In the expert review, I don’t just do a demonstration. I’ll ask things like, ‘If I clicked here what do you think would happen?’ or, ‘How do you think this would work?’ before explaining the functionality. Sometimes I’ll show people a flow and ask them to comment on it, people can comment on a workflow quite well. We can do a review like this before we actually have working software, and at a stage where we can make larger changes easily,” says Sadikali.

Implementation

The design and development of VitreaTeach was accomplished in about nine months. For the first six months it was a solo project, with one person working on both design and development. A designer and team of developers took over for the redesign during the last three months of the project.

<table>
<thead>
<tr>
<th>PROJECT TIMELINE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2010</strong></td>
</tr>
<tr>
<td>Investigation by marketing</td>
</tr>
<tr>
<td><strong>Nov 2010</strong></td>
</tr>
<tr>
<td>Version 0.1 pitched to focus group at RSNA, a large radiology conference</td>
</tr>
<tr>
<td><strong>March 2011</strong></td>
</tr>
<tr>
<td>Phase 1: Alpha release to three teaching hospitals</td>
</tr>
<tr>
<td><strong>April 2011</strong></td>
</tr>
<tr>
<td>Feedback from some users that we need to work on design, this filtered through organization as a “visual design problem” or “graphics problem”</td>
</tr>
<tr>
<td><strong>May 2011</strong></td>
</tr>
<tr>
<td>Design team proposes a full redesign (rather than visual tweaks)</td>
</tr>
<tr>
<td><strong>June 2011</strong></td>
</tr>
<tr>
<td>Directors approve redesign project</td>
</tr>
<tr>
<td><strong>Early June 2011</strong></td>
</tr>
<tr>
<td>User research with two sites, one radiologist, and Chief Resident Radiologist</td>
</tr>
<tr>
<td><strong>Late June 2011</strong></td>
</tr>
<tr>
<td>Ideation and conceptualization (two weeks)</td>
</tr>
<tr>
<td><strong>June 22, 2011</strong></td>
</tr>
<tr>
<td>Expert review via WebEx using Fireworks mockups to step through screens; subsequent reviews via email</td>
</tr>
<tr>
<td><strong>July 7, 2011</strong></td>
</tr>
<tr>
<td>Review work with original developer</td>
</tr>
<tr>
<td><strong>August 2011</strong></td>
</tr>
<tr>
<td>Phase 2: Release to one teaching hospital</td>
</tr>
</tbody>
</table>

Although the redesign changed the front end, the developers were able to re-use most of the back-end code written for the first version. VitreaTeach also makes use of layouts developed for VitreaView, a related product; for example, the VitreaTeach Case View layout is simply a modified version of the same screen in VitreaView. Since the VitreaView design was already optimized for lower resolutions such as 1024x768, these layouts work well on the iPad. Sadikali explains, “The application is web-based so you can access it on a tablet through the browser, and we also adjust it specifically for different platforms, like Android. We’re planning some tweaks to increase the button size and row sizes. Some of them were already large; but some areas need improvement. We’re working now on refining it for tablet use.”
LESSONS LEARNED

The process of creating VitreaTeach reinforced an important principle of user-centered design: There is no substitute for working directly with users. Even if you know a lot about the users in other contexts, if the activity type is new (such as learning rather than practicing medicine,) it’s easy to make incorrect assumptions about what’s important and how people behave. Sadikali says, “Start from the beginning, and don’t try to skip the design process. It’s something you need to do every time.”

TEAM

VitreaTeach was initially conceived and developed by Thomas Myers, a developer working out of the marketing department. Navid Sadikali, a user experience designer, led the redesign project.

Navid Sadikali is a user experience designer at Vital Images. He has an undergraduate degree from Brown University and a Masters in Computer Science from the University of Waterloo. His focus is on interaction design and he manages visual designers to cohesively bring the entire user experience together.

“What motivates me is helping analytical-thinking businesses (medical, bio-tech) grow using design. I believe leapfrogging the competition requires involving professional designers in product strategy rather than having design be a downstream activity you do once everything (markets, customers, requirements) is figured out.”

Thomas Myers (Senior Manager, Strategic Technical Marketing, Member of the Office of the CTO—Vital) is a technology entrepreneur who is passionate about building products and services that make a lasting difference in the lives of patients.

“VitreaTeach will produce better trained radiologists who will be empowered to deliver better care to us all and I am very proud to have been a part of the team incubating it.”

Navid Sadikali
Wakelicious

Organization: Utopian Army

Application: Alarm clock for friends to wake each other with YouTube videos

Headquarters: Eindhoven, The Netherlands

www.wakelicious.com

OVERVIEW

Main function: Wakelicious is an alarm clock that wakes you up by playing YouTube videos sent by your friends.

Why it’s unique: Wakelicious is a great example of lightweight design because it makes a complicated process feel simple and intuitive. Instead of long explanations, the application uses structure and action labels to help users understand how it works. It makes hearing a wake-up alarm more fun by involving friends in the experience.

The organization: Wakelicious is the first product released by Utopian Army, a startup software company.

WHO ARE THE USERS?

Wakelicious is targeted at people who already use Facebook to interact with friends and family, and are interested in sharing music with friends and discovering new music. Although it was developed in the Netherlands, most Wakelicious users are in the United States. The application is currently only available for iOS mobile devices, and is used by more than 1,000 people.

THE APPLICATION

Wakelicious makes the moment of waking up a shared experience with a friend; instead of a generic alarm sound, users start the day with a message or song sent to them by a friend. As one Wakelicious user put it, “Using Wakelicious is like receiving a present every day.”

Users can set their Wakelicious alarm clock and when that time arrives, Wakelicious automatically plays the first video in their wakelist, or video queue. Users can also send videos to their friends’ alarms, view their history of sent and received videos, and share their favorite videos on Facebook.

Wakelicious in a nutshell: a social alarm clock.
The queue indicator shows whether a user has any videos waiting, but Wakelicious is designed to keep the experience of waking up a pleasant surprise by keeping the titles and senders of videos hidden until the alarm goes off. (If no one has sent you any videos, Wakelicious will play a video from a list built by the designers.)

Rather than trying to build an entire social network and content repository from scratch, Wakelicious leverages two well-established entities, Facebook and YouTube. When users start Wakelicious, they are prompted to link the application to their Facebook account. From that point, the content from both Facebook and YouTube is tightly integrated into the Wakelicious experience. From within the app, you can browse your Facebook friends list and choose someone you’d like to send a video to. Once you pick a friend, you are taken to a screen within Wakelicious where you can search for a YouTube video to send.

Because Wakelicious integrates with these well-established sites, the learning threshold for users to begin using Wakelicious is much lower. Their friends and a huge number of videos to choose from are easy to access from right within the application.

Integrating with Facebook also offers the potential for marketing the application to potential new users by posting to existing users’ Facebook pages, but Wakelicious takes a cautious approach to this goal. An early prototype was set up to post a comment to the user’s Facebook wall each time Wakelicious played a video, but this behavior alienated users and was quickly removed. Mike Holenderski, who developed Wakelicious, explains, “We don’t want to spam our users. We wanted to make an app that we would personally want to use. We don’t like those apps that spam your wall or your email account, so we didn’t want to do this to our users either. It’s about long term vs. short term benefits. Maybe from the marketing perspective it would be better to be more aggressive, but it’s better in the long run if the application doesn’t spam users.”
Wakelicious plays videos sent by your friends to wake you up.
Users can choose videos from YouTube and send them to their Facebook friends.

SIMPLICITY

One of the biggest hurdles for Wakelicious is simply getting people to understand what the application is for and how it works. Holenderski and Runge took a few very specific steps to simplify the Wakelicious experience. In fact, after spending three months to build a working prototype, they spend the next nine months refining and simplifying the design to make it easy to use.

Matched users’ mental models

In order to minimize the development cost, the first version of the Wakelicious mobile application could only receive videos, not send them. Users would receive videos on their phone, but would only be able to send videos via a separate Facebook application. Rik Runge, who led the design for Wakelicious, explains, “When we started, it seemed like it would be technically difficult to build a mobile app that could send videos, therefore we adapted the Wakelicious concept to
eliminate sending from the mobile app. But later on we realized that was where we got it wrong. For most users that's not a logical way to use an app. People expect to have two-way communication through the same app.”

Holenderski adds, "This is a perfect example of where we started programming too early, and the development influenced the design. Because it was technically difficult, we took the feature out. But this was a mistake because the programming is just there to support a great design: It should not be done the other way around. It's important to have a great design first.”

After doing usability tests of the working prototype with the split functionality, it was obvious that Wakelicious needed to be able to both receive and send videos from within the same interface. Adding the send feature made the application much easier for users to understand and engage with.

**Top:** The first iteration mobile app could only receive videos. Users had to go to Facebook in order to send a video. **Bottom:** Now, users can both send and receive videos from the iPhone app.

**Less reliance on instructions**

Early versions of Wakelicious relied on seven screens worth of copious instructions to explain the application, but user testing revealed that people simply skipped over the instructions, and just wanted to explore the application and start using it immediately. Runge says, “When we made it we thought that we just needed a lot of explanation to convey the concept. But we learned that people can’t get the concept via text. They understand it by seeing the buttons and what they lead to. That helps them build their mental model. They won’t read instructions.”
Because the prominent instructions weren’t actually helping users, they were shifted into an auxiliary role: still accessible for users who seek out the help icon, but no longer a main navigation option.

Instead of instructions, the design now focuses on making sure the structure and controls of the application are self-explanatory. Some contextual help is also built into the interface; for example, a brief description of the backup alarm appears just below the control for that feature on the Settings screen. An instructional screen also appears once an alarm is set, to remind users to leave the application open.

![Left: An explanation of the backup alarm feature is built into the Settings screen. Right: Once an alarm is set, instructions for leaving the application open appear.](image)

**Restructured navigation to prioritize key tasks**

Making the interface self-explanatory required a big shift in how the functionality was organized, presented, and labeled. One of the biggest changes was removing the persistent menu bar at the bottom of the screen. Holenderski explains, “Initially we had tabs on the bottom, but that’s actually better for a different kind of application, where you quickly switch between parts of the interface, independent of what you were just doing.”

Because the core Wakelicious activities (setting an alarm and sending a video) are a linear series of steps, it’s actually better for users to move sequentially through the necessary steps. Keeping the options for secondary tasks, such as adjusting the
settings, always visible just creates a distraction and more opportunities for users to get confused and go down the wrong path.

Removing the persistent navigation bar helped de-emphasize secondary tasks, such as adjusting the settings, accessing help content, and viewing the history. After removing those distractors, Runge focused on making the remaining core tasks obvious and understandable, “I came up with a two-button interface, instead of a tab bar and a button. This made the app really focus on two key features. The two buttons at the bottom are big and simple.”

The naming of the two main task buttons on the home screen buttons was critical to instantly conveying how Wakelicious works, “We talked a lot about the labels for the buttons; when you push the bottom one, you’re not actually waking a friend. But if you wrote select a video for a friend that would be less active. The two buttons on this main screen help people understand what the app is about. It lets people know that they can set an alarm and wake a friend. And they can start to understand the application.”

Once these changes were in place, users were able to pick up the application and start using it without reading the instructions. “The two main tasks, waking a friend and setting an alarm, are more important than the history or settings. So those buttons are smaller, the text is smaller. Seeing the time is important, so that’s big. So getting the size of elements right helps people understand the application. When we added this step, people started to get it, and then the drill down interface guides them through the sending process,” says Runge.

**Left:** Early versions of Wakelicious included a persistent navigation menu at the bottom of the screen. **Right:** now, the home screen has just two simple calls to action.
Wakelicious guides the user through the *Wake a friend* task with a linear pathway which presents each step in sequence. **Left:** First, users see their list of friends. **Right:** After choosing a friend, the user is taken to a screen search for videos.
**Left:** After selecting a video, users have the option to play to preview the video and make sure it’s the right one. **Right:** When watching the video preview, the *Done* button returns you to the previous screen.
**Left:** Once you tap *Send Video*, you see a confirmation message. **Right:** An animated graphic is displayed while the video is being sent.
Once the send process is complete, the user is returned to the home screen and a Sent confirmation is displayed.

Removing the navigation bar to simplify the task options was a vast improvement, but there is such a thing as going too far with simplicity. Due to technical constraints, Wakelicious at first did not have a snooze function. This turned out to be another instance where user needs should be prioritized over development constraints, however, because people really wanted to snooze. Wakelicious now includes a snooze feature: when the alarm goes off, a video begins playing automatically. If you tap I’m awake the video continues playing through to the end, but if you tap Snooze the application returns to the clock screen and then launches the video again nine minutes later.
The snooze feature will cancel the video, and then re-launch it 9 minutes later.

**INTERFACE ELEMENTS**

Wakelicious uses simple and familiar UI elements, and follows Apple’s Human Interface Guidelines. Large buttons and fonts make it easy for users to navigate between screens, and the dark background color is more comfortable for night viewing. Runge explains, “We’ve tried different visual styles...We wanted something remarkable, but being remarkable often gets in the way. We ended up with an interface that doesn’t get in the way. It’s there, but it’s not the main attraction of the application.”

In the History view, buttons at the bottom of the screen let you toggle between Received and Sent videos. When an interface has buttons to switch between two states, it is sometimes difficult to tell which state is currently active, because the visual treatments of the two buttons don’t clearly look raised or depressed. The shadow visual treatment on the Wakelicious Received and Sent buttons, however is dark enough that the currently selected button does actually look depressed.
The visual treatment of the *Received* and *Sent* buttons is strong enough to clearly indicate which is currently depressed.
INFO@NNGROUP.COM

Executive Summary

Left: Standard interface controls such as those for setting the alarm time are familiar to iOS users. Right: The *Slide to cancel* control for canceling the alarm resembles the standard iOS unlocking control.

**DESIGN PROCESS**

When they started the project, Holenderski estimated it would take about two weeks to build Wakelicious. But after quickly putting together a working prototype and testing it with users, the team realized that making this unusual functionality understandable and easy to use would require a much more extensive investment in design. “In order to save time for the customers, we had to invest this huge amount of time. This is our third or fourth iteration of the design, because our initial designs were just too complicated. We now have much more appreciation for simple applications. Just because it looks simple doesn’t mean that there’s not a lot of work behind it. Usually the designers put a lot into making it simple,” says Holenderski.

The first version had actually been built in code before user testing revealed the deep conceptual and structural problems. Though it was painful, Runge and Holenderski went back to square one and started over with paper prototyping.

“With the first version, we had a working prototype very early. This was actually a mistake because we hadn’t worked out the details of the interface yet, so we ended up needing to do a lot of redesign. We started programming much too early. From then on, we did more iterations of the interface in paper prototypes and wireframes before programming,” says Holenderski. Their subsequent prototyping efforts focused on the processes unique to Wakelicious, such as sending and receiving videos, to make sure these were understandable.
Usability Testing

Usability testing was an invaluable part of the Wakelicious design process. “After a while you get used to the flaws in the design. You lose the general overview and tend to focus on details. It takes another set of eyes to pull you out. At this stage you will get most insights by showing the work to other people through user testing,” explains Runge.

The Wakelicious interface was tested with users at several different stages of development, including

- interactive prototype (built in code)
- paper prototypes
- keynote presentations.

Because Wakelicious is intended for a broad audience— anyone with a mobile phone, a Facebook account, and an interest in music— finding participants for usability studies wasn’t difficult, as Runge explains, “We tested it on our family and friends and also with people who are really into technology. One guy worked at Google, and we thought, ‘He’ll understand it.’ But he didn’t. At first he totally dismissed the manual and started playing around. At that point we realized there was something wrong with the way we designed it.”

Simple, informal tests with friends and family interacting with paper prototypes yielded important design guidance, “We learned that users dismiss anything that looks like a lot of text, and go straight to playing with the buttons and interface.

Once we saw that, we realized that the application had to be easy enough for people to get the idea gradually,” says Runge.

To refine the user interface, designers conducted usability tests on paper prototypes with friends and family.
Implementation

Wakelicious was designed using Adobe Illustrator to create visual designs, and Axure and Keynote to build interactive prototypes. The actual application is a native iOS application built with the iOS SDK. The native application platform was necessary to minimize load times and include functionality that simply wasn’t accessible through a web platform. In order to make the application function quickly and simply for users, the backend is set up to perform some actions concurrently, such as loading video titles and thumbnail images.

iOS Constraints

The iOS provides a large potential audience, accessible documentation, and built-in distribution method. But this platform created some complications and functionality constraints. In order to automatically play a video at a certain time, Wakelicious must be open and running in the foreground of the device. Although iOS does allow applications to stay open in the background, the actions an app can take while running in the background are limited to a few basic things like continuing to play a song. They can’t play a YouTube video, because background applications can’t connect to the internet. Only Apple’s own alarm clock built into the iOS system has the ability to launch at a specified time even if it wasn’t open in the foreground.

A few different features are built into Wakelicious to work around this constraint. First, users are advised when they set an alarm that they need to leave the application open. In order to conserve battery life overnight, Wakelicious also instructs users to turn the device face down, and uses the proximity sensor to turn off the screen.

Since waking up at the right time is potentially a really important task for users, Wakelicious also includes a backup alarm, in case the application is accidentally closed or the Internet connection is lost. If Wakelicious isn’t open and running at the time the alarm is set, the backup alarm generates a local notification and plays a backup alarm sound. When the user acknowledges that notification, it triggers Wakelicious to open and automatically start playing a video.

Facebook and YouTube Constraints

Integrating with Facebook and YouTube provided significant advantages in terms of access to content, but also introduced difficulties and constraints.

When relying on external services, such as Facebook and YouTube, one becomes dependent on changes to their APIs. For example, about two months after launch Wakelicious experienced downtime due to a change in Facebook’s authentication API. Holenderski has therefore subscribed to alerts about changes to the relevant APIs to keep up-to-date. In addition, Wakelicious backend servers continuously monitor the health of the system and generate alerts when something goes wrong.

The YouTube API is more stable, but because of the video format provided by the API, Wakelicious cannot download a video in advance and then play it later. For now, Wakelicious relies on the standard YouTube video format and advises users to make sure their wireless or 3G connection is active.

YouTube also sometimes removes videos due to copyright violations. Wakelicious is set up to automatically confirm that a video you’ve received is available, and if not, remove it from your queue and play the next available video.
## PROJECT TIMELINE

<table>
<thead>
<tr>
<th>Month</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2010</td>
<td>V0.1:</td>
<td>Interface design</td>
</tr>
<tr>
<td>October 2010</td>
<td>V0.1:</td>
<td>Working prototype</td>
</tr>
<tr>
<td>November 2010</td>
<td>V0.1:</td>
<td>User testing (with working prototype)</td>
</tr>
<tr>
<td>January 2011</td>
<td>V0.2:</td>
<td>Interface design (simple black, drill down, no manual)</td>
</tr>
<tr>
<td>February 2011</td>
<td>V0.2:</td>
<td>Wireframe prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Working prototype</td>
</tr>
<tr>
<td>March 2011</td>
<td>V0.2:</td>
<td>Paper prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User testing (with paper prototype)</td>
</tr>
<tr>
<td>April 2011</td>
<td>V0.3:</td>
<td>Keynote prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Paper prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User testing (with paper prototype)</td>
</tr>
<tr>
<td>May 2011</td>
<td>V0.3:</td>
<td>Working prototype</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Beta testing (with working prototype)</td>
</tr>
<tr>
<td>June 2011</td>
<td>V1.0:</td>
<td>Submitted to AppStore</td>
</tr>
<tr>
<td>June-August 2011</td>
<td>V1.1-3:</td>
<td>Bug fixes</td>
</tr>
<tr>
<td>October 2011</td>
<td>V1.4:</td>
<td>Added snoozing</td>
</tr>
</tbody>
</table>

## LESSONS LEARNED

The Wakelicious team shared a few lessons learned from this design effort:

- **Testing with users is essential.** "You may think you have a good idea about what will be easy to use, but you need to get your idea out, prototype it and put it to the test. If the prototype fails to do what you have hoped for, i.e. when you see users dismiss your whole interface or things you think are important, you will learn a ton about how to improve the user experience."

- **Fail early.** "The sooner you gain insights into how people perceive your interface, the less effort it will cost to fix the design. Failing seems bad, but those insights are so valuable. Working with paper prototypes allows you to solve problems much more efficiently than during development."

- **People don’t read instructions.** "People do not care for manuals—the design must be self-explanatory. Our initial design required a lot of
explaining our instructions; it just didn’t work.”

• **Consider building for multiple platforms.** “If we had decided earlier that we wanted to do it for multiple platforms, we would have used something like PhoneGap, or created our own framework that would allow us to port the app much more easily to another operating system. Those decisions have to be made early in the implementation.”

• **Design first, code later.** “Anxious to get started swiftly, we made some decisions from a developer’s perspective. It was only until later, during user testing, we saw how these decisions had influenced the user experience. So take note that programming too early is likely to compromise your design decisions.”

**TEAM**

Wakelicious was built by Utopian Army, which is a startup created by Rik Runge and Mike Holenderski. Rik developed the concept, crafted the design and evaluated the interface to ensure the app’s ease of use. Rik also works as an interaction designer at Mirabeau, a full service internet company in The Netherlands, focusing on mobile development and user testing. Mike is responsible for the technical part of Wakelicious, from app development to server maintenance. He is also employed at the Eindhoven University of Technology, where he is currently finishing his PhD in computer science in the domain of real-time systems.

Rik Runge (left) and Mike Holenderski (right) make up Utopian Army, creators of Wakelicious.
Workflow Applications

Sharing and approvals are necessary and overall improve content and keep processes moving. But tracking and maintaining the integrity of content, and communicating status throughout the editing or status-changing cycles has presented a difficult design challenge for many years. Who has the baton now? What does he need to do? What is he allowed to do? Where does it go next? Who should be involved, and when? What if that person is not available? The question designers must consider and answer seem to be endless, yet the great designers of the applications in this section ferreted out all of the answers swimmingly.

Workflows may add levels of politics, security, and intricacy depending on who is sharing the content, such as sharing between an organization’s workers, an organization and its end users or customers, or B2B. Additionally, if there are different user types in varying job roles, the interface needs to present different views tailored to each context. And if diverse organizations or individuals actually own the content, or nobody actually owns the content, these can bring in even more complexities.

Designers working on these applications must thoroughly examine the workflow structures for the tasks, users, and content in order to build great applications.

Some UI elements that may behoove people using this type of application include:

- Process steps (sometimes numbered) and progress indications to define stages in the process
- Ability to comment and otherwise deviate from the process flow when absolutely necessary, in the name of getting the tasks done and not being beholden to a structured UI
- Wizards and linear information paths
- Pop-ups and tips within and throughout the process
- Previews
- Clear display of content properties such as creation time, person, team; and what has been done thus far.

Winning applications in this category include:

- Climate Action Planning Tool, helps research campuses and universities reduce carbon emissions.
- eReview, an intranet application that integrates complex business functions into the organization’s intranet.
- Eventbrite, a self-service software tool used by event hosts to organize, sell and manage tickets for events. The Create Event workflow described in this report is the process used to set up a new event within the Eventbrite system.
- Hobsons CRM product, product suggestion portal tool for the educational institutions Hobsons serves: gives its customers an opportunity to make comments/suggestions.
Climate Action Planning Tool (NREL)

**Organization:** National Renewable Energy Laboratory

**Application:** Carbon reduction calculator

**Headquarters:** Golden, Colorado

**www.nrel.gov**

**Design team:**
- Ruby Nahan, Communications Representative/Project Manager
- Rachel Sullivan, Project Manager/Writer
- Laurence Hoess, Developer
- Erica Augustine, Designer
- Shauna Fjeld, Usability Consultant
- Kristin Tomley, Web Content Producer

**OVERVIEW**

**Main function:** The Climate Action Planning Tool was built to help research campuses and universities reduce carbon emissions. The application helps users create realistic plans for reaching specific carbon reduction goals, by assessing the total current emissions and calculating the impact of different possible reduction strategies.

**Why it’s unique:** The Climate Action Planning Tool was chosen as an example of a great design because it

- Helps users through a difficult task by breaking it up into steps that make sense
- Uses interactive visualization to communicate complex information in an understandable way.

**The organization:** The U.S. Department of Energy’s National Renewable Energy Laboratory (NREL) is the only national laboratory solely dedicated to advancing renewable energy and energy efficiency technologies from concept to commercial application.

NREL develops renewable energy and energy efficiency technologies and practices, advances related science and engineering, and transfers knowledge and innovations to address the nation’s energy and environmental goals.

**Design philosophy:** The design team had two main goals for the Climate Action Planning Tool: they wanted the final design to be interactive, but simple enough to be used without instructions.

**WHO ARE THE USERS?**

Management and facilities directors of research campuses who are tasked with developing a climate action plan to reduce greenhouse gas emissions are the primary target audience for this application. A typical usage scenario would be, “The user’s boss says, ‘We need a climate action plan, figure it out.’ The user may not know where to start in terms of taking a big picture approach to energy use across the campus. The tool helps them figure out, if we want to get to zero greenhouse gas emissions in 40 years, these are the areas we need to address. We can’t just look at replacing light bulbs, we have to look at all of these other areas such as transportation and building plans,” says Rachel Sullivan, the team’s project manager and writer.
This application is targeted specifically at facilities directors who manage campuses—which could mean a university or college campus, but could also be a government or private research facility. Essentially the tool is useful for anyone interested in reducing the carbon footprint of a piece of property that is:

- Composed of multiple structures
- Owned or managed by a single organization
- Used by large numbers of people.

The target audience is technically oriented, but the team’s conversations with users revealed a varying degree of knowledge about energy use and goals for greenhouse gas emissions. Sullivan explains, “There was one person who knew exactly what the goals were; another person was new in the position and really had no clue yet.” Regardless of how long they’d been on the job, much of the information needed to use the tool is so detailed that users simply won’t know it off hand. One of their biggest needs is guidance about how to start the planning process.

As a federal government organization, NREL also complies with Section 508 accessibility requirements, and considers people with disabilities or those using assistive devices as an important audience.

Potential users discover this application by visiting websites such as the National Renewable Energy Laboratory, U.S. Department of Energy Federal Energy Management Program, or Labs21 websites, or through outreach efforts such as presentations by NREL technical leads and project leaders at industry conferences.

**THE APPLICATION**

A project nearly two years in the making, the application was originally an Excel spreadsheet created by an industry expert at Cornell University. The NREL Communications team was tasked with transforming that spreadsheet into an interactive tool. The result is a web-based application that:

- Manages users’ expectations up-front with a downloadable form listing the data points users need to collect in order to use the application.
- Allows users to enter their data via an accordion menu that automatically calculates data.
- Produces a dynamically generated, printable graph that gives users the overall picture of energy use on their campus.
- Quickly gives the user an overview of what their best options are for developing a climate action plan by arranging the data so that the energy sectors are in order of those that will have the most impact on reducing emissions.
- Provides a dynamically generated summary report of all the data, graphs, and calculation results that can be used to inform climate action plans.

One of the most important things this design team did was minimize explanatory text, and instead let the functionality speak for itself. Sullivan explains, “Early on in the process we had an instruction sheet … there was way too much content. It’s easier to write more content than it is to write a small amount … but people just weren’t interested in reading instructions; they just wanted to put the data in and see what happens.”
Executive Summary

The Climate Action Planning Tool provides a structure for calculating greenhouse gas emissions and how to reduce them.

**TASK FLOW**

After testing the prototype with users, the team introduced several modifications that dramatically improved the workflow, including the following:

- Revised the overview to articulate the specific steps
- Created a separate step for collecting data offline
- Numbered the steps and placed a progress indicator on each screen.

---

**Step 3: Set, Adjust, and Re-Adjust Emissions Technology Options Goals**

The graph below shows a 40-year "business-as-usual" scenario of your campus greenhouse gas emissions based on the baseline energy consumption data you provided.

**Set Overall CO₂ Reduction Goal**

Enter the 40-year greenhouse gas emissions reduction goal for your campus. A 100% reduction means your campus will have zero greenhouse gas emissions in 40 years.

Goal: [ ] % Reduction

**Adjust Technology Options**

Enter your goals for reducing carbon emissions for each campus energy sector below.

After you calculate the first goals you entered, you can go back and re-adjust goals and years to see which options will have the most impact on meeting your overall reduction goal.

The reduction or conversion goal percentages in the pull down menus are considered reasonable for most research campuses.

---

**People and Policy**

Strategies that impact human behavior and energy policies can save as much money, energy, and carbon emissions as technical solutions. Here you can calculate:

- Space Planning and Use — this tool calculates growth reduction by a specified percentage, which is in re campus policy decision to actively manage existing space and reduce the need for building new areas.

  **Reduction Goal %** [ ] Year to Achieve Goal

- Energy Conservation — this tool calculates savings due to energy reduction in existing buildings. Scope 1: emission impacts on fuels (gas, oil, and coal) and scope 2 indirect emissions from purchased power are used to calculate energy conservation savings.

  **Reduction Goal %** [ ] Year to Achieve Goal

The Climate Action Planning Tool provides a structure for calculating greenhouse gas emissions and how to reduce them.
The Climate Action Planning Tool homepage provides a basic overview of the application, a call to action to launch the workflow, and a few links to related information. Overall this page is not complex, but during usability testing the team was able to identify some weak areas of the page and introduce improvements.

For example, while the first version of the home page included four bullet points that basically describe the four steps of the tool, the descriptions were rather vague and wordy, and the bullets actually make the copy look like a feature list rather than a sequential set of instructions.

In the updated home page, the steps are numbered one through four to indicate the linear sequence and the description of each step is shorter and describes very specifically the exact action the user will be taking: for example,

- Adjust specific reduction goals by technology option to see how they impact your overall goal

became:

3. Set and adjust emissions and technology options goals.

The updated page also moved the links to secondary information (such as Help and details about the calculation formulas) from the left navigation menu to a footer element at the bottom of the page. This is a vast improvement, because showing related links in a left navigation bar violates user expectations: users who do look to the left will be expecting to find global navigation options such as the main steps in the application; users who are interested in the secondary related links will probably look to the footer, as that is where such items are more commonly placed.

Adding icons to these links calls them out as meaningful content, but moving them to the bottom and enclosing them with a box makes it clear that they are secondary to the main focus of the page: launching the application.
The first version of the home screen was less specific about the steps and included a left-side navigation menu.

The final home screen includes four brief numbered steps, a single strong call to action, and a footer element for related links.
In order to use the application, users must first gather specific information about current energy usage at their facilities. These are the data used to calculate which technology options will lead to the most significant reductions in consumption of fossil fuels for each individual campus, and will therefore meet greenhouse gas reduction goals.

User testing showed that gathering this information can be a fairly involved process, “Often they didn't know much about the different energy sources used by the campus, so they’d have to go to other people and collect different sets of data. It was crucial for them to go out and seek help from other people on the campus that had more expertise in these different areas. That’s why we came up with a downloadable worksheet, to help you gather the information,” explains Laurence Hoess, the developer.

The worksheet was optimized to fit the way users perform the task and make it easy to transfer the data into the application, as Erica Augustine, the designer on the project, explains, “We did it in a PDF so that they can print it, and it looks exactly like the input screen so you know it will translate directly. Also, there are some people that want to just play around with the tool, so they can skip this step.”

Because collecting this information is so essential to getting value out of the application, the entire Step 1—the first screen you see when you launch the application—is devoted to the link to download the worksheet. Paradoxically, the application is much improved by adding this step to the process, even though the “step” is actually a wholly off-line activity, and the only thing for users to actually do on this page is print out the PDF form.

In the first version of the design, the initial screen was called Before you start; it described the need to gather information but it appeared to be instructional text and many people skipped over it. Access to the form for collecting information was a text link within a paragraph. This confused users, as Shauna Fjeld, the usability consultant on the project, explains, “People just wanted to dive in and use the application, but it doesn’t work that way; you need to collect data first. So we changed the design to make it very apparent that you have to gather data before you can use the tool.”

As soon as you start the application, a progress indicator appears showing which step you are currently on, and how many steps remain. This indicator appears on all subsequent pages, and is especially helpful for this type of process where you expect people to move backward and forward through the steps, checking their work and making changes.
Step 1 introduces a progress indicator and prompts the user to stop and gather the energy use information specific to their institution that they will need to use the application.

The first version of the first screen appeared to be just instructional text. Users thought they could just skip it.

After collecting all of this information, the user can input it into the form provided in *Step 2: Enter Baseline Energy Consumption Data*. This form automatically calculates...
the total carbon output from both indirect and direct emissions, based on subtotals provided by the user.

During development, Hoess built a function to populate the application with sample data in order to make it easier to test the application without having to make up a whole data set repeatedly. Once the application was complete, he removed this function but users asked to have it added back in. It turned out that people wanted to try out the application and see how it worked before they invested time in collecting all the up-front information, and found the pre-populate feature helpful for ‘test driving’ the application to get a sense of how it works and what they would be able to do with it.

Step 2 also includes a bit more explanation for the last two input items (Growth Rates) since testing revealed some confusion about that terminology. Now, people can click on the What’s this? link next to those items to see contextual help information.

The data input form in Step 2 looks almost exactly like the data collection worksheet, with the addition of Total fields (which are automatically calculated.)

<table>
<thead>
<tr>
<th>Scope 1: Emissions (Direct Combustion)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T CO₂e = Tones of Carbon Dioxide Equivalent*</td>
<td></td>
</tr>
<tr>
<td>Gas (T CO₂e)</td>
<td></td>
</tr>
<tr>
<td>Oil (T CO₂e)</td>
<td></td>
</tr>
<tr>
<td>Coal (T CO₂e)</td>
<td></td>
</tr>
<tr>
<td>Fleet (T CO₂e)</td>
<td></td>
</tr>
<tr>
<td>Total (T CO₂e)</td>
<td></td>
</tr>
</tbody>
</table>

*Metric tons of CO₂ equivalent or 2.204 lbs of CO₂.

<table>
<thead>
<tr>
<th>Scope 2: Indirect Emissions (Purchased Power)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MWh</td>
<td>Lbs, CO₂/MWh</td>
</tr>
<tr>
<td>Vendor 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Scope 3: Other Indirect Emissions (Transportation)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Commuters</td>
<td>T CO₂e</td>
</tr>
<tr>
<td>Business</td>
<td>T CO₂e</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Growth Rates</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Buildings</td>
<td>0% compounded over 40 years</td>
</tr>
<tr>
<td>Commuter</td>
<td>0% compounded over 40 years</td>
</tr>
</tbody>
</table>
FLEXIBILITY

The first few steps of the Climate Action Planning Tool are very straightforward, but when users arrive at Step 3 they are confronted with a complex set of controls and choices.

At this point in the process, users have entered the existing carbon emissions at their facilities, and it’s now time to start figuring out how they can reduce the amount of greenhouse gas emissions.

The calculations represented on this screen are complex, but several significant design elements have been built in to simplify the process and give users the flexibility to control the calculations.

First, the nineteen different inputs are divided into six major categories, grouped by topic such as Transportation and Renewable Electricity. This categorization allows the interface to present the user with an organized list of understandable titles, rather than an overwhelmingly long page. The sequence of categories is also intended to
subtly guide users towards more realistic and effective strategies for reducing carbon emissions; goals related to energy efficiency such as optimizing the use of existing space, conserving energy, and transportation come first in order to encourage users to consider these as their primary tactics, before considering options that can be more expensive (such as converting to renewable energy) or less direct (such as carbon offsets.)

The categories are presented in an accordion with the first category expanded by default — an absolute requirement with accordion or collapsible interfaces, since people easily overlook such elements if they are all collapsed.

After making selections in the first category, a Next button allows the user to close that category and automatically opens the next group.

This provides an easy and straightforward path for walking through the data entry points. But the interface also makes allowances for users who want more freedom to move back and forth between categories, or even more than one category at a time.

The colored title bar of each category acts as a button to collapse or expand that module, and the small triangle on the far left rotates to indicate whether the module is currently collapsed or expanded. An especially nice feature of the design is that the entire colored title bar is clickable, rather than just the triangle or just the label. This provides a much larger target area for the mouse, making it easier for users to click on the control.
Step 3 lets users input their specific reduction goals; different types of reductions are divided into separate categories such as Transportation and Renewable Electricity.
People and Policy

Strategies that impact human behavior and energy policies can save as much money, energy technical solutions. Here you can calculate:

**Space Planning and Use** — this tool calculates growth reduction by a specified campus policy decision to actively manage existing space and reduce the need for

**Reduction Goal %**  
**Year to Achieve Goal** 1

**Energy Conservation** — this tool calculates savings due to energy reduction in emission impacts on fuels (gas, oil, and coal) and scope 2 indirect emissions from calculate energy conservation savings.

**Reduction Goal %**  
**Year to Achieve Goal** 5

Buildings

Research campus buildings often represent the most cost-effective opportunities for reducing calculate:

**New Buildings Design** — this calculation assumes that new campus buildings will existing campus buildings. You can set a goal to reduce energy use and resulting below the percentage of energy used by your existing buildings. This reduction is direct emissions and scope 2 indirect emissions.

**Reduction Goal %**  
**Year to Achieve Goal** 1

Transportation

Energy Sources

Renewable Electricity

Carbon Offsets and Certificates

Users can open more than one module at a time; the entire colored bar is a clickable button to toggle the state between open and closed, and the triangle next to each module title rotates to indicate the current state.
INFORMATION DISPLAY

After entering the reduction goals and clicking the Calculate button at the bottom of the screen, the graph in the upper right corner displays a visualization of the total amount of greenhouse gas emission reductions over time.

The graph also divides up the total greenhouse gas emissions by category, to show how much of the contribution comes from the different components of the reduction strategy—extremely helpful information for policymakers who may be balancing budgets and trying to decide what to spend their money on.

The colors were deliberately chosen to correspond with the colors used on the main NREL website; in addition to color coding the graph categories, the legend is placed right next to the graph in order provide a vertical label listing, which corresponds to the order of elements on the graph — rather than relying exclusively on color to distinguish which category is which.

Since users are likely to want to experiment with different reduction goals and strategies, this screen retains all the inputs for selecting goals. The first version of the application actually placed the graph at the bottom of the screen, rather than the top; this presented huge discovery issues for users, who never noticed the graph preview at the bottom of the screen. Hoess explains, “The first layout had the graph at the bottom of the page and all the inputs at the top; we found that users weren’t seeing that as they made changes they were reflected in the chart below. So we had to redesign the layout to help make that connection between the inputs and the graphic.”

Once the targets have been entered and calculated, the graph displays a visualization of how much carbon emissions will be reduced over time.
The first version of the application placed the graph at the bottom of the screen, where it was easily overlooked.
After adjusting the goals to their liking, users can create a formal summary of the plan. This step is important because the users are not actually the final consumers of the application output, “Users also need a way to report back the information to their management teams, a way to take the information that’s gathered in the tool and incorporate it into their climate action plan or a presentation,” explains Sullivan.

The final summary report includes a recap of all the information entered into the tool, the graphs, and a list of the numerical outputs, and links to additional information about climate action plans on the NREL.gov website. In addition to a print function, there are options to download each of the graph images for use in reports or presentations.

The *Generate Summary Report* button appears after the first calculation graph has been generated.
The final report can be printed as a PDF document, and individual graphs can also be printed or downloaded.
The application was built by members of the NREL communications team, which operates at the direction of internal clients: technical experts who are also employed by NREL. The team took a user-centered design approach to developing this application, which included:

- User profiles
- Task analysis
- Sketching
- Usability testing of prototypes.

The project was kicked off with a brainstorming meeting that brought together design, development, writing, usability experts as well as the internal client to provide direct technical information. The ideas generated at this session provided a great starting point for the project, but as the team discovered there is no substitute for talking to actual users. Sullivan explains: “Typically on other projects we do usability testing early on, but in this project we had a deadline to get a beta version
ready for a conference that was happening in three months, so we didn’t have time for user testing. So for the first version, our design prototypes and content were just our best guess at what we should present to users, which reinforces the importance of usability testing; we completely changed the tool after testing it. You don’t want to create something that people won’t or can’t use.”

**Usability Testing**

After creating the first version of the application, the team evaluated it by conducting usability tests with four different campus facilities managers. The internal project client helped recruit participants through his industry contacts, and since they were located in different cities around the country, all sessions were conducted remotely. Users were asked to complete specific tasks with the interface and the design team observed the sessions via LiveMeeting.

Each usability session was one hour long, and, overall, the usability testing phase took about 20 hours for planning, testing, and analyzing findings. Even testing with this very small sample of users yielded great design feedback. There was a lot of repetition of themes between the different users, and the designers were able to turn these into concrete interface improvements.

**Implementation**

The project was stretched over several different phases—and led by three different project managers in total, due to shifts in staffing and resource allocation. This is far from ideal but fortunately during the last phase, the writer who was already a team member was able to step into the role of project manager, avoiding the need to completely start over with a new person.

Aside from personnel changes, the team faced significant difficulties with implementing the complex mathematical calculations needed for the application, as Hoess explains, “From a development standpoint, it was pretty challenging to take all the calculations from the spreadsheet and convert them into formulas that we could use on the HTML page. I had to reverse engineer the spreadsheet ... For each formula I created unit tests ... which are a way to testing small pieces of code, to make sure it’s still performing the way you expect. Sometimes you have to go in and make a change to a program, and it accidentally affects your output, but without the unit test you might not realize that the output had changed.”

Transforming the calculation results into an appropriate visualization was another implementation challenge, especially because most graphing programs are good at straightforward data but this chart needed to show both a projection of current emissions and subtract reduction goals from that to show the difference in emissions if those goals are achieved. “The graphing tool is good at doing basic graphics, but because this application is sort of working backwards, I had to think differently about how to build the visual display for the end user,” explains Hoess.

The application is primarily built with ColdFusion and JavaScript, with a jQuery plugin to generate the graph. The first jQuery plugin used to generate the graph had several limitations, including difficulty in lining the legend up with the graph elements in order to comply with Section 508 accessibility requirements. Hoess ended up switching to a different plugin. “I found another tool that added a lot more animation and some nicer options, like the ability to export images.” The unit tests
created to validate data calculations were especially helpful in switching between graphing plugins, to verify that the formulas were still working appropriately.

<table>
<thead>
<tr>
<th>PROJECT TIMELINE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>February 2010</strong></td>
</tr>
<tr>
<td><strong>June 2010</strong></td>
</tr>
<tr>
<td><strong>July-August 2010</strong></td>
</tr>
<tr>
<td><strong>September 2010</strong></td>
</tr>
<tr>
<td><strong>November 2010</strong></td>
</tr>
<tr>
<td><strong>December 2010–March 2011</strong></td>
</tr>
<tr>
<td><strong>March-June 2011</strong></td>
</tr>
<tr>
<td><strong>June 2011</strong></td>
</tr>
</tbody>
</table>

**LEssonS LEARNed**

The NREL team shared a few lessons learned from this design effort:

- **Plan for change.** “Include enough room in your budget to implement changes identified in usability testing; we had to go back and negotiate more money to implement changes from usability testing; It’s better to establish metrics and a budget to track those metrics,” says Sullivan.
- **Test early, test often.** “We might not have had as many drastic changes between phase one and phase two [had we tested earlier],” says Augustine.
- **Test with actual users.** “When you have your colleague sitting next to you... they can point out some areas of confusion but it’s not the same [as watching a user,]” says Sullivan.

**team**

The Climate Action Planning Tool design team included:

- Ruby Nahan, Communications Representative/Project Manager
- Rachel Sullivan, Project Manager/Writer
- Laurence Hoess, Developer
- Erica Augustine, Designer
- Shauna Fjeld, Usability Consultant
- Kristin Tromly, Web Content Producer

**Erica Augustine** is a senior graphic designer at NREL with more than ten years of experience. She is the lead Web and user interface designer for NREL’s
Communications Office, where she focuses on designing clear and consistent user experiences for websites and web applications developed for NREL and for the Department of Energy Office of Energy Efficiency and Renewable Energy. Erica enjoys refining user interfaces for web applications, studying best practices and patterns for interface design, and observing user interactions with her work. She has a bachelor’s degree in Graphic Design and a master’s degree in Marketing.

**Shauna Fjeld** is a senior web developer at NREL with more than 15 years of experience in web application development. She works currently as a business analyst and user centered design specialist for web applications, including audience research, task analysis, and interaction design. She translates client needs to technical requirements and works with the technical team to implement them. Shauna’s passion is making web applications intuitive for end users. She has a bachelor’s degree in Journalism and a master’s degree in Information Systems.

**Laurence Hoess** is a senior software developer with more than 10 years of experience designing and building web applications. He has a special interest in usability and user interface design. Before turning to software development, Laurence was a technical writer focused on software documentation. In his free time Laurence likes to fly fish the rivers and streams of Colorado.

**Rachel Sullivan** is a senior communicator at NREL with more than 12 years of experience in communications, marketing, and public relations. Rachel has extensive background in strategic communications, communications planning, project management, and both print and electronic communication. At NREL Rachel is an office-wide leader in user-centered design and usability testing. She is committed to quality and uses her organizational and project management skills to help improve every project she works on. Rachel has a bachelor’s degree in English.
### TEAM ROLES

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communications Representative</td>
<td>• Interface with client, technical lead/expert, and manage client expectations</td>
</tr>
<tr>
<td>Project Manager</td>
<td>• Develop project goals, schedule, and budget</td>
</tr>
<tr>
<td></td>
<td>• Manage project team</td>
</tr>
<tr>
<td></td>
<td>• Provide updates to the communications representative</td>
</tr>
<tr>
<td></td>
<td>• Lead user-centered design process</td>
</tr>
<tr>
<td>Designer</td>
<td>• Responsible for mapping out how users will interact with content and functionality</td>
</tr>
<tr>
<td></td>
<td>• Responsible for validating design choices with user testing</td>
</tr>
<tr>
<td>Developer</td>
<td>• Implementing designs, testing, and bug fixing</td>
</tr>
<tr>
<td>Usability Consultant</td>
<td>• Advises on user-centered design and technical solutions</td>
</tr>
<tr>
<td>Writer</td>
<td>• Writes all content per NREL’s style guide</td>
</tr>
<tr>
<td>Web Content Producer</td>
<td>• Reviews content and application overall to ensure consistency with NREL brand and web content guidelines</td>
</tr>
</tbody>
</table>

The Climate Action Planning Tool design team included: Rachel Sullivan, Shauna Fjeld, Laurence Hoess, and Erica Augustine. Photo by John De La Rosa, NREL.
eReview (IMF)

**Organization:**
International Monetary Fund (IMF)

**Application:**
eReview, an intranet application that integrates complex business functions into the organization’s intranet

**Headquarters:** Washington, DC (USA)

**Design team:**
eReview was originally developed in-house at the IMF. The new interface (June 2011) was developed by an external agency, Eastbanc Technologies.

**Members:**
- **In-house:** Martin Mühleisen, Assistant Director, SPR (business sponsor;)
  Deb Reilly, Division Chief, Information and Knowledge Management Division (TGS;)
  Luzmara Monasi, Senior Research Officer (acted as business project manager;)
  Hari Maddineni, Team Leader Desktop@IMF team (team responsible for SharePoint integration, development and support;)
  Helene Faures, IT Project Manager;
  Rajitha Devineni, Business Analyst;
  Vera Rhoads, User Experience Specialist;
  Bassel Kateeb, User Experience Designer;
  Tushar Parekh, SharePoint Developer;
  Vijay Challa, SharePoint Architect and John Valter, QA Specialist.

- **Eastbanc Technologies:** Wolf Ruzicka, CEO Eastbanc Technologies;
  Slava Koltovich, Technical Partner;
  Alexander Mandrik, Senior Developer and Evgeny Popov, Head of Microsoft Department.

**OVERVIEW**

**Main function:** The International Monetary Fund (IMF) regularly generates reports on individual countries’ current economic situations, which go through several rounds of internal review. eReview is designed to carry out this document review process on an integrated collaborative web platform, rather than by exchanging drafts over email. eReview has streamlined and modernized a core business process by giving users one-click access to instructions, documents, and comments across an extensive workflow.

**Why it’s unique:** While working through a very complex review process, users are guided by a review status bar (located at the top of the UI) that provides a reference so that they always know what has been done and what is left to do. Users can perform their tasks easily using intuitive buttons. Buttons are customized for different stages and user profiles, so a user only sees the actions available to him at this specific point in the review.

**The organization:** The IMF is an organization of 187 countries created to foster global monetary cooperation and help secure economic and financial stability around the world.

**Design philosophy:** The team’s design philosophy was: simplify, simplify, simplify. Streamlining and simplifying the review process was the main goal of eReview when it was first conceived and was also the driving force behind the most recent UI redesign. The goal for the project was to put all the information required to conduct a review in one single location, allowing access to instructions, documents, and comments in one click, and allowing users to perform actions with ease. The new interface accomplishes this and more. It is now so intuitive that the organization no longer needs to provide user training.
**USERS**

The user base for eReview is broad and encompasses economists, legal staff, and administrative personnel from all departments involved in the review process (a total of thirteen departments plus IMF management.) The app is at the center of everyday life for IMF staff. Each country report goes through the following stages: A team of economists goes on a visit to assess a country, and produces a Staff report upon their return. The country team sends the report to other departments (such as the legal department) for review. These other departments post their comments, and then the original authoring team incorporates those comments into the document. The report is then finalized and sent to management for approval, then finally presented and discussed at the Board.

So a report is touched by a wide range of people during the course of a review cycle. The following list shows how the user groups handle the document as it goes through the different workflow stages:

- **Administrative staff:**
  - Initiates a review process
  - Uploads the document(s) to eReview

- **Economists:**
  - Posts comments: some via administrative staff and some directly themselves
  - Checks status: Economists often use eReview to check the status of the document, find out where it's at in the workflow, monitor comments, etc.
  - Uses previous reports for reference: Economists also use previous reports and associated comments as reference materials for their own comments on current reviews.

- **Legal Staff:**
  - Reviews and comments on reports to ensure compliance

**THE APPLICATION**

eReview takes a myriad of steps between several people and teams, and whips them into a clear, intuitive, manageable workflow process accessed from a central location. This helps not only the reviewers, but also the IMF’s office of internal audit, which at year-end counts how many reviews were produced and who was involved in each. Without a central, consolidated application, this audit process was time-consuming and potentially error-prone. In general, eReview is a one-stop-shop for all, where users can read instructions, upload documents, post comments, and identify a document's current status in a single place.

Users may access eReview by clicking an icon, but most users click on the link from within the email notification automatically generated by eReview when a report review workflow is initiated.

On the eReview landing page, users can see a list of reviews, filter the list by country, and sort the list by Status, Area Department Due Date, and OMD due date.
When looking at reviews, the right rail displays relevant calendars, contacts, and email libraries. This is especially helpful as people may need to attend meetings about the reviews, and they are all handily consolidated here.

From the landing page, users can see a list of current reviews and open a review or filter the list of reviews by country (using a dropdown list.) Calendars are handily placed in the right rail, for easy access to details about meetings related to the review.

To begin a review process, the user clicks the blue **Start Review** button in the upper right. Buttons may not be the most exciting UI elements, but let’s just state why this one is good: 1) it looks like a clickable element due to the shape, size, edges, and color treatment, 2) the words in the button, **Start Review**, imply action as well as communicating the main point of the button, and 3) it is placed in a visible location, with priority equal to the importance of the task the button supports.

eReview customizes the display of buttons based on two important criteria: 1) the step in the process, as you can do certain things at certain points, and 2) the user’s identity, as based on her role a user may or may not see certain commands. So a user who is not allowed to start a review wouldn’t see this button at all, clearing the UI of unnecessary elements and feeding to the user the information and features she does need.

Users may start a review easily by clicking the visible **Start Review** button in the right rail.

If you are part of the department who is going to kick off the workflow process (creating a review) then you begin with a form-like dialog box to assign various metadata. All of the questions are on one screen, and most are drop-down lists. This power of suggestion can make answering the form easier.
The dialog for creating a new review steps users through the process with questions and answers.

The review process itself is quite complex, involving various steps and approvals, from multiple departments and divisions. One way to help keep items in order in a scenario such as this is to present a visual workflow. In this case, it is achieved by the progress bar at the top of the eReview page. This is the anchor and shows each step of the review process, including the following: **Planned, In Interdepartmental Review, Interdepartmental Review Completed, Awaiting OMD Clearance, OMD Clea

ance Granted, and Country Review Completed.** Steps in the workflow appear in sequential order from left to right; as the steps are completed in the workflow the corresponding boxes turn green. The current step is indicated in blue and the steps that have not yet been addressed are grey. This helps users easily tell where an item is in the process. A user who wants more details will find them in the *Review Information* box in the right rail.

The *Subscribe to Alerts* button in the upper right lets users keep abreast of where the document is in the review process by requesting to be automatically notified as changes are made. Users also receive specific email notices that are automatically generated by the system when someone else assigns them a task (for example, lists them as a reviewer for a new report.)
The status bar across the top of the review page shows the current step in blue, and completed steps in green. Users can subscribe to alerts about a review using the button in the top right corner.

eReview automatically sends users a notification when they are assigned a task, such as responding to comments.

Once on the individual review page, the more pertinent information about this particular review appears in the main content area, where the review name, due date, review departments list, and comments are located. Most sections on this page are collapsed by default to enable users to focus only on the expanded content they are currently interested in, with the other content collapsed and out of the picture.

All comments are displayed under the Comments section of the individual review page, and people may print or email those comments, giving users the flexibility to keep and share comments in more ways than just in the eReview app itself. The Area Dept. Review tab is related to the area or department which created the review, or began the review process. The action buttons in the upper right present the next steps once review comments have been received: Reply to Comments, Send Email, or send to the next step in the workflow, Send to OMD (the Office of the Managing Director.)
As reviewing departments post final comments, their workflow tasks close automatically, removing the need for a separate task window. Comments can be printed or emailed from within the comments zone.

eReview has many features that streamline the review process, and take unnecessary work out of it. For example, the document review process encompasses two different workflows: a) cross-departmental: collecting feedback across multiple different departments and b) intra-departmental: the process an individual reviewing department goes through to draft and assemble comments it plans to provide to the original authors.

The tabs at the very top of the individual review page clearly differentiate between these two workflows. The first tab, labeled Area Dept. Review, is where the cross-departmental interaction is managed, and the other tabs are where the reviewers can conduct their internal departmental reviews. Thus, different actions are available in the first tab than are available in the other tabs. In the first tab, the department that initiated the review can respond to comments reviewers made, send an email to Review Departments, and send the review to the next phase (after review) in the process, which is to the Office of the Managing Director (OMD.) On the other tabs, individual reviewers may do intra-departmental action such as reassign their task to someone else in their department.
A review document has different views accessible via tabs at the top of the page; the left-most tab is one that shows the overall process, and another tab is created for each department asked to review the report (to manage that department’s internal review tasks.)

The internal department review screen displays Assigned Reviewers, Reviewers’ Comments, and Internal Workflow History, and options such as Reassign Task; departments can switch between this tab and the main Area Dept. Review tab set up by the document’s originating department.
Within the department internal review workflow, individual reviewers use the *Post Comments* dialog box to submit their work. Users are given the options to create rich text with different typefaces and treatments such as colors and bolding.

Once these comments are submitted by individual reviewers, eReview automatically notifies the main departmental reviewer (or *Country Reviewer,* who aggregates the comments and posts the official departmental comments on the main *Area Dept. Review* tab.

![Post Comments dialog](image)

**The *Post Comments* dialog supports rich text formatting.**

When individuals have posted comments in the internal department workflow, eReview automatically notifies the main point of contact for a department and describes the next step: post aggregated comments back to the original authors on the *Area Dept. Review* tab.

After comments from each review department have been addressed, it’s time for the department that originated the review to send it to the Office of the Managing Director (OMD.) At this step of the process, they can upload the document they want cleared, but all of the initial versions of the main document and supporting documents as well as comments submitted by the Review Departments are available for OMD to look at. The originating department can also send OMD an email with additional information and instructions. The selection in the status bar in eReview moves from *Interdepartmental Review Completed* to *Awaiting OMD Clearance.*
When a document is finalized, it’s submitted to the Office of the Managing Director as a link to the IMF’s document management system. The draft which was circulated for initial review, as well as supporting documents, is also present in the document zone. Note the status of Awaiting OMD Clearance, which indicates that the document has been submitted to the Office of the Managing Director, but not yet approved.

A pop-up message appears on hover and notes what the button does.

Once the document is cleared, OMD emails the initiating department that the document has been granted clearance.
The message from the OMD states who approved the document and where comments are posted.

It’s almost impossible to make a mistake using IMF’s eReview application. The workflow, reviewers, departments, and associated steps are so ingrained in the process that it is seamless. Creating, commenting on, and approving reviews is leaps and bounds better than the organization’s old system, and holds a candle to any app out there.

**DESIGN PROCESS**

IMF staff used to rely on email to share documents back and forth. Seeing the limitations of this process, the IMF looked for a solution and SharePoint seemed to do the trick. People in the user community liked the idea of eReview. They could see why it would be useful to have everything in one location. But the process of getting it in that location was the pain point.

**Using the SharePoint ‘Out of the box’ UI**

The original eReview user interface was set up using SharePoint out-of-the-box collaboration features such as document libraries and discussions. Though the online review platform and automated workflows were a step in the right direction, people struggled because the user interface was difficult, confusing and lacking. This made the whole review process cumbersome. People used the system since it was mandated by IMF management, but they were not satisfied with it. The standard SharePoint interface proved to be a functional technology solution, but it did not provide a satisfactory user experience.
Feeling the Pain

The first step toward creating a new interface was for the design team to map all the places users were going to when they needed to retrieve information and figure out all the steps they were taking to process their documents in SharePoint. This process was eye opening for the design team. “When you really write down all they [the users] needed to know to be able to use the interface, you realize the complexity of the old UI,” says Helene Faures, IT Project Manager. “People who were using eReview every day would suffer but at least would remember the steps to go through. But those who were only using eReview occasionally would forget from one time to the other.”

So, the users turned to the staff for help. “They would get confused and didn’t know what to do to perform basic tasks such as uploading a document. They could find some things, but not others,” says John Valter, QA Specialist. “It kept me busy.”

It had reached the pain threshold with support staff receiving 3-4 calls per day from people trying to use the system.

“I was forever getting calls,” says Valter. “Sometimes people would even circumvent the system. They would actually send [the document] through an email because they got confused over how to use the old interface and therefore they thought it was unreliable.”

Redesigning the Interface

It was clear that the UI needed work so the team conducted the following exercises to create the new interface and make sure that it addressed the needs (and pain points) of the users:

- **Requirements gathering**: Instead of a formal requirements gathering process the team looked to a variety of existing data sources to inform their design decisions. Through feedback received while conducting user training, support calls, and desk support visits, the team felt they had a very good understanding of the issues users were experiencing.

- **Wireframes**: The design team went through several wireframe iterations (with different layouts) and several designs (with different color scheme and styles.)

- **User reviews**: Budget and time constraints meant the team had to limit the type of task-based testing required by formal usability studies. They chose to conduct a couple of working group sessions to walk users through screen mock-ups and assess their ability to determine a clear path to what they wanted to accomplish on the page. “I think that we got valuable feedback from the sessions,” says Luzmaria Monasi, a Senior Research Officer who acted as the Business Project Manager.
- **Working groups**: Findings from the working groups helped the team make design decisions, such as how to adjust the labeling of buttons and headers to make them more explanatory. Users also suggested including explanatory text when hovering the mouse over specific buttons and headers, and provided feedback on other issues such as the positioning of buttons, how to show certain tabs or headers, and what not to show.

An analysis of some of the things needing improvement in the old UI.
Pictured: The first design sketch the team scribbled out on paper to try to improve the UI.

LESSONS LEARNED

The team members share a few lessons learned from the redesign effort:

- **Involve the users.** “It is very important to get user feedback while going through the design process. Work together with them. We had great partnership with our business sponsors and we involved actual users, economists and assistants throughout the process. We ran the design through them and asked specific questions such as how to name a particular button so that it is most intuitive,” says Faures.

- **Plan for delays.** It may seem counterintuitive to plan for delays, but even the most fine-tuned project plan can run into trouble at some point and it’s better to leave a little wriggle room in the schedule for the unexpected (but inevitable.) The project encountered several unexpected complexities, “The development of the UI was more complex than planned and introduced unexpected delays (particularly the integration with our existing document management system.) The QA testing and user acceptance testing took a lot more time than anticipated due to the complexity of the application,” says Monasi.
• **Building on top of SharePoint isn’t easy.** What the team learned about SharePoint was that it is complex to integrate a new UI into standard platform. “We spent significant time trying to figure out how to have the new UI integrate with our Document Management system. In the old UI, we were using an out-of-the box third-party tool to interface between SharePoint libraries and our document management system. Now with this graphical layer on top of SharePoint, we could no longer use that tool. We had to develop custom connections to the document management system. It was a lot more difficult to implement than we initially thought,” says Faures.

• **Understand the other team’s perspective.** “Being in the middle, between the technical people and the business side, I realized that what looked very straightforward and easy for the IT people, was not straightforward for the users. So that was crucial: understanding what users were seeing, and listening to them to try to understand what they wanted,” says Monasi.

**TEAM**

The IMF hired Eastbanc Technologies to design, develop and integrate the UI into the SharePoint environment. This project was a close collaboration between the two teams involving both Eastbanc and IMF project managers, designers, usability experts and developers.
The team roles are outlined below:

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>APPLICATION DESIGN TEAM</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Project Manager</strong></td>
<td>• Guide feature and functionality specifications</td>
</tr>
<tr>
<td></td>
<td>• Set project schedule</td>
</tr>
<tr>
<td></td>
<td>• Liaison with business stakeholders and users</td>
</tr>
<tr>
<td></td>
<td>• Facilitate user communication</td>
</tr>
<tr>
<td><strong>Business Analyst</strong></td>
<td>• Document functional and technical requirements</td>
</tr>
<tr>
<td></td>
<td>• Test design choices through user research</td>
</tr>
<tr>
<td></td>
<td>• QA testing</td>
</tr>
<tr>
<td><strong>Designer</strong></td>
<td>• Map out how users will interact with content and functionality</td>
</tr>
<tr>
<td></td>
<td>• Create wireframes and graphical design</td>
</tr>
<tr>
<td><strong>User Experience Specialist</strong></td>
<td>• Provide input on usability of design</td>
</tr>
<tr>
<td><strong>Developer</strong></td>
<td>• Implement designs</td>
</tr>
<tr>
<td></td>
<td>• Conduct testing and bug fixing</td>
</tr>
<tr>
<td><strong>QA Specialist</strong></td>
<td>• QA Testing</td>
</tr>
</tbody>
</table>
The IMF eReview project team included two staff from the Strategy, Policy, and Review (SPR) Department, which is heavily involved in the review of country papers produced by the IMF, and several staff from the Technology and Services (TGS) Department. A close partnership with the business users helped the technical team understand the needs of the eReview user community as well as providing business credibility to the effort.

Eastbanc Technologies was contracted by the IMF to design and develop the new eReview interface. Using JQuery over standard SharePoint 2007, Eastbanc built a user interface that has become the envy of other teams at the IMF.

The IMF’s Chief Information Officer, Jonathan Palmer, has long expressed the belief that technology should be a “delightful experience” for the end users. This philosophy was an underlying factor in the development of the new eReview interface. In addition, all parties involved strongly believed that an easy, intuitive, and pleasant interface could be the façade behind which complex business processes are carried out, similar to well-known online shopping applications, where shoppers can easily review the products they purchase. And this is indeed what the IMF has managed to do with eReview.

In the months since implementation of the new eReview interface, user feedback has been overwhelmingly positive. The team is often asked when eReview can be expanded to include other types of papers produced by the IMF, proving that eReview has indeed become a delightful experience.
Eventbrite “Create” Workflow

**Organization:** Eventbrite Inc.

**Application:** Online Ticketing Platform

**Headquarters:** San Francisco

**www.eventbrite.com**

**Design team:**
- Melissa Cooper, Senior Interaction Designer
- Julia Thompson, Vice President of Product
- Laura Coltrin, Product Manager
- Adit Vaidya, Group Product Manager
- Tom Censani, Product Design Manager
- Ryan Lottering, Visual Designer
- Dana Lamb, Senior Program Manager
- Galen Krumel, Engineering Manager
- Nathan Yergler, Senior Software Engineer
- Nesan Waran, Senior Software Engineer
- Michael Scheibe, Software Engineer
- John Terrenzio, Software Engineer
- Gordon Mei, Software Engineer
- Yinka Fatona, Senior Software QA Engineer
- Inga Gadimova, Software QA Engineer

**OVERVIEW**

**Main function:** Eventbrite is a self-service software tool used by event hosts to organize, sell and manage tickets for events. The *Create Event* workflow described in this report is the process used to set up a new event within the Eventbrite, including entering details about the event and determining how to structure and price tickets.

**Why it’s unique:** The new Eventbrite *Create Event* workflow was chosen as an example of a great design because it:

- Elegantly communicates a complex set of choices in simple terms, reducing the number of steps from eight to just three.
- Makes excellent use of progressive disclosure to make complex settings available, but not cluttering the main workflow.

**The organization:** Eventbrite is used by hundreds of thousands of event hosts in 150 different countries to organize concerts, professional conferences, intimate classes, private social events, nonprofit fundraisers, political rallies, competitive athletic events, and everything in between. There were 21 million tickets issued in 2011. Eventbrite offers organizers an all-in-one, technologically sophisticated solution with social media integration and seamless payment processing.

**Design philosophy:** Two major themes guided the redesign: the need to simplify the design while retaining powerful features, and the desire to help users feel confidence in the interface and in their own decisions. These principles informed many different aspects of the final design, from the sequence of steps to the importance of the *Preview* function.
WHO ARE THE USERS?

Eventbrite has two audiences: organizers and ticket buyers. The Create Event process, however, is only used by people setting up events. Within this group there are a few different types of people with distinct goals and behaviors.

Eventbrite started out as a do-it-yourself service for small events but has since grown into a more robust resource for larger organizations. The range of people using the Eventbrite Create Event flow now includes:

- **Professional event planners** view Eventbrite as their ticketing platform, are likely to rely on the sales team for help in leveraging features, but may not spend much time personally using the application
- **Support staff** for professional event planners who deal with day-to-day management of event information, and tend to blame themselves if they can’t find something
- **Self-taught enthusiasts** organize events related to their personal interests, may have little knowledge about Eventbrite’s features, and are likely to overlook or miss things and interact with the customer service team.

Both the larger professional clients and the “long tail” of individual users are extremely valuable to Eventbrite, so the interface must meet the needs of both groups. It must be simple enough for users to do tasks quickly without being overwhelmed by detail, and powerful enough to handle more complex ticket and event arrangements.

THE APPLICATION

The Create Event workflow is arguably the most important and difficult thing users do with Eventbrite, and it affects the experience of everyone who uses the application. The project to redesign this page, which began in 2011, had a few specific goals:

- Make it easier to set up and publish an event page
- Make it easier to discover advanced features
- Help organizers feel more secure and confident about using Eventbrite
- Increase the number of events published
- Decrease the time to publish (lag time between starting the creation process and actually publishing an event.)

The new design delivers on the above points with a simple and straightforward process that is even more impressive when you consider the full scope of what the page needs to accomplish. In addition to letting organizers input basic information such as the event name, time, and location, this workflow includes the ability to:

- Create multiple ticket types and prices
- Set start and end dates of ticket sales
- Determine the event capacity
- Schedule recurring events
- Customize visual appearance (background color, text color, images)
- Select promotion preferences such as search categories
In a single page, the Eventbrite Create workflow provides access to both basic and advanced options for setting up an event and selling tickets.
**TASK FLOW**

The new Eventbrite *Create Event* flow reduces the number of steps in the event creation process from eight to three while simultaneously exposing more settings and options to users. This apparent contradiction was elegantly accomplished through a combination of in-depth task analysis and layout optimization.

The previous *Create Event* flow had significant issues with feature discoverability, as Melissa Cooper, Eventbrite’s Senior Interaction Designer, explains, “People would talk to us because they loved our product and thought it was extremely intuitive, and tell us a story about a function that we didn’t have and how it affected them, except [I knew that] we actually had the function. So it was a case of the things that people didn’t know that they didn’t know that was failing in our software.”

The design team considered several possible structures for the new *Create Event* workflow:
- **Edit the preview**: Editable versions of each page the ticket buyer sees
- **Wizard**: Walk users through several steps
- **All details, customize**: A two-step process for entering information and customizing
- **Single page**: All actions presented on one screen.

Early testing found that the Edit the preview workflow did not match users’ mental model of the activity: They had come to the site thinking about creating an event details page, not registration and order pages. The Wizard alternative was also problematic, because separating the process out into several pages resulted in a drop-off and fewer people actually completing the workflow.

The process of outlining and testing these alternative workflows was valuable because it helped better define the problem space and constraints. Cooper explains, “We ended up with the *All details, customize* structure, but it was the single-page model that got us there. From testing the single page concept we learned that people really wanted to be able to review everything in one place.”

While users liked the single-page structure, they were daunted by the length of the page. “So then the challenge was to work out how to make the single page suit the design goals of simplicity,” says Cooper.

In addition to current Eventbrite customers, the Create Event flow serves an important function for people who are just considering the service, and want to find out what features and functionality it offers. The single-page design has a clear advantage for this type of user, since they can immediately see all the functions.
Early planning explored several different possible workflow structures, such as a linear wizard or an editable preview.
Users liked this early prototype of the single-page structure, but found the length of the page daunting.
**Improved Layout**

One of the most obvious improvements in the new design is the change from a two-column to a single-column layout. The previous version was intended to mimic the end product, the *Event Details* page, which displays the event location and time in a right-hand sidebar.

In general, using consistent layouts is helpful, but in this case there was not a close enough relationship between the *Details* page and the *Create Event* page to make the consistency meaningful. When people first arrive at the *Create Event* page, they may have never seen a *Details* page before, or may not have looked at one recently. Users aren’t likely to recognize the layout or have a strong association with that placement, so there isn’t much value to be gained by making them the same.

There is, however, a significant drawback to the two-column layout for a data entry page: It doesn’t present a clear path to guide users through the process. Observation of people actually using the page showed them jumping around haphazardly between different fields and elements. In part this was because the placement was optimized for people consuming or reading details about an event, rather than for people writing or engaged in inputting event details.

The new layout arranges the steps in a single column down the page, creating a straightforward progression for users to follow.

The new visual design also calls out each of the four steps with a bright number that jumps out from the page. This page element is a favorite with the Eventbrite customer support team because the clear numbers act as labels for the steps, which makes phone support conversations much easier.
Left: The previous *Create Event* workflow placed some elements on the sidebar, creating a scattered visual appearance. Right: The new design presents steps in a single vertical column, creating a clear visual path.
Eventbrite’s standard event details page displays the ticket information and event description on the left and the time, location, and host information in the right sidebar.
The previous *Create Event* page relegated core elements such as the event time to the right sidebar in order to use the same layout as the end product (the *Details* page). This design was not entirely successful.
Task analysis of the previous Create Event flow shows how the layout forced users to jump back and forth around the page repeatedly. Each colored line represents a different user who was observed using this page.
 Improved Task Sequence

In addition to changing the layout, the redesign significantly altered the order and grouping of input fields. For example, the new step 1 includes the input fields for title, details, time and location; which are items that used to be divided into steps 1, 3, 4, and 5.

Most users coming to this page already know exactly what they will put in the fields, so the form is easy to complete. Grouping fields in a single task doesn’t feel overwhelming to users, and in fact gives people a sense of accomplishment and progress when they make it through the first step so easily.

This grouping specifically omits step 2 from the old design: *Add Ticket Information*. This omission was a deliberate choice, because of the complexity of setting up ticket types and prices, says Cooper, “We were asking people to do the hardest thing on the page as the second of eight steps... But in testing we found that people just wanted to do the easy stuff first: name, date, time, location. And that resonates in the design that we have today... With step one you’re halfway through the page, and you’ve done it in just five minutes.”

Step 1 now combines the easiest input fields (which used to be divided into four separate steps) into a single task, letting users quickly make significant progress in setting up an event.
The previous sequence was also problematic because it encouraged users to overlook some important options, contributing to the feature discovery issues experienced by many users. The last three steps on the page were:

- **Step 7: Add Colors**
- **Step 8: Set Privacy**
- **Additional Options.**

The privacy options relate to whether the event will appear in Eventbrite search listings, and which keywords and categories should be used to describe the event. These settings are important for people interested in promoting an event, but the placement after the *Add Colors* step acted as a signal to some users that the rest of the page was not important. "Personalization is adored by some people, and others don’t care about it at all, they think it’s just colors and don’t find the important features underneath," says Cooper.

The new design has moved the *Additional Options* items to different locations where they are more relevant (e.g., *Show Number of Tickets Remaining* now appears in the ticket settings step.) *Set Privacy* has been renamed *Promote Your Event* to better communicate the purpose of these options, and promoted to be the third step of the *Create Event* process.

**STEP 7: ADD COLORS**

![Color options](image1)

**STEP 8: SET PRIVACY**

![Privacy options](image2)

**ADDITIONAL OPTIONS**

- Show number of tickets remaining on the registration page.
- Show attendees list on the registration page.
- Show custom header & footer.
- Track registration pages using Google Analytics.

The previous *Create Event* page listed the *Set Privacy* step after *Add Colors*, which made it seem less important to user.
The new Create Event workflow relies on progressive disclosure to meet the twin goals of simplifying the interface and making more features visible. This technique consists of displaying information and elements only at the point of need, which keeps the main interface uncluttered for the majority of users, while giving those who need it the flexibility of more advanced options.

For example, not all users need to see the full set of options for creating a recurring event, as these settings are only relevant for the minority of people who actually do need to set up a repeat event. Instead of including these input fields on the main interface, they are located in a lightbox that appears only when a user changes the value for the Event Repeats dropdown.

Since most events don’t repeat, the default setting of the Event Repeat dropdown is Never, and most users don’t see or interact with the full settings for repeating events.
Event Repeat settings are located in a lightbox, which appears only after users interact with the Event Repeats dropdown.

The procedure for adding an event’s location was significantly streamlined by shifting the bulk of the process from front-end user input fields to a back-end map search. Now, instead of five fields to specify the location, users see just one field which is actually a search box. As the organizer begins to type in the field, he is shown search suggestions. And after selecting a suggestion or entering the full address, a map of the location appears.

“Using the Google map was important for simplifying the page: It’s one field now, not five. Google is kind of doing the work for you there, its assisted search for a location. We still remember people’s locations, which is important. And if people want to put in information that wouldn’t traditionally come up in a Google map they can still do that,” explains Cooper.
STEP 5: ADD WHERE

The previous Add Where step had 5 fields.

The new interface has just one field for entering the location.
As you type a location you see search suggestions from Google Maps.

When you enter a location, a map is displayed to help verify that you’ve entered the correct address.

**INFORMATION DISPLAY**

Step Two in the *Create Event* workflow, setting up the ticket structure, is the most complex part of the process. There are all sorts of choices for organizers to make here including:
• What kinds of tickets to offer (e.g., general admission vs. VIP)
• How much to charge for tickets
• Whether to absorb Eventbrite’s service fee or pass it on to the customer
• When tickets sales will begin and end
• How many tickets are available, and whether this information should be displayed on the event listing page
• How many tickets can be purchased in a single transaction

Not only are there a lot of decisions to make, but the set of decisions can have a huge impact on the success of the event. And the answers are not necessarily easy to go back and correct later. For example if a user accidentally selects to pass on the fee but later decides to absorb it, he is faced with the prospect of have to figure out how to refund everyone who already bought a ticket.

The previous workflow attempted to simplify this process by using the progressive disclosure techniques that worked so well for other aspects of the design. The ticketing structure, however, is actually a case where it’s better to show users more information up front so they can understand their options and make an informed decision. The old design led to mistakes and missed opportunities, because users didn’t dig deep enough to understand their choices.

For example, organizers need to know the venue capacity when determining how many tickets they can sell. Cooper explains, “One thing that was really failing in the old design was, when you create the ticket quantity and save your settings, you couldn’t see them anymore. And the event capacity was off to the side and hard to discover. Now we’ve lined them up, and you can go in and create your tickets and make 100 general admission, and 100 VIP. That adds up to 200, but you can have a capacity of 150 and you just don’t care how many of each sell. You can lock that capacity and keep adding different ticket types but your capacity remains at 150. Things like that are really important in terms of managing capacity, and those are things that the long tail weren’t discovering. That’s where the interface really helps make your event more successful, because you’re no longer making those mistakes or overlooking those details.”

The need to help people set up complex business rules while maintaining a simple, self-explanatory interface creates a huge design challenge. As you can imagine, a great deal of design exploration, prototyping, and testing went into making the new ticket creation process.

As Cooper explains, the key goal for this step was to create confidence by providing straightforward information, “This ticket module is [designed] around transparency and instilling trust. It’s very much like what the event details page is going to look like once they get done with it, and it’s easy for that self-taught event planner to sit here and play with the numbers.”

A few specific elements contribute to making the entire ticket and pricing structure much easier to understand and control. These are the following:

• The end price that will be paid by the buyer is automatically calculated and dynamically updated
• Different service fee options are visible without expanding
• The number of each type of ticket and the total event capacity are visible
• Individual ticket settings are visible on hover
• Two rows of tickets are shown by default, to introduce the idea of creating more than one type of ticket.
The new *Create Tickets* module lets organizers see the total price and service fee options by default, without having to access advanced settings.

Clicking on the *Settings* link within the ticket row expands a window where users can customize individual settings (such as the sales start and end time) for just that ticket.
Organizers can review the individual settings for each ticket type by hovering over the Settings link.

The previous design for the ticket creation modal did not display any information about service fees or sales timeframe in the basic settings.
In the previous version, after creating a ticket you could not see the quantity of tickets available or the total price paid by the ticket purchaser.

**INTERFACE ELEMENTS**

Sometimes a picture is worth a thousand words. This often the case when customization apps, where a real-time preview of changes is much more effective than trying to describe them.

The Eventbrite Create Event workflow allows users to edit the appearance of different background and text elements, and makes this process much easier by providing a preview version that shows the actual page on the same screen as the customization controls, and updates dynamically as you make changes.

This full preview was difficult to build but really helped meet the goal of building the organizers' trust and confidence. Cooper explains, “You want to see all your work and feel confident that your settings are correct, and you understand what’s going to happen with your page when you publish it.”

The previous Create Event workflow did offer a preview function, but it launched in a new window and did not incorporate the actual customization controls into the preview screen. So organizers couldn’t actually see the changes they were making in real time, they had to switch back and forth between the controls and the result.

The preview function is accessible through a tab rather than being a step in the main workflow because, as Cooper explains, “The [preview] is of really high value to many of our customers, but it’s also something that some people don’t even want to look at. So they can easily skip it.”

For those who do care about the preview function, the design provides constant access by keeping the top tab control bar visible even if you scroll down the page. "As you scroll down the page, the bar at the top is persistent to the top of the browser. So we’ve kept the Save and Live buttons visible at all times, and you can toggle between [Event Details and Preview & Appearance] without having to save,” says Cooper.
The **Preview and Appearance** displays the event information as it will appear on the final event listing page, along with customization controls. Changes made in the top control panel are immediately reflected in the page below.
DESIGN PROCESS

Since Cooper had just joined Eventbrite as the user experience lead, she conducted extensive user research to familiarize herself with the product and customers. This research was particularly focused on the Create Event workflow (since it is such a critical piece of the Eventbrite experience,) but she also took time to gather general background information about event organizers and their goals and behaviors, “This was a huge project but it’s really core to our business, it’s absolutely crucial for people to complete this process. ...The product managers are always out there talking to customers, but they tend to have very specific needs when they go out. It had been a significant amount of time since we’d done some dedicated, deep-dive research with our customers.”

Her user research activities included:

- Observation of customer support calls
- Workshops with customer support representatives
- Exploratory interviews with customers
- Usability testing of the existing design
- Behavioral Analysis of web analytics data
- Usability testing of multiple prototype iterations.

An outline of the user research and design plan for the Create Event redesign project.

**Workshops with Customer Support**

In order to take advantage of existing internal knowledge about users’ needs and problems, Cooper listened to recorded support calls and conducted a series of workshops with Eventbrite’s Customer Support team members. “I threw key pages from the website up onto the wall and had them hurl problems at me for an hour per workshop. So I knew from that that there was really poor discoverability in our Create Event flow.”
During a series of workshops, customer support staff identified recurring issues with the current design.

**Exploratory Interviews**

In addition to talking with support staff, Cooper interviewed Eventbrite users directly to find out about their usage patterns. A range of users from both large and small organization participated.

These sessions were targeted more towards gathering general knowledge, “I’d start broadly with the general event planning process and see where they go, then shift into more focused questions about software. I learned how people view the product as a whole, what it means to them, how they position it in their mental model, and
what needs were and weren’t being met, before going anywhere near event creation... When we got to event creation...because I’d conducted the support interviews I knew there were things about the page that didn’t work. And when people said, ‘It’s easy,’ I knew there were things they hadn’t found, because of things they’d said earlier...I found a lot of cases where people didn’t know what they didn’t know.”

**Usability Testing**

Behavioral research in the form of usability testing was another important piece of the design process. The first phase included usability tests of the existing design, with users from different kinds of organizations who had different levels of experience with Eventbrite.

After synthesizing all the findings from the interviews, support team insights, and user testing of the old design, Cooper identified several different possible workflow structures, and developed visual representations of each. These were then tested with users by simply opening the image file in a browser. Cooper used Silverback to recording the sessions and GoToMeeting to broadcast them to others in the organization.

The next design phase continued to rely on user testing but evolved into a more focused, detailed prototype, “Once we chose our direction we moved into html prototyping, and focused on different parts of the page every week. As the weeks went on it became more and more interactive,” says Cooper.
Rather than a long formal report, the usability testing results were recording on a “live wall.”

Overall, the extensive user research performed for this project has helped build “a much better sense of what’s of high value to different verticals, which is really important,” says Cooper.

It’s easy and quick to throw a whole list of features and controls up on a page, but the end result is often a confusing jumble that leaves users feeling lost. Paradoxically, the designs that seem the simplest and most intuitive are often those that someone spent the most time thinking through and creating. This was definitely the case with the Eventbrite Create Event workflow, as Cooper explains, “Sometimes
I feel like I just facilitated this design. We did so much user testing, then it was like, ‘Oh! This is what you were meant to be.’ That’s great design: When it pops out at the end it feels like it should have been obvious.”

LESSONS LEARNED
The Eventbrite team shared some lessons learned from this design effort, including the following:

- **Leave time for planning and synthesis** of research. “At the end it felt huge... I did not leave enough time for the synthesis and reporting of my research. That part’s always going to take a lot longer than you expect. Also, there’s nothing more valuable than using a transcription service, and recruiting also takes so much time.”
- **Iterative design with users is invaluable.** “It only takes about two seconds... for you to actually be able to see something with fresh eyes.”
- **Be strategic with your research efforts.** “These techniques are highly valuable and really work, but they are also a significant investment. For us it was applicable to all of Eventbrite, not just this project. But moving forward we have to be strategic about where we really want to invest in that level of research-driven design, and where we want to trust in the experience of the team.”

TEAM
The design team included:

- Melissa Cooper, Senior Interaction Designer
- Julia Thompson, Vice President of Product
- Laura Coltrin, Product Manager
- Adit Vaidya, Group Product Manager
- Tom Censani, Product Design Manager
- Ryan Lottering, Visual Designer
- Dana Lamb, Senior Program Manager
- Galen Krumel, Engineering Manager
- Nathan Yergler, Senior Software Engineer
- Nesan Waran, Senior Software Engineer
- Michael Scheibe, Software Engineer
- John Terrenzio, Software Engineer
- Gordon Mei, Software Engineer
- Yinka Fatona, Senior Software QA Engineer
- Inga Gadimova, Software QA Engineer
Product Suggestion Portal (Hobsons)

Organization:
Hobsons, Inc., a (SaaS) provider of CRM technology and marketing solutions for education professionals

Application:
Product Suggestion portal

Headquarters: Cincinnati, Ohio (USA)
www.hobsons.com

Design team:
Members of the Hobsons, Inc. team did all design and development for the Product Suggestion portal in-house.

Members:
Abram Greene, Quality Assurance Engineer; Vanessa Keeton, UI Designer; Jaswinder Johal, Software Engineer; Sidharth Malhotra, Business Analyst and Vishal Saboo, Product Manager.

OVERVIEW

Main function: As a customer-centric organization, one of Hobsons’ guiding principles is to seek and incorporate market feedback into its products. The Product Suggestion portal provides end users of the company’s CRM suite a simple tool so they can communicate directly with the company’s Product Management team.

Hobsons’ CRM product is a mission-critical tool for the educational institutions Hobsons serves. With the product suggestion portal the company gives its customers an opportunity to make suggestions on the spot, while using the CRM suite as part of their day-to-day workflow. So if they’re in a specific section in the CRM and they feel like a certain feature or a certain function can be enhanced, or if something’s completely missing, or if they’re not able to figure out something, they can instantly make a suggestion, vote and comment on others’ suggestions and track the status of those proposed improvements as they make their way through the pipeline. The tool ensures that Hobsons customers have a voice in a product suite that affects their day-to-day job function.

This tool was a major leap forward for the organization, replacing a previous system that created too much inbox clutter and prohibited any meaningful analysis of the data it collected.

The organization: Hobsons is a ‘Software as a Service’ (SaaS) provider of CRM technology and marketing solutions that empower education professionals to manage the entire student lifecycle, including recruitment, application, enrollment and retention. The CRM suite (which this application is a part of) provides clients with the ability to communicate with prospective and current students using multiple communication channels: email, text, web portals, social media, telephone, letters, etc. The application suite provides clients with the ability to create customized college applications, manage application data, communicate admissions decisions, and provide a customized experience to each student that utilizes the system.

Design philosophy: “We try to balance the product management principles, innovation and usefulness in thinking about new products and features. The Product Suggestions portal is a good example of executing these principles primarily because of the speed at which we were able to bring a simple, elegant, and integrated portal to our customers. It continues to be one of primary tools for communication of new features and ideas between end users and Product Managers,” explains Vishal Saboo, Product Manager.
**WHO ARE THE USERS?**

Users of the product suggestion tool are the same users who rely on the CRM application suite in general. These users fall into two main groups:

- **Admissions staff at higher education institutions:** Nearly 2,000 end users at 500 client institutions utilize the portal each month and submit new suggestions to improve the functionality of the CRM and Application suite of products. They vote and comment on existing suggestions and follow the status of suggestions.

- **Product Managers:** Ten users in the Product Manager role are responsible for reviewing, moderating, and publishing incoming suggestions, merging duplicate suggestions, commenting on suggestions, and editing suggestion statuses.

**THE APPLICATION**

Every page of the CRM suite includes a footer link to the Product Suggestions portal. When new customers start using the CRM and go through the professional implementation and training, they see that there is a channel to communicate with the product management team while they’re using the product. Existing customers have become accustomed to this link. Before the Product Suggestions portal was created, users could click a footer link at the bottom of any page to send feedback, but the link simply generated an individual ticket and sent it to Hobsons’ Product Managers. There was no system to provide visibility to users about what happened to the suggestions after they were submitted, and dealing with and analyzing the suggestions was a near impossible task. The Product Suggestions portal is a savior in helping to manage and address product suggestions.

The top two tabs in the app, **View Suggestions** and **Submit Suggestion**, make the most important tasks readily visible and achievable, and the names on the tabs clearly denote the more passive activity, **View**, and the more active one, **Submit**. In the **View Suggestions** tab users may sort by the sub-tabs: **All Suggestions**, **Suggested by Me**, and **Suggested by Co-Workers**. This enables people to easily review and compare suggestions depending on the source. Still, dealing with these categorized lists of suggestions could be taxing, so designers included filtering options in the **Refine Results** section in the left rail. In the top box, a user may refine results in a few different ways: he can type in a keyword to search, or select a checkbox to filter by status (**Suggested, On Roadmap, In Development,** or **Launched**.) Finally, he could also sort the list by other criteria such as **Recently Updated**. Building a variety of sorting and filtering mechanisms into a UI is an excellent way to accommodate multiple usage scenarios.

The **Suggestions by Product** section (above the refining section) offers a list of popular topics, accompanied by the number of items in each topic. Each topic is an underlined link for easy affordance and access. UI’s such as this take advantage of the power of suggestion: a flat list with all items visible, as long as the list is short and understandable, exposes the user to possibilities she may not have thought of. Making smart suggestions to the user is not only helpful, but breeds engagement and loyalty.

Since a finite number of items will fit in the allowable screen real estate, designers enable users to choose the number of items that appear on the page. This, as well as the pagination option, appears in the grey menu just above the list of suggestions.
In a relatively small space designers are able to describe a lot about each suggestion, namely: suggestion title, who created the item, date it was created, status of the suggestion, a longer description of the suggestion, a category for the type of product being suggested, a short description, a link for comments, a link to choose to subscribe, and finally thumb up and thumb down voting buttons along with the current vote score. The horizontal section for this area is full but not crammed. Designers use a combination of larger and bolder fonts to direct the eye to the different content.

On the View Suggestions, All Suggestions main page, users see the most recent suggestions activity at a glance. From here, the user may vote and comment on a suggestion, search for suggestions, or submit a new suggestion.
Sort options on the left include **Score (High to Low,)** **Score (Low to High,)** **Newest,** **Recently Updated,** and **Oldest.**

When a user makes any kind of input—such as an order, comment, or suggestion—he wants some feedback and the ability to seek out status when he is ready to do so. The **Suggested By Me** tab makes it easier on the user to track the progress of his own suggestions. This view resembles the **All** tab, but the voting icon colors are less vibrant, to indicate that the vote has already happened. Additionally, a note at the top of this page states that even items which are assigned **Pending** status, and not posted for voting and comments by others yet, are visible here. But it further explains Hobsons’ procedure for managing suggestions: to review and then post them for votes and comments once a month.
The View Suggestions > Suggested by Me section provides a listing of suggestions submitted by the user and includes the status of the suggestion.

Allowing people to vote on suggestions serves a few purposes, not the least of which are: 1) more customers feel heard, and 2) Hobsons can get a better sense of how important each suggestion may be to the customer base. The voting process itself is expected and easy, with just two icons placed near the suggestion title. The standard thumbs up with a green background, and thumbs down with a red background are understandable. While the icons are small, they are distinguishable based on both color and the thumb graphic. This is positive for people who are colorblind because the graphic suffices. For people who are not colorblind but who may still have lower vision, the different background color may make it easier for them to pick out the right icon.

Once the user votes, he is presented with a message, for instance, Your Thumbs Up Vote has been successfully saved. This type of feedback gives users confidence and enables them to feel in control.
Voting provides a confirmation message in a dialog to ensure that the user is aware that their action was recorded.

In addition to voting on suggestions, users may also comment on them. Again, this feature can help the customer base feel heard; it also allows people to offer a different perspective, and can help Hobsons get a better understanding of users’ desires and views. Users may also read other customers’ comments for more perspective. The Comments link displays the number of comments already made, disclosing how much activity there is surrounding a particular suggestion. These niceties give more information with little effort on the design and development side.

Clicking the Comments link expands an area to expose all comments which have already been made, leading with the most recent first. Users may, of course, scroll to read the less recent comments. Above the list of comments appears the Leave a Comment field. The label accompanied by the open box and Post Comment button are inviting elements and summon people to type their thoughts. An effective status label below the field counts down the number of characters the person has left to play with, a la Twitter’s UI.
Users may read and add their own comments, elaborating on a suggestion or providing a different perspective or use scenario.

If a user is particularly interested in a suggestion he may choose to subscribe to it by clicking the *Subscribe* link. The message the user is presented with communicates that the user has subscribed, the name of the suggestion he has subscribed to, and choices to be notified when status changes or comments are posted. All of this is pertinent, helpful information for a confirmation message.
After subscribing to a suggestion users are presented with a confirmation message with the suggestion name as well as subscription options. Once the user confirms, he will begin receiving email notifications when the suggestion is updated.

In addition to viewing, commenting on, and subscribing to suggestions, users may of course make new suggestions. They do this under the Submit Suggestion tab. Here they are presented with a basic form with fields for Suggestion Title, Product, and Description. The Product field has a drop-down list, so users don’t have to type in the product name.

Once a suggestion is submitted, users are again presented with an important feedback message, which explains what will happen to their suggestion next: Thank you for your suggestion. All product suggestions are reviewed by our Products Team on a monthly basis. We publish suggestions based on similarity to the development work already in progress, the level of effort, and market relevance. Once a suggestion is published, all clients have the opportunity to vote for and comment on the suggestion...
Submitting a suggestion provides a detailed confirmation message explaining the process and setting the user’s expectations.

Once the suggestions are submitted, the product managers must review and publish them before they are available to the rest of the user community for votes and comments. The Manage tab in the Product Suggestions Portal is designed for these back-end users who need to analyzing and publish suggestions.

To review suggestions product managers may search and filter by various criteria, such as: keywords, name of the suggestion, status (such as inactive, archived, pending, suggested, on roadmap, in development, launched, and merged,) and date or range of dates the suggestion was created on.

The user may also click the Download Suggestions button to copy the suggestions to a spreadsheet file where they can manipulate the information. Allowing people to download or export the data and use a different, powerful program such as Excel is usually far better than trying to build some of the basic spreadsheet features directly into the app. Users may also view the Manage> Reports area for a summary of suggestions.
The Manage Suggestions tab provides Product Managers with a simple way to search by different criteria for suggestions. Suggestions can be modified in bulk and downloaded into a spreadsheet for additional analysis.
The two suggestions reports provide a snapshot of the Product Suggestions for all products in the various statuses; the top report summarizes all portal activity, and the bottom report shows only activity from the last month.

As product managers address the suggestions they are able to change each suggestion’s status to reflect its current stage in the workflow. They may do this directly on the page of suggestions by selecting the item’s related checkbox on the left, then choosing from the *Mark Suggestion as* drop-down list of choices.
Product managers may select suggestions in the Manage tab and then assign a new status.

Once the status is updated, the product manager sees a confirmation, and is asked if he would like to send an email to subscribers notifying them of the status update.
Product Managers see a confirmation when they update a suggestion’s status, and can choose to email subscribed users about the new status.

The email notice sent to subscribers presents some automatic information about the updates, such as the name of the suggestion and the new status category assigned to it. It also has a link to the suggestion for review.
Product Managers can easily send an automatic message to subscribers, informing them of a status change.

Additional maintenance is required to keep the suggestions in order, namely eliminating duplicate suggestions. The team’s research shows that 20% of suggestions each month are duplicates, and users do not search for suggestions to see if their idea has already been entered before creating a new suggestion. In the future the team plans to offer a smart, dynamic area on the page where people make suggestions. As the user types, the app will query the current body of suggestions for similar or same suggestions and present them to the user to choose from and comment on.

In the meantime, until that feature is created, product managers may merge a duplicate suggestion and notify the user who made the suggestion. They find like suggestion entries and then use a merge feature to consolidate them. After opening a suggestion they would like to merge, the product manager can choose the Merge To button in the upper left of the page.
Users select the *Merge To* button in the upper left to merge the currently selected suggestion with another suggestion.

At that point the suggestion is ready to be merged, and the next page says as much, repeating the suggestion again, and asking for the ID for the suggestion it should be merged with. Once the ID is entered and the page submitted, the system displays a page with both suggestions visible. This preview helps users understand the process and feel confident about what they are about to do.
The Merge Suggestion page displays both suggestions to help product managers understand exactly what is about to happen, before the system actually merges the documents.

Once the suggestions are merged, the merged suggestion shows up at, basically, the first comment in the comments area. The merged suggestion appears in front of a yellow background, so it is differentiated from the comments, and the heading is titled Merged Suggestion.
Merged suggestions become special “comments” to the main suggestion. Also, any user votes for a merged suggestion are automatically added to the count for the parent (merged with) suggestion.

Since Product Managers are in contact with many different types of customers and people at the organization who interface with customers, they have their finger on the pulse of what customers want. Thus, the Suggestion Portal app makes it possible for Product Managers to vote about a suggestion on behalf of someone. This way the importance of the suggestions they hear are not lost, and instead are voted upon in the system as though the third party had access, interest, and time to do it.

To use in this feature, the Product Manager clicks the *Product Manager Update* button in the *Edit Suggestion* view. After clicking it they are presented with a dialog box and a drop-down list of sources to choose from, including: professional services, current client, prospective client, sales, pre-sales, product engineering, product development/ QA, or other.
Product Managers can vote on behalf of an internal or external customer using the Product Manager Update feature when editing a suggestion.

Product Managers are asked to choose a source for the vote.
Product Managers may vote on behalf of the following sources: Professional Services, Current Client, Prospective Client, Sales, Pre-Sales, Product Engineering, Product Development/QA, or Other.

The Suggestion Portal at Hobsons supports a core business edict as well as workflow. It’s imperative that the organization collects feedback about their apps. On the flipside, customers expect to be communicated with and responded to when they make a suggestion. Ordinarily this makes for a lot of work for an organization, but this suggestions portal gives users much more visibility into the process, with far less effort on the part of product managers. The app takes much of the burden off the users and deals with it efficiently and transparently. The creation, review, commenting, and sorting of suggestions is an inclusive, easy process that is seamless to work with.

**DESIGN PROCESS**

**Usability Methods**

The design team knew they had to address the inefficiencies of the old system, but it wasn’t clear at the start if they could or should build their own tool, or piggyback off an existing platform. So, the first step in addressing their challenge was to exhaustively research the landscape of ideas and tools while conducting a thorough cost analysis.

**Learning from similar tools.** One of the first research efforts the team embarked upon was to look at other, similar tools, across their own organization for guidance and best practices. Though these tools are admittedly different products with a different user base, the concept was similar: give users the ability to submit suggestions. The design team felt they could learn from their colleagues’ hits and misses by getting first-hand knowledge from those teams about what worked and what didn’t during their development efforts. The design team also analyzed public-
facing customer suggestion tools, such as Starbucks’ Ideas Portal and GetSatisfaction.

The research results made it clear they would have to build their own tool, from scratch. So, the first step was to recruit a development team to build a prototype as a proof of concept. This effort cemented their mission: to build a custom tool.

**Site visits.** Hobsons Product Managers occasionally visit customers, so during the design phase of the project, they used some of the time spent during those visits to gather data about how customers use the Application Suite. This data influenced how they designed the suggestion tool. Observing users in the field was a big opportunity for the design team. During a routine customer visit, if there was a little extra time at the end, the team would make the most of it. They shadowed customers as they used the application, probed when necessary to understand a workflow, and basically just observed how customers were using the product. These observations, along with feedback in response to ideas generated by team members informed the feature set and design of the tool.

**Using customer calls to inform design.** The business analysts who create the wireframes do not attend site visits, but when work begins on a new feature they are there, right from the start. During the initial design phase for this project they were also able to listen in on customer calls. Customer calls are an opportunity to give customers a glimpse of what to expect in the product over the next six to nine months and for the team to hear the feedback. There may be up to 40 customer calls per year so the design process is closely in touch with the opinions of the end users. The feedback gathered during these sessions gets rolled into the wireframes the analysts create.

All of these activities—competitive research, site visits, and customer calls—informed the development of early wireframes and prototypes. The team relied on user feedback to guide some key changes as the design evolved, as Product Manager Vishal Saboo explains:

- **Combined search box:** “The search box was originally placed in the top right because that’s the most common place for a Search box. However, as we iterated through the design, we realized that searching, filtering and product breakdowns made sense to all be together. That’s what we did in the final design.”

- **Additional statuses added:** “We started out with just four status options: In Review, On Roadmap, In Development, and Launched. But as we iterated through the design, we realized these were not sufficient. We added five additional status options: Pending, Suggested, Archived, Merged, and Inactive. The Pending status was added to help incorporate a significant process change suggested by one of our senior executives: that Product Managers should review all suggestions before making them visible to other customers for voting and commenting.”
Originally, search was placed top right and there were only 4 status options.

**Designating a stakeholder team.** Another way to keep the end users close to the design process is by creating a proxy group that represents the needs of the end users. The design team at Hobsons handpicked a group of internal and external stakeholders, drawn from design partner organizations and clients, to participate in a steering committee. The design team returns to this group every six months or so and presents whatever’s on deck on the roadmap in terms of product development and features. This group, with its interest and expertise, can act as a sounding board for potential initiatives — both providing feedback on new ideas and providing ideas themselves. The Hobsons team did this early on in the design process for the suggestions tool.
<table>
<thead>
<tr>
<th><strong>PROJECT TIMELINE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Late 2007</strong></td>
</tr>
<tr>
<td>Launched a “Product Suggestions” link in the CRM tool, but beyond simply collecting the feedback, there was no follow-on process and no good way to parse the responses. It was untenable as a feedback collection mechanism. The team was bombarded with emails, clients complained about getting no response to their feedback.</td>
</tr>
<tr>
<td><strong>January 2010</strong></td>
</tr>
<tr>
<td>Commissioned a project to investigate better options. Some existing ideas portals were mined for ideas and best practice examples, including: Salesforce.com, Starbucks Idea, GetSatisfaction, etc. Buying and integrating an existing tool was deemed too expensive so the decision was made to build one in-house</td>
</tr>
<tr>
<td><strong>February 2010</strong></td>
</tr>
<tr>
<td>Development team built a prototype proof of concept (took one month to build)</td>
</tr>
<tr>
<td><strong>April 2010</strong></td>
</tr>
<tr>
<td>Official project launch: the company gathered a team of business analysts, UI designers, developers and started wireframes and design mock-ups</td>
</tr>
<tr>
<td><strong>August 2010</strong></td>
</tr>
<tr>
<td>Application launched</td>
</tr>
<tr>
<td><strong>Post-Launch</strong></td>
</tr>
<tr>
<td>The application has gone through four additional releases since its August 2010 launch. These releases have added incremental functionality including:</td>
</tr>
<tr>
<td>- Ability to download suggestions (for product managers to parse the suggestions)</td>
</tr>
<tr>
<td>- Reports</td>
</tr>
<tr>
<td>- Added a merge feature to reduce the amount of duplicate submissions. This sends an automatic notification to client that their suggestion has already been submitted.</td>
</tr>
<tr>
<td>- Notes: product managers can make notes on suggestions</td>
</tr>
<tr>
<td>- &quot;Proxy vote&quot;: feature that allows a product manager to vote on behalf of someone else</td>
</tr>
<tr>
<td>- Subscribe and Notifications features added</td>
</tr>
<tr>
<td><strong>Future Phase</strong></td>
</tr>
<tr>
<td>The team is considering integrating the suggestion application with internal roadmap tool (an Excel spreadsheet)</td>
</tr>
</tbody>
</table>
LESSONS LEARNED

The Hobsons design team shares a few lessons learned from designing the suggestions tool:

- **Start small.** Sometimes even a good idea can be too big a project to tackle. Start with a small, achievable piece and work from there.

- **Choose the right people.** When choosing members of a stakeholder group, choose wisely. Choose people who are close to the customer, who do training, who visit clients regularly, or who are early adopters.

- **Stay in touch.** Once you have a solid steering committee in place, make good use of their services. Reach out to them every six months, at least, to keep their interest and participation alive.

TEAM

“Our customers, who are the users of the Product Suggestions portal, are the ones that seeded the idea for the Product Suggestions portal and deserve ultimate credit. Once the need was evident, a small team got together to do a rough prototype, somewhat of an anomaly from our typical Product Management process. The team was influenced by Marty Cagan's book *Inspired: How To Create Products Customers Love,*” explains Saboo. The design team for this project is located within the product management department; their roles and responsibilities break down as follows:

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Managers</strong></td>
<td>• Ownership of features and functionality; Determining when features are released and what is going into each release</td>
</tr>
<tr>
<td><strong>Business Analysts</strong></td>
<td>• Brainstorming with Product Managers and end-users to create wireframes and functional specs</td>
</tr>
<tr>
<td><strong>User Interaction Designers</strong></td>
<td>• Realize the vision of the wireframes in HTML and mockups with a special attention to usability</td>
</tr>
<tr>
<td><strong>QA Engineers</strong></td>
<td>• Writing Test Cases, integration/performance testing, reporting defects, regression</td>
</tr>
<tr>
<td><strong>Back-end developers</strong></td>
<td>• Implementing designs, unit-testing, and bug fixing</td>
</tr>
</tbody>
</table>

**Vishal Saboo** is a Product Manager at Hobsons. He is passionate about design and loves simplicity. He is responsible for all the functionality in the product and is the driver behind new features and improvements. He graduated with a Master’s in Business Administration from The Ohio State University, and a Bachelor’s degree in Instrumentation Engineering from Mumbai University. His interests include tennis, photography and music.
Sidharth Malhotra is a Senior Business Analyst at Hobsons. He is a creative problem solver. He is responsible for requirements gathering and building solutions for the product roadmap. He has over 10 years of IT experience and has worked in different roles in the software development lifecycle. He graduated with a Master’s in Business Administration from Chenni University in Mumbai, and a Bachelor’s degree in Business Administration from Mumbai University. His interests include squash, cooking, travel and music.

Abram Greene has been with Hobsons for over three years. He is a graduate of Spencerian College, with a degree on Computer Engineering Technology. As a key member of the Quality Assurance Team, he is responsible for making sure our products are effective, functional, consistent, intuitive, efficient, usable and reliable. Outside work, Abram is passionate about his UK basketball team, and loves to play golf.

Vanessa Keeton is an Interaction Designer at Hobsons. Vanessa is very passionate about clean and reusable solutions for design and coding. Before she became a professional designer, Vanessa played and taught saxophone professionally. She has a Bachelor’s in Jazz Studies from Morehead State University and an Associate’s in Interactive Media Design from the Art Institute of Ohio — Cincinnati.

Jaswinder Singh Johal is a Software Engineer at Hobsons. He is a developer with wide array of skills and a broad knowledge base. He is responsible for writing and maintaining the code behind the product. He graduated with a Master’s of Science in Information Technology from Punjab Technical University, and a Bachelor’s degree in Computers Application from Punjab University. His interests include music and learning about new technologies.

Pictured: (left to right) Vishal Saboo, Sidharth Malhotra, Abram Greene, Vanessa Keeton, and Jaswinder Johal.
Multi-Platform Applications

Applications’ in the Multiplatform area have transmedia strategies and goals which are well defined such that the applications actually remain true to their brand across multiple environments. Designers in the multiplatform area ensure that the core functions are supported appropriately across diverse platforms while taking full advantage of each medium’s finest elements, and avoiding the feeble areas. For example, designs in this group exploit the power and screen real estate a desktop environment affords, adding more features, such as analysis and manipulation, commenting, and sharing. These app designers focus on the importance of mobility and always-available environments, but they know the limitations of small screen displays and restricted input devices, and design within those confines. Designers also know the nuances between similar media, such as the fact that designing for a smartphone display is only similar and is not the same as designing for a tablet.

The winning design teams in this category were able to keep straight not just one operating system or hardware platform but several. They handled multiple strategies, schedules, and the tasks of sharing design sometimes across teams, all while keeping true to the wonder and limitations of the application’s central goals and to each medium.

Winning applications in this category include:

- Dixio, multiplatform lookup technology
- Highlight, lets users with iOS devices make audio recordings, mark interesting points, and share recordings with others.
- Mobile Unified Communication (UC) client, allows business customers to link two phone numbers, their mobile and business lines, with their mobile phone.
Dixio Multiplatform Lookup Technology

Organization: Semantix Group SL  
Application: Dixio multiplatform lookup technology  
Headquarters: Barcelona, Spain  
www.semantix.com  

Design team:  
In-house: Ferran Gatius, Product Manager; Jesús Liarte, Developer; Francisco Javier Hidalgo, Developer; Laura Vilar, UI Designer and Joaquim Triginer, Quality Testing.  
External Agencies: Blit Software and Vinco Mobile

OVERVIEW

Main function: Dixio is a multiplatform lookup technology that, together with dictionaries and encyclopedias, provides readers with the most relevant reference information possible without leaving the document that he is reading. Dixio works on several platforms and in many applications, by accessing the Dixio API that connects to the Semantix Cloud Knowledge Engine (CKE) or the Local Knowledge Engine (LKE) to provide the best possible result.

- Dixio Desktop can be used on just about any desktop application on a user’s computer system: documents, emails, PDFs, spreadsheets, etc. The desktop tool is used by approximately 82,164 users.

- Dixio eReader Solution can be used on a variety of devices from smartphones to touch screen tablet devices and is used by approximately 10,000 users.

Why it’s unique: The knockout feature of this technology comes from the semantic engine that drives the lookup results. This “capture & search” function reads the entire context of a given word, analyzes the content and returns a context-sensitive definition in a just a few seconds. This ensures that Dixio can provide the best, most appropriate definition possible, each and every time. It is non-intrusive software that remains silent until the user calls it and it works on the fly, rather than requiring the user to launch it as a separate task. Dixio is able to identify the language of the text in question, in order to provide definitions and/or translations.

The organization: Semantix, the company behind Dixio, develops tools to help people communicate better. Its tools bring together people who speak different languages, making their communication easier, either in their own language or across different languages. Semantix expertise is drawn from its diverse staff including: linguists, computer engineers, philologists, translators, lexicographers, communication specialists and marketers, all working to improve the power of communication amongst people of any language.

Design philosophy: While most dictionary apps rely on the age-old paradigm of a paper dictionary where you “go” to look up a word, the Dixio technology is trying to turn that idea on its head. Dixio gives the dictionary a whole new look and feel, where the dictionary becomes more of a gadget than a book. It’s there when you need it but it won’t get in your way when you don’t.

USERS

Dixio users are drawn from a broad range of people who use an array of electronic devices (such as e-Readers, tablet, and smartphones.) The users are an eclectic
group of individuals from educational centers, e-learning platforms and publishing companies.

**THE APPLICATIONS**

On all platforms, Dixio acts like a very good restaurant waiter: there in full force when you need him, but not interrupting your dining and conversation when you don’t. Dixio is an ambient power, falling into the background in any one of these three devices: the desktop, phone, or tablet. Hiding the app until needed enables people to read, work, and engage with content without having to tune out the noise of an overzealous dictionary app. Designers wholeheartedly took this approach, as they believe, according to Laura Vilar, “[Users] are interested in reading the book, not working with an application.” Astute designers are comfortable with this concept, and have little ego about needing their app to be center stage.

Once the user is stumped on a word he just read and summons its definition, Dixio snaps to attention. Calling the app is simple on all platforms: on the web, double click; on desktop, control + right click; and on smartphones and tablets, long tap and double tap. The features are advertised and described to the customer when he begins using the service, and there is a small banner in the right rail of webpages also announcing the service and features.

Upon completion of the small summon gesture, which works when directed at any word on a page, a result dialog box appears containing a contextual definition gleaned from more than 95 dictionaries, including Oxford, Vox, and Semantix Bilingual.

**Multiple Media**

In each environment, the Dixio UI is simple, and consistent with other platforms to a degree, but also exploits each platform’s offerings. According to Vilar, “As long as each application is used for a different platform, we had to adapt it to each device. Obviously if you’re reading in a tablet it has to be something light that appears and disappears easily.”

And Ferran Gatius, Product Manager, agrees, stating that designers needed to make some tradeoffs and change navigation concepts which are understandable on the desktop but not on other devices. For mobile, he notes a known design guideline, “It doesn’t make sense to have so many features on the tablet, like settings; we need to make the navigation simple.”

Knowing each platform’s strengths and weaknesses and designing for those is part of what makes a great transmedia strategy, as does defining which elements should remain consistent no matter what the medium. In the case of Dixio the following three items are those the team determined should be implemented consistently across all platforms:

- **Content:** The word definitions are the same.
- **Access:** The “crown” area which appears on the top of the tabs, as well as the tabs themselves.
- **Visibility:** The Dixio window will never appear over the word a user has selected to define.

The semantic engine which is the power behind these apps’ success is not exactly the same for all platforms, though this is on the product roadmap. The current plan
is to move towards a server-based lookup function, as cloud computing becomes more popular and stable. In the meantime, some features which are available online are not available in the local versions, for example, only on the EReader Solution can you click a tab to get the online definitions.

**Dixio Desktop**

The Dixio Desktop is more than your average dictionary, with easy design and powerful features. In the search results, users may select their preferred dictionary via the top horizontal tabs, which display not only the source name but also the number of items on the respective results pages. With this, users get a sense of the results before they click or read anything. They may keep the smart default selection tab, *All*. In many apps like this where data is coming from multiple sources and the user may actually care to know the source, the results are sorted in tabs by source. But since it is quite possible the users won’t see, or may just not be inclined to click the tabs, then having the *All* tab—and having it as the default—is essential. While there is more information to scroll through in the *All* tab, there is a higher chance that people will find the information there than on the other tabs. The drawback is that, it’s more work for users to look through the information categorized by source on one long page. But, that’s what the companion UI tabs are for, and the two interfaces provide an effective safety net for one another, and cover a wider range of user scenarios.

The app also helps people to recall and learn definitions, as users may view the previous word they looked up by clicking the arrow to the left of the search field.
When using Dixio Desktop, users can look up a word’s definition by pressing control + click (or a customized key + click combination of their choosing.)

Dixio Desktop not only defines words, it also translates them. The semantic analysis gives the most appropriate meaning according to the context. And because the engine reads the whole article and has context for words, the translation can easily help people find out what a random word in another language is. So if the user is reading an article in English but there is a Spanish word in the text which he chooses to translate, Dixio presents the definition of that word in English, the language most of the article is written in, rather than in Spanish. This helpful feature makes content more seamless, engaging, and understandable. Users may access translation features via the Define and Translate button in the upper right.
A submenu by the search in Dixio Desktop allows people to define and translate words. In this example content comes from Wikipedia, and can be visual and thematic, for example regarding: medicine, law, sport, computing, etc.

For even more knowledge about the lookup in question, users may obtain their results as images instead of text. Images are retrieved from Google, Flickr, or the Semantix visuals gallery by selecting from the search sub menu in the upper right.
On Dixio Desktop you may switch from text results to images.

The desktop application not only works on websites, but also with MS Word, PDF, email, and spreadsheets, offering the same commands and features.
Floods have washed hundreds of homes away.

Dixio Desktop work on MS Word and PDF documents, as well as websites. To make Dixio work best for each individual, the app allows people to see and make changes to their account, change the language, and change preferences about internet connectivity, as well as the dictionaries. Users may also add their own customized glossaries. (An administration panel allows IT staff to keep this in check.) These types of customization features increase ease of use and build loyalty.
Users may edit preferences such as which dictionaries to search and when to connect to online dictionaries over the web.

The Contents tab displays the Dixio Library of dictionaries that users may include in their definitions. Users can choose which dictionaries to enable or disable, and rearrange dictionaries so that definitions from their preferred sources appear first.
The Dixio Library, under the main menu, allows users to change their dictionary preferences.

**Web**

Dixio Technology may also be installed on a website by the site’s owner so that everybody who visits the site may use it, even if they don’t have Dixio Desktop installed locally.
When installed on a website, users may look up any words on the page using Dixio for Web by double clicking on any word.

**Dixio eReader Solution**

To start the eReader Solution application, users double tap or tap and hold the device screen on the word they want to define. A simple dialog box appears, which displays the definition and two tabs: the default *Dixio* tab, which uses only local content, and the *iDixio* tab, which accesses online definitions on the cloud if an internet connection is available. Local search is the default, which is important because device storage is always an issue, and this helps keep the app quick and responsive. But at any time, the user can switch to a wider range of definition content through second tab.

The dialog also displays icons in the upper right to access secondary functions. For example, the books convey which dictionaries the user is working with, the magnifying glass opens a new dialog to allow manual search, and the X dismisses the dialog box.
When a user looks up a word, the definition appears, along with tabs that allow users to switch between local and online definitions. The dialog also displays controls for secondary functions: the books convey which dictionaries the user is working with, and the magnifying glass opens a search box.

When users tap the magnifying glass icon, a new dialog opens to perform a manual search.
The definition dialog box also displays the term a user searched for, the dictionary searched, and controls at the bottom to view definitions of similar words and phrases. Taking advantage of the swiping gesture available on tablets and smartphones, users may swipe in a definition to display another definition from a different source for the particular word.

Users may swipe within a definition to display more definitions for the particular word from different dictionaries.

In addition to working with tablets and smartphones, eReader also works well with electronic readers.
Dixio eReader also works seamlessly with electronic book readers.

Dixio vanishes as quickly as it appeared, when users click outside the dialog box, or on the X in the upper right. But with an app this good and helpful, we are sure users will call it back often.

DESIGN PROCESS

The Dixio technology originated as a desktop application called Gotitclear, which was basically a dictionary app. In order to speed up the lookup process, the company decided to move the technology to the Cloud.

The team has employed a number of research methods to inform their journey from Gotitclear to Dixio and to improve Dixio over time. These include:

- **Competitive analysis:** The team started their research by analyzing similar products available in the market and studying design guides, prior to developing their own. This snapshot provided a good starting point for the Dixio project. This competitive research included a thorough look at the tablet space writ large, to see where tablets were succeeding and where they were failing to stack up to eReader Solutions. What they found was a clear demand based on how the tablet apps performed as eReader Solutions.
• **Reading blogs:** An important font of knowledge for the team was to read blogs where users were expressing their opinions (and complaints) about existing dictionary apps. In this forum the team was able to learn a lot about the user experience flaws people were encountering in their reading.

• **Studying user patterns:** One of the key metrics the company used to guide their re-design from desktop to Web was to take a deep dive into the user stats and let the user metrics and usage patterns guide their decisions.

“We started measuring what the user was doing with our product,” says Gatius. “And looking at that [data] we are constantly changing small things to make it easier [to use] and eliminating functions that are not being used.”

A good example of this type of change cycle was when the team noticed that the data showed users were frequently accessing the library within the application. The design team decided to move the library to the main menu to give users quicker access.

• **Expert review:** The Dixio team commissioned an expert review of the functionality, tools, and content from an outside agency.

• **User trials:** Users at a school were given a trial version of the Dixio app and allowed to use it for a limited time. The students were surveyed after using the tool, and results from their feedback were rolled into the product.

• **Observing users:** An effective method the team has employed is to take a group of users who are unfamiliar with the product and just sit them down in front of it and say, “use it” and see what happens. And this has produced some of the most surprising results. “This is normally a process that always surprise you,” says Gatius, “because when you are working on the same thing for a long time, you know all the details behind the screen, and you don’t see what users do, which may be staring right at you on the screen. So by observing people that have no experience with [Dixio], we get important information to improve the product.”

One of the most surprising findings from these observation sessions is that no matter how hard the Dixio team tries to push the envelope and provide users with a robust contextual lookup via the “capture” function (and no matter how much they try to educate the user about the same), most of the users (approximately 60%) are still using the tool for a very simple and traditional type-a-word-in-the-search-box definition search.

**Implementation**

**Designing the Application Engine**

The application engine is written in C++ and there are cross-compiling tools that are used for each platform. “We implement an upper adaptation layer that interfaces with the devices,” says UI Designer Laura Vilar. “For example, the Android
application has a C++ core and is compiled for that processor, but we need to put an adaptation layer on top that is able to communicate within all the Android systems.”

This same approach will be applied across other devices and platforms such as tablets and iOS. At the same time, the team is working on the online engine. This has created some differences between the online engine and the local device engine, which the team is working to resolve.

“This same approach will be applied across other devices and platforms such as tablets and iOS. At the same time, the team is working on the online engine. This has created some differences between the online engine and the local device engine, which the team is working to resolve.

“Some of the features of the online engine are not available on the local engine,” says Vilar. “So now we are starting the conversion process to have exactly the same on both platforms. We are working to transfer the benefits of the C++ technology on the devices over to the server engine.”

**Moving from a Device Model to a Server Model**

Next on the company’s roadmap is an effort toward moving the application processing from a device model to a server model; for example, by allowing the local engine that is supplied on the eReader Solution to also call to the Dixio server for supplemental content.

One of the problems of this model is of course the space limitation on mobile devices. The application accesses large dictionaries and that creates a storage space problem on mobile devices. So the solution is to offer complementary information that sits on the server.

“So [mobile devices] would have the ability to look for online resources,” says Gatius. “You would get the local immediately, but if you want the online ones, you would click the *Online* tab, and have access to more dictionaries online than you have locally.”

Moving toward an online model means using different technology. The online engine was developed using standard Web languages and standard server languages. “We’ve determined that we could improve the server engine a lot by using low-level languages like C++, and we would benefit as well from having common code, common functions, and common behavior for local and online engines,” says Gatius. “So our plan is to, as much as possible, synchronize the modules running on our servers with the local device modules—while taking into account the fact that some language processors can be too resource intensive for local devices. Maintenance will be easier if the engines are fully synchronized.”
<table>
<thead>
<tr>
<th>Year</th>
<th>Event Description</th>
</tr>
</thead>
</table>
| 2006   | • Gotitclear (predecessor to Dixio) released. This is a desktop dictionary/look-up app. The goal of this app was strictly just to sell local dictionaries.  
• Gotitclear application and dictionaries are able to detect inflected forms and idioms. |
| Q3 2007| • Gotitclear product renamed Dixio and moved online to make app lighter and faster and remove processing/space limitations of desktop environment.  
• In the new tool (www.dixio.com) users can look up words, create accounts and manage their own dictionaries.  
• Programming begins on Dixio for Web (DFW), a web app for use on any website.  
• Both products work thanks to the linguistic server, the CKE (Cloud Knowledge Engine) that is released in 2007. |
| Q4 2007| • Dixio online beta (www.dixio.com /DFW) released.  
• Development begins on Dixio Lite, a thin client for Windows. |
| Q1 2008| • Dixio Lite released (thin client Windows PCs.) This product only allowed capturing and displaying definitions on the desktop.  
• Preference management and dictionary management are now handled online. |
| Q2 2008| • Development begins on Dixio Desktop for Win, Mac OS and Linux.  
• Dixio desktop is a multiplatform online desktop application with library management and Cloud storage. |
| Q4 2008| • Dixio Desktop v.1.0. for Windows released. |
| Q1 2009| • Dixio Desktop v.1.0. for Mac OS and Linux released. |
| Q1 2010| • Redesign of Dixio Desktop (Windows) v2.0 begins.  
• Throughout 2010, the Cloud intelligence of the CKE is improved:  
  o The app identifies the language of the text the user is reading. |
The app makes a linguistic analysis of the captured text.
The app indexes online encyclopedic content (like Wikipedia), which is continuously updated.

### Q2 2010
- Dixio Desktop v2.0. released
- Dixio.com is no longer available; instead a new desktop version is released with new design and usability rules. The main functionality is the same but the GUI and engine changes significantly.
- Team defines metrics to analyze user behavior (such as consulting the library, capturing a word, or using the search box for manual lookup.)

### Q2 2011
- Redesign of the Dixio Desktop v.3.0. for Windows.
- User experience testing and modifications to improve usability.
- Develop and implement Dixio for e-readers (Dixio eReader Solution.)
- Dixio is installed in devices such as Movistar ebook bq, Fnacbook and Tagus.

### Q3 2011
- Development begins on Dixio for tablets, beginning with the Android OS.

### Q1 2012
- Launch of new Dixio Desktop 3.0.
- Launch of Dixio for Android.

### Plans for 2012 and beyond
- Launch of Dixio for iOS.
- Changes to improve the CKE (Cloud Knowledge Engine) intelligence.
- Expanded language options, including French and German.

## LESSONS LEARNED

**Listen to users.** The primary lesson the Dixio team has learned through their design efforts is the importance of getting actual information from users, whether it be analytics about their behavior or direct conversations. “Listen to the users and try to give them what they want and what they need,” says Vilar.
## TEAM

### DIXIO PROJECT TEAM ROLES

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
</table>
| **Product Manager**
Ferran Gatius    | Determine features and functionality                   |
| **Developers**
Tomas Couso (agency)
Jesús Liarte
Francisco Javier Hidalgo
Blit Software
(Agency for Desktop app) | Implement technology solution and perform bug fixing |
| **UI Designer**
Laura Vilar | Design interface and graphical elements                |
| **Quality testing**
Joaquim Triginer | QA testing                                            |

Pictured (left to right): Jesús Liarte (Developer), Laura Vilar (User Interface Designer), Ferran Gatius (Product Manager), Xavier Hidalgo (Technical Product Manager), Joaquim Triginer (QA Testing.)
Pictured (left to right): David Llort, Carlos Martin, Daniel Ricart, Maria Monterde, Montserrat Civit, Laura Vilar, Francisco Javier Raya, Ferran Gatius, Christian Cicceri, Laia Pacheco, Javier Miguel Sastre, José Córdoba, Joaquim Triginer, Sergi Fernández, Marc Centelles, Xavier Hidalgo, Jesús Liarte, Antonio Riu and Fernando Ferri.
Highlight

Organization: Cohdoo, LLC
Application: Record and Annotate Audio
Headquarters: Portland, OR
www.cohdoo.com

Design team:
• Ryan Russell, Designer
• Eric Wuehler, Developer
• David Stewart, Design Consultant

WHY THEY WERE CHOSEN
Highlight was selected as an outstanding example of a multiplatform experience because of several elements of its user experience:

• Simplicity: By maintaining a tight focus on the core functionality, the application omits clutter and confusion, and delivers an intuitive experience.

• Flexibility: Optimized for phone, tablet and personal computer, Highlight offers a feature set and interface tailored to fit each device and context of use.

• Purpose: As our gadgets proliferate, we are overwhelmed with tools for creating content but far too few of these help us manage and consume content in a meaningful way. Highlight addresses this need with an elegant system for extracting meaning from content and sharing it with others.

OVERVIEW

Main function: Highlight lets users with iOS devices make audio recordings, mark interesting points, and share recordings with others.

Why it’s unique: Quite a few audio recorders are available for the iPhone - including the standard recorder provided by Apple; Highlight distinguishes itself with an incredibly intuitive design. The interface presents both basic and advanced features in just a few self-explanatory screens.

The organization: Cohdoo, LLC is a startup created by Russell and Wuehler. Since Cohdoo is a side project for both men, they constrain their project scope by aiming to design, code, and release each product within 90 days. This timeframe limitation helps ensure that the project stays manageable and the end product isn’t rendered obsolete by changes in the platform or market.

Design philosophy: A simple principle guided the design of Highlight: a smaller focus leads to a bigger impact. Russell explains, “By focusing on one thing you can have a much bigger impact. There’s a quote I really like, that goes something like, ‘Your ability to have an impact is directly related to your willingness to narrow focus.’ We wanted to have an impact with our application and delight people, so we knew we’re going to have to put all our effort on one thing, because if we try to do a lot it’s just going to be mediocre.”

WHO ARE THE USERS?
Highlight’s core user group consists of people who conduct in-person interviews for ethnographic or design research. Russell explains that he knew from personal experience how time-consuming it is to analyze audio recordings and extract key insights.
“Synthesis and processing of recordings takes so much time. If you’re in the field an hour, that could mean two, three or more hours of processing. [People] who work in design firms say, the client pays for us to do all this research and they want us to move instantly into design work. It’s like, synthesis is just a box to be checked; they want it done [instantly]... What good is 100 hours of fieldwork if it never gets used?” says Russell.

The Highlight designers aimed to meet the specific needs of user experience researchers, calculating that doing so would also lead to a product that was good (if not perfect) for other types of users, such as:

- Journalists
- Lawyers
- Students and Professors.

Russell says, “If we can delight our primary user — designers — we hit a whole secondary ring around them that will be pretty happy too: anyone who does audio recordings. It may not fulfill exactly their need, but it will be pretty close.”

After Highlight was launched, its creators found that it was especially valuable to a particular group that they had not previously considered: students taking courses in their second language. “Highlight allows them to focus more on what’s going on in the classroom and not have to struggle so much with what the professor is saying, because they can go back to it later,” says Russell.

**THE APPLICATION**

Highlight lets users create audio recordings using an iPhone, iPod Touch or iPad, and offers

- a simple, intuitive interface
- the ability to easily share the recording with others
- the ability to mark particular points of interest within a recording.

Making an audio recording seems like a straightforward task, but many of the applications targeted at this audience are surprisingly complex, offering little visibility into how files are created, stored and managed. Highlight stands out from the pack because it features incredibly simple controls for starting, stopping, editing and sharing recordings. Russell explains, “What makes Highlight special is its simplicity. We worked very hard to remove every unnecessary element and feature from the application until we got Highlight down to its pure essence. There are only five screens in the entire application, and our users consistently comment on how easy and intuitive Highlight is to operate.”

Highlight’s users need to share their recordings with others, but just sending an audio file to colleagues is not enough. Many research interviews are one to two hours, but few designers have time to listen to hours and hours of audio recordings. Highlight lets the field researcher mark particular points of interest within a recording; listeners can then skip straight to these points, vastly reducing the amount of time needed to consume the content.

In addition to marking highlight points while making a recording, users can make a few modifications to the recording file after it has been completed:

- Name the recording
- Add a picture
Highlight lets users create audio recordings, mark points of interest within the recordings, and share them with others.

INNOVATION

As technology advances and application users work on multiple devices, it makes sense to expand an application experience to reach them on new platforms. But it is essential to approach new platforms in a thoughtful way, to ensure that the features and offerings are appropriate to the platform.

The ability to share recordings with others is essential for Highlight’s core user group. The smartphone is an excellent tool for field researchers who need to create recordings without hauling around large devices or accessories. But the personal nature of such mobile devices makes them less than ideal when it comes to making content available to more than multiple users.

Since the standard audio players can’t display the unique metadata (such as point-of-interest markers) in Highlight recording files, Highlight’s designers created Highlight Player, a companion application for consuming Highlight recording files on a desktop or laptop computer.
The feature set for both Highlight, the mobile application, and Highlight Player, the desktop application are tailored to address the users’ tasks while keeping the device capabilities and limitations in mind.

For example, early plans for Highlight included features such as the ability to tag recordings with different labels. Eventually the designers realized that while the phone was great for capturing recordings, the ultimate purpose of doing research — to influence the team’s design decisions — was better served by encouraging users to get the recordings off their phones and onto another device or platform where more people could access the content. As Russell explains, “We’re not against people storing recordings on the phone but we really see the value when you get it off your phone. When you start to proliferate it through your organization, your team, the community — then it becomes much more powerful. ...So instead of promoting the ability to hoard on your device, we want to promote the ability to push it off your device.”

Likewise, the Highlight Player could have included many additional features, such as the ability to edit and manipulate the content; however in the interest of keeping a consistent experience and releasing this essential tool in a timely manner, the designers ended up going with a very simple tool. As Wuehler explains, “We needed something to let people give their teammates the benefit of the highlights. Because of that we had to build a player. ...We kept the functionality focused...so people can play back what someone else created. This let us limit the scope enough to quickly get it to market. We thought about shipping with some basic editing capability, but we ended up pulling it back in order to get the playing capability out the door.”
Highlight encourages users to transfer recordings out of the application and onto another platform via email or using Dropbox (a popular third-party file sharing service.)
The Highlight Player is a simple tool that lets people listen to highlight recordings and skip to specific highlight points on a laptop or desktop computer.

Highlight’s also takes a proactive approach to saving content. Russell explains, “There’s really no concept of save anymore. If I write on a piece of paper, it doesn’t magically get up and vanish. If you do work it’s always there. You can get rid of it later if you want to, but you don’t have to explicitly tell me to save it. It’s valuable because you spent time making it.”

The first version of Highlight gave users who paused a recording the option to resume, save, or cancel the recording. To minimize accidental deletions, the ‘cancel’ button was followed by a confirmation dialog. “It just felt awkward; I hate having to ask people, ‘Do you really want to cancel?’” says Russell. When working on the second version, Highlight’s designers revisited the ‘cancel’ scenario and realized that there really wasn’t a need for a cancel button at all — because the only time it saw significant use was while testing the application or demonstrating its functions. The few cases when real users needed to discard a recording were easily addressed by the ‘delete’ functionality available elsewhere in the application.

Acting on this principle of ‘everything the user creates is valuable,’ Highlight Version 2 now offers a single button labeled “Done,” rather than a choice to either save or cancel a recording.
INFO@NNGROUP.COM Executive Summary

Left: Highlight saves every recording; the secondary action of deleting a recording can be accomplished through the home screen. Right: A previous version of Highlight gave users the option to either save or cancel a recording.

CONSISTENCY

The Highlight Version 2 recording application is designed to work on both the iPhone, iPod Touch, and the iPad. While the layout changes to take advantage of the iPad’s larger screen size, the content and functionality remains the same, creating a consistent user experience regardless of the device type.

On the iPhone, the home screen displays all the recordings in a vertical list, with thumbnail images showing the picture associated with each recording.

The iPad layout uses the same picture as an icon for the recording, in a horizontal array. The delete control is also placed differently in the iPad layout. On the iPhone it appears on the top left corner, but on the iPad layout many users would be likely to overlook a control placed so far away from everything else; instead, it is moved to the top right corner to form a sort of ‘utility controls’ area of the screen. Eyetracking research shows that people tend to look in the top right corner for such controls, so this placement conforms to user expectations. (The placement works on the iPhone version because the screen is so much smaller that everything is closer together and easily noticeable no matter where you are looking.)
The Highlight is designed to work on both iPhone and iPad. The iPhone home screen layout displays recent recordings in a vertical list.
The iPad home screen layout differs slightly from the iPhone version, but maintains the same content and features for a consistent experience.

Similarly, the Highlight Player application expands on the Highlight Recorder interface but maintains a consistent experience. While the player interface adds visual indicators of highlighted points on the timeline and a details panel on the right side, both interfaces use the same simple playback controls and list of highlighted points to let users jump to important moments in the recording.
The Highlight iPhone application allows users to play back recordings from the Details screen, using simple controls and a list of highlighted points in the recording.
The Highlight Player maintains the same simple interface as the mobile and tablet application: simple playback controls with just a bit of additional detail, such as the timeline markers indicating where highlights appear.

**FLEXIBILITY**

Applications should strive to accommodate the user’s task, rather than forcing people to change their behavior in order to use the application. Highlight includes several features specifically tailored to the interview and recording process to make the researcher’s task easier.

If you’re interviewing someone it’s probably because you need information they have, but you don’t. That is, you don’t already know what they’re going to say. This obvious circumstance has important implications for the process of marking points in a recording, because many people won’t recognize the points they want to mark until after they have occurred.

Highlight makes it easy to mark these moments that have just passed with an offset feature, which automatically adjusts the marker time to be 2, 5 or 10 seconds in the past, depending on your settings. (You can also choose ‘no offset’ to have markers applied in real time.)

Instead of a stop button, Highlight uses pause as the primary control for ending a recording. Users love this simple accommodation, absent from many other recording applications, which makes it easy to omit dead time (such as bathroom breaks) from the final recording without stopping and creating several different files.
The offset setting lets users place markers a few seconds before they actually tap the highlight button, more accurately identifying the beginning of important points.

Researchers can take a picture of their subject and associate it with the Highlight recording, which helps to ‘put a face to a name’ for people consuming the research later on. It also allows users to browse recordings by either name or thumbnail image.
The Pause feature allows users easily to omit inactive periods from a recording, the quickly resume when the interview begins again.

INFORMATION DISPLAY

Although Highlight deals with a relatively small volume of information, the screen size limitations of mobile devices make it essential to convey information clearly in a very small amount of space.

The Highlight home screen has evolved through several iterations, each one increasing the utility of the screen by providing access to relevant data. In early sketches, the home screen was simply a menu listing the main tasks, such as starting a new recording. Existing recordings were listed on a separate screen. As the design progressed, the home screen evolved from a simple task hub into a more information-rich screen displaying recent recordings, as Russell explains, “Initially we thought, ‘We’ll have a button that goes to all of them and you can pick.’ Well really, people want to get to the ones that they just did, so how do we make that easy and straightforward? ...We’re trying to help the user get to the information that they want quickly, without having to dig through the interface.”

The process of optimizing Highlight for the iPad actually led to further revisions of the iPhone home screen. Showing only the three most recent recordings left a great deal of empty space on the iPad screen, which sparked the idea to show all recordings rather than just the most recent. This idea was then applied to the iPhone design: Now the iPhone home screen also shows a full list of all recordings as a vertical list. “Now instead of the last three recordings, I can show all of them, and we’re able to get rid of the all recordings screen and provide an easier to use program, but it was really created because of that iPad work,” explains Russell.
Early sketches for the home screen included just a menu of pages and did not display any actual recording titles.
Left: The first version of Highlight for the iPhone had six screens, including one screen to display all recordings. Right: Version 2 of the application displays all the recordings on the home screen, rather than on a separate page.

In an early prototype of Highlight, ratings were indicated with blue stars. If a user hadn’t yet rated the recording, it displayed five white stars. So the star was both the method for assigning a rating and the method for displaying a previously assigned rating, with only the color difference to distinguish between the two states.

Because stars are so frequently used to display ratings, many users guessed at the display function and assumed that the five white indicated a five star rating, never guessing that they were in fact an input method.

To eliminate this misunderstanding, the interface was changed to use different elements to signal the input and output of the rating: before a rating has been assigned, five small dots appear in that row, and these dots change into stars when tapped, to display the assigned rating. Separating the two functions into two different visual indicators is much more effective at communicating their purpose than relying on just one symbol to fulfill two purposes.
Left: In an early prototype, users did not realize that the white color indicated that there was no rating. Right: When tapped, the stars turned blue to indicate the rating.
The most noticeable elements of the Highlight interface are the enormous *Start* and *Mark Highlight* buttons; each of these takes up more than half the screen. This approach is unusual among mobile applications, where screen real estate is at a premium; but the emphasis is extremely effective at optimizing the screen for the single key task users need to accomplish.

Because Highlight is intended to assist users while they are primarily engaged with another activity, conducting an interview, the design acknowledges that the users’ attention is *not* primarily focused on the screen, especially during the actual recording process. Russell explains, “The highlight button that allows users to gracefully touch it while keeping eye contact with their participants, an important part of building rapport.”

Early sketches for the *New Recording* screen devoted more space to other information and elements, such as the elapsed time indicator and a list of recently created highlights. But eventually the designers realized that those items weren’t really needed while making a recording, and that the user’s activity would be better served by placing even more emphasis on the *Mark Highlight* and *Pause* buttons, the two primary actions that every user will need to perform quickly.
The *Mark Highlight* and *Start* buttons dominate their respective screens, occupying more than half of the screen real estate. The buttons are easy for interviewers to click.
Early sketches for the recording screen devoted less space to the Highlight button and more space to other items such as the elapsed time and data about recently created highlights.

The playback area of the Details screen includes controls to play the recording, skip back ten seconds, and create a new highlight. Highlight users report that skipping backward is one of their favorite features, and both that and the triangle control to begin play are easy to recognize. The third element, the light bulb symbol, however, presented some difficulties. It is used to represent the action of marking a highlight, but some users did not notice or understand this element. The light bulb symbol is used in several areas of the Highlight application, sometimes to represent an action, and sometimes just as a branding element. The dual meaning of this element and its placement on the far right of the Details screen, away from the timeline, combined to make this element appear to be purely decorative rather than functional. To address this issue, Highlight Version 2 has adjusted the placement of the three playback controls so that they appear close together, all directly over the timeline, to make it obvious that they are all related.
Left: In Highlight Version 1.0, the control for creating a new highlight appeared on the far right of the Details screen, directly over the end time display. Right: Now in Version 2.0, the control has been moved in closer to the other playback controls, to make its function more obvious.

Although all the elements on the Details screen can be edited by tapping and holding them, this screen also includes an explicit Edit button in the top right corner. Users will probably learn about the advanced selection options after they gain some experience with the application, but the visible signal of a button is extremely helpful for new users. This is especially true for touch interfaces: with a cursor, users are already interacting with the interface and moving the mouse around, often automatically pointing at the element they’re interested in, which creates the opportunity to communicate with hover effects. With a touch interface, however, there is generally no contact between the user and the interface until the user has already decided what to tap, so the functionality must be clearly communicated by the basic interface.

Russell and Wuehler chose to label the button Edit to help keep the interface simple enough to use immediately, without any instructions or help files. As Wuehler explains, they felt that “if it was complicated enough to need a help file, it was probably too complicated. ...That was definitely one of our goals: to make it obvious enough that somebody can just pick it up and use it? ...That’s why we chose the word ‘edit’ instead of a pencil or something: because that’s obvious. If we’re not going to include a help file, we have to make it self-explanatory.”
A final interesting aspect of the Highlight interface is an element found on many mobile applications, but not in Highlight: a bottom navigation bar. During the early planning stages of the design, Russell and Wuehler considered including persistent menu bar as well as more navigation controls between different screens of the application. Ultimately, they decided against these options in order to keep a strong focus on the primary task of each screen. For example, when creating recording, the entire screen is devoted to that task and the user is expected to be immersed in the process of managing the interview and marking highlights. Because the task is resource-intensive, it’s highly unlikely that users would want to multi-task and do other activities within the app (such as edit a different recording.) So a persistent menu bar giving access to other parts of the application would complicate the recording screen and add the potential for errors, while offering very little benefit. Accordingly, Highlight’s designers went with an immersive navigation experience where users drill up and down from a home screen rather than a more flexible persistent menu bar which would have given users access from any page, to any page.
DESIGN PROCESS

Usability Testing
Highlight’s creators subjected each stage of the design to quick, guerilla-style usability testing. Friends and family served as test subjects, both for convenience and to make sure that the application would be easy enough for anyone to pick up and use immediately, regardless of technical ability. Wuehler explains, “I would make a build and hand it to my twelve-year-old and say, start recording and mark two highlights. Once she had seen it a few times she could do it so I had to switch to my wife and my other kids…it was kind of a poor man’s usability test.”

Findings from these tests were immediately incorporated into the design. This lightweight methodology allowed the designers to explore a variety of design options in a short amount of time. “We could experiment, try stuff, get feedback, and iterate in these really tiny loops. So we got to try tons of things in a short period of time to get the winner, versus doing stuff that was very heavyweight and only getting to try two or three options because the testing was expensive, or done at the end but nothing at the start,” says Russell.

Beta Testing
With any application, and especially with mobile applications, actually using the program reveals issues that don’t emerge in simulated settings. For example, the possibility of getting a call that interrupts your recording, or having files that are too large for email. Local designers helped beta test Highlight by actually using it for field research, which yielded valuable design insights. Russell explains, “One of the biggest things they said is they really want to get [the recording] back to their broader team… initially the only way you could share the recording was to email it to someone. People said, ‘We want to share it in other ways, we can put big files in Dropbox but the email server can’t handle something too big.’ That was something we learned in research, and were able to add.”

Implementation
While Russell sketched iterations of the front end design, Wuehler began developing the back end functionality. This simultaneous approach required significant up-front communication and ongoing check-ins between the two, but ended up being far more efficient and productive than traditional linear approaches. “It worked well because as we progressed we had both sketches we could play with and stuff we could try that was actually in code. The two together were really powerful, because we learned a lot in development that informed the sketching,” explains Russell.

Another benefit was that because the development was initially focused on just the back end functionality, the front end user interface code was left unfinished until later in the process, after sketching and testing had refined the design into a better user experience. From the development perspective, Wuehler felt that this approach improved both the design and the code, “Really tight collaboration between the design and the development allowed design to understand what the development constraints are, and the flip side, having design push me to think a little bit differently about how I might be able to do things.”

Wuehler’s early development efforts focused on created a ‘black and white’ functional prototype, built mostly with the standard widgets in the iPhone toolkit. This
The prototype actually helped define the structure of the user experience, because it illustrated the basic platform behaviors shared across many applications that would be familiar to platform users and easy to implement. While the final application doesn’t consist solely of these standard elements, it’s helpful to know the default way of doing something that will be easiest to implement and easiest for users to understand. “Then you can decide, is the tradeoff worth it, having to teach your users something new as opposed to trying to find something that they’re more comfortable with,” explains Wuehler.

The process of creating a simple functional prototype and then integrating it with the polished user interface design makes for an efficient and graceful process, as Wuehler explains, “With design, you can imagine anything... With development you’re a little bit more restricted with what you can do easily or quickly. By agreeing on the basic functionality up front, then as a developer I know I can [build it] in time. So this black and white design and feature agreement... gets us 80% of the way there. As we start to merge the black and white with the design we can finish off that last 20% of marrying the features to the design.”

**Left:** The ‘black and white’ functional prototype of the Highlight Home screen.  
**Right:** The functional prototype of the New Recording screen.

The close collaboration between design and developed saved time, both by helping to quickly determine which design alternatives would be most practical to build, and by solving the major usability issues while the interface design was still in the form of sketches rather than code. Russell explains, “We really had very little rework. In many projects...a bunch of stuff gets coded, you do a big usability test halfway through the project and [find problems.] Now the cost [to fix it] is huge because we’re almost done. Our rework... was pushed to the lowest-cost step of the project. By collaborating with sketches intensely, there was more rework on sketching than
there was on Photoshop, and more rework on Photoshop than there was in the code. We went through quite a few iterations but it was very inexpensive.”

**Long-term planning**

Highlight’s creators also strive to keep future product plans in mind during the design process. 'We try to design out a couple of features ahead...For example we know there’s a couple of things that we won’t have time to build into Highlight Version 2. But we’re designing out what those should be so we can make sure that as we’re developing it now we don’t box ourselves into a corner,” explains Russell.

This approach applies to both specific features and to entire applications, and this philosophy of long-range planning and reuse has worked well. For example, a product called the Highlight Converter used to be offered as a way of transforming Highlight files into several different audio formats. The Converter was a less than ideal solution as it did not preserve any of the highlight metadata, but Wuehler was able to re-use the core basis of the Converter when he built Highlight Player. Russell and Wuehler are now considering creating a web service for sharing Highlight recordings, and plan to re-use the code from the Highlight Player. “Because of limited development resources, we have to figure out how to achieve multiple things with one product. We think about not just features but technology use one or two products ahead, and how, in what we’re building today, we can re-use some of those guts tomorrow,” explains Russell.

**LESSONS LEARNED**

Russell and Wuehler shared a few lessons learned from this design effort:

- **Get your design onto the device ASAP.** Interface elements may look big enough on a monitor, but it’s not until you actually see them on a small device that you understand how easy or difficulty they will be to manipulate. “Regardless of how you’re rendering the design, you’ve got to get it on the device really quickly. Even just saving it as an image file and putting it as a photo on your phone. On a monitor...everything looks big enough to touch, but as soon as you put it on a device, things are too packed together, they’re too tight and not enough room for my fingers in there. So now I render in Photoshop and copy it over to my iPhone or iPad to quickly see what it looks like and if the spacing’s ok,” says Russell.

- **Don’t try to be clever.** “If we’re trying to be clever it usually means we don’t have the best solution, and we need to keep working. Because clever usually equals ‘hard to figure out.’ It’s only clever to the person who came up with it,” says Russell.

- **Fight Feature creep.** “With Highlight, we’ve been able to prove that when you focus on figuring out what the heart of the application is... you can create something that’s really magical and that delights people. There’s always a push for more features, so it was fun to build something focused and spend our time removing, not adding... We can be better than our competition by doing one thing well, even though they do 10 more things than us. There’s a creative director at a local design firm who uses Highlight as an example for his clients. When their client wants a million features, they show them Highlight and say, ‘Look how easy it is to use, look how delightful it is. We can make your app like this.’ It’s exciting for us to have someone using it as an example of that philosophy,” says Russell.
TEAM

Highlight was created by Cohdoo, a small Portland, Oregon startup composed of:

- Ryan Russell, Designer
- Eric Wuehler, Developer

“We are focused on helping individuals bring their ideas to life. As UX professionals we had a need for some additional tools to help us get our job done. We made a few tools for ourselves, others liked them too, so we decided to go into business,” says Russell.

“The right gear in your pack makes all the difference. It allows you to be agile and ready for what’s ahead. At Cohdoo we believe great gear does what you need it to without getting in your way. It’s not about the gear, but what the gear allows you to do.”

The company name is taken from “Coh doo,” words that rang throughout Russell’s home as his little boy. This small phrase was the beginning of a more complex declaration, “I can do that.” In those first words an experience was built, an idea expressed: Cohdoo’s goal is to create products that help grow others’ ideas from infancy to maturity.
Mobile Unified Communications (UC) Mobile Client
(Verizon)

**Organization:** Verizon Wireless

**Application:**
The Mobile Unified Communication (UC) client allows business customers to use two different phone numbers, their mobile and business lines, on a single mobile device.

**Headquarters:** Basking Ridge, NJ (USA)

**Design team:**
The application was designed in-house at Verizon Wireless in Waltham, MA. It was designed internally by a cross-functional team drawn from members of the company’s Device Marketing User Interface Design group, Product Management and Product Services Integration teams.

**Members:** Linda Borghesani, Manager, Device Marketing; Samantha Kalita, Consultant, Device Marketing; Graham Sysko, Consultant, Device Marketing; Kathleen Murphy, Manager, Product Development; Victor Alba, Manager, Product Development; Yeharn Hwang, Distinguished Technical Staff, Technology and James Elter, Manager, Technology.

**OVERVIEW**

**Main Function:** The Mobile Unified Communication (UC) client allows business customers to link two phone numbers, their mobile and business lines, with their mobile phone. It runs on Verizon Wireless Android Smartphones and enables customers to use one mobile phone both at their desk and on-the-go to make and receive personal and business calls.

**Why It’s Unique:** Simply put, this application is unique because it lets the user be connected anywhere. No matter what kind of call a user needs to take – business or personal - he can receive it on a single mobile device, on his mobile phone, seamlessly. Additionally, when on-the-go, users can place business calls and select whether their caller ID appears from their business or mobile phone number. Plus, this app affords them business capabilities they would have on a desk phone such as merging and switching between calls, transferring calls, and creating up to a four-way conference call while mobile.

**The Organization:** Verizon Wireless is the largest mobile communications network operator in the United States. It had 108.7 million subscribers as of the end of 2011, the largest number of any wireless service provider in the United States.

**Design Philosophy:** The genius behind this application is how effectively the design team has put a simplified face on top of complex technical architecture. Though there are a lot of complicated things going on behind the scenes, the user is exposed only to a coherent, smooth interaction with quick access to what he needs, when he needs it.

**USERS**
The primary audience for this application casts a wide net. It includes any business customer who has an Avaya, Cisco, or BroadSoft PBX in their office and has the need for employees to be mobile. This means users are primarily (though not exclusively) employees of larger organizations, which are more likely to have such PBX systems.
THE APPLICATION

The Mobile UC client balances the traditional office phone features with the convenience of a mobile phone. The UI is intuitive, simple, and covers a variety of tasks as well as user scenarios, all making a great user experience.

The first interaction with the Mobile UC client, the Device Setup page, helps users get their app set up in the way they need it. A wizard-like process leads people through the set-up. And while a wizard process can be overly constraining at times, designers in this case avoided this by planning for various user scenarios to make this process helpful and productive for all. These scenarios include users who:

- Have a hardware dock to use with the app and their mobile phone: A message on the Device Setup page suggests that users with a dock should dock the device to assist with set up. The docking station enables users to automatically configure their service after they have been provisioned for the service by their PBX administrator. (A separate PC-accessible administrator configuration tool is provided to PBX administrators for provisioning users.)

- Have an email from their system administrator with configuration information: A message on the Device Setup page suggests that users with an email from a system administrator select the Import button.

- Desire or need to set up the app for himself: The user may choose the Manual setup button on the Device Setup page.

- Don’t have time or cannot set up the system now: The Set up later button defers the configuration steps.
A wizard walks the user through the setup process, during which he may import a configuration file, manually set up, or dock the phone in order to easily configure the service.

The docking station was designed to replace existing desk phones. It is specifically designed to have a universal cradle, so that different devices can fit into the same type of dock, even though each device may have its micro-USB connection in a different place. Additionally, it goes the extra mile to connect to mobile phone with cases on them: No need to make your mobile naked to use this dock.

When the mobile phone is docked, Mobile UC turns a regular mobile phone into a powerhouse desk phone, as it supports the following functionality: corporate directory lookup, multiple-line and shared-line access, transfer, conference, “do not disturb,” mute, hold, voicemail, speed dial, and call history of incoming and outgoing calls. The docking station also charges the phone battery, and the phone can be used for all other smartphone functionality even while it is docked.

Mobile UC also allows users to keep the current call active while they either dock or undock the mobile phone, so users don’t need to pause or be disconnected when they need to leave the office or arrive and want to dock.
An accessory, the universal docking station supports the Mobile UC application while a user is sitting at his desk. There is a universal connector to support many devices in any desired screen orientation.

When the device is docked, the Mobile UC home screen allows users to accomplish many tasks, such as: call, check voicemail and call logs, and retrieve contacts and favorites, including any smartphone functionality. Designers chose to display each command as its own button in a flat display rather than consolidating in menus, drop-downs, or panels. Since there is only a small number of frequently used commands, the display accommodates them well and is a smart choice. The screen shows the following large buttons for easy access: Phone, Voicemail, Call Logs, Favorites, Contacts, Applications, Settings, Add, and Undock. The user may customize and reorganize these buttons. Buttons show recognizable and recallable icons along with clear text.
The Home Screen for the Mobile UC application when docked displays the following default buttons: Phone, Voicemail, Call Logs, Favorites, Contacts, Applications, Settings, and Add. The user may customize and reorganize these eight shortcut buttons.

But not all important commands are presented at that flat top-level. The designers made decisions at each step of the way about the most expected and helpful places to put each command in the call workflows. For example, in the case of making a call, designers chose to use progressive disclosure—hiding some commands until the user is ready for them. Once the user selects to make a call he is then presented with the options to make a Business call or a Mobile one. Designers could have placed these commands at the home level but it would have made that page too busy, and potentially confusing to offer types of calls at that level.

These buttons are color coded so users may easily pick them out, and the colors are carried through the UI. If you are on a business call it is indicated in blue and a mobile call is indicated in green. The call type also displays in the top status area above the buttons.

Users may also add customized widgets their device’s OS home screen, so that when the phone is not docked they have a shortcut to make calls of a specific type. The widgets can be customized to Business, Mobile, or Prompt (which prompts users to choose a call type every time they place a call.) A second widget, Quick Call, changes appropriately depending on the default settings so that a user can make a call quickly.
Mobile UC Widgets allows users to easily select and change their calling preferences while undocked. There are two Mobile UC widgets: the default call type and Quick Call which can be installed. The widgets are color coded to match the application where business calls are shown in blue and mobile calls are colored green.

If the widget is set to Prompt, then the user is presented with a screen that allows him to then choose Business or Mobile. This UI helps people who make roughly an equal number of calls from their business and mobile lines. So regardless of her usage scenario—mostly make business calls, mostly make mobile calls, or make roughly an equal number of mobile and business calls—the UI lets the user be in control and happy.

To set the default call type selection users may tap the Always place calls this way checkbox.

The buttons are quite large, easy targets to hit, which comes in handy when the user is walking or otherwise moving while using the phone. Designers strove to make the buttons throughout the UI as large as possible so people who are moving, and yes, even driving, can make selections quickly and error freely, having no-miss experiences.

When placing calls while undocked, the user has the option to select whether to place the call using their business or mobile line. Business calls are shown in blue and Mobile calls are colored green.

Incoming calls are as easy to deal with as outgoing ones are in Mobile UC. When the phone rings the user is prompted with a screen with the most important elements displaying, including:
- Action items: large buttons to Answer or Decline the call, and
- Status items: the caller’s name, telephone number, and image.

The affordance on these items is appropriate, as the status items are legible and recognizable, and the action items (buttons) appear to be interactive elements in all important areas, like: size, visual treatment, and verbal terms used as commands.

When receiving a call, users are prompted with caller ID information and the ability to answer or decline a call. If they have multiple lines, the incoming line information is also displayed.

Once on a business call, the user may create a conference call or transfer the call. (On a Mobile call the phone will be as it was before the app.) Users may choose who to transfer the call to by tapping in their contacts, call log, or favorites.

If a user decides to transfer a call he may enter the number or select from the call log, their contacts, or their favorites. He may also have the choice of talking first (consultation), transferring without talking, or sending the call to his voicemail.
The panel displays the length of time of the call in the lower left and displays the time of day in the lower right. The name of the person on the call is clearly displayed in the upper left. The commands to add more people to the call, transfer, and end the call are all visible and are easy touch targets. The user may also conference in another person, or make another call by tapping the New Line button.

![Conference Call Interface](image)

When a user is on a conference call, the other callers are displayed, and the UI indicates that all parties are on a conference call.

The user may end the entire conference call, or even disconnect the last user who was added to the call.
All callers are shown during a conference call. Users can disconnect from the last person they added to the conference call by selecting the *End Last* button.

Sometimes a user may want to say a few words to a caller before adding him to the conference call. Imagine you are talking with a manager or client and she needs another person to answer a few questions. You may want to call the second person and say just a few words to brief him before conferencing him in with the manager or client. In these cases the user may choose the *Talk First* button.

After talking with that third party, the user has the option to add him to the conference or cancel adding the person to the conference call.

When docked, users may have up to three concurrent calls and may start a new call by tapping the *New Line* button. This button is only visible adding a new line is an available option. While docked, all available features are shown on the screen, including the ability to make another call, transfer, conference, or look up a contact in the corporate directory. Each person the user is speaking with is displayed—name, picture, and phone number—at the top of the call.
When docked, users may have up to three concurrent calls and may start a new call by tapping the *New Line* button.

The *New Line* button is only present when users can actually add another line (that is, they have fewer than three currently active calls.)

When the user has three lines in use, the three callers are displayed at the top and the *New Line* button is removed since a user can only have a maximum of three concurrent calls.
Docked users can have up to three concurrent calls. Caller ID information for the calls are provided along with indicators to communicate when a call is active (green) or on hold (red.) A user can switch between calls by tapping the call they wish to make active.

The Call Log feature is one that is often not available on traditional office phones, but it is helpful and available in Mobile UC. Selecting Call Log displays a list of all incoming and outgoing calls, sorted by most recent. Whether the call was outgoing or incoming is displayed with text and icons in the list. Received calls show a blue down arrow, placed calls show a green up arrow, and missed calls show a twisted red arrow. Designers made it very easy to take action from this page too, rather than making it status only. Helping users to move forward easily is one sign of a great app design. The caller’s name and phone number are visible, as are the possible commands to Talk First or Transfer.
The Call Log displays all incoming and outgoing calls, plus the ability to select calls that were recently dialed or received, for transferring or other available functions.

Users may access their settings via the options menu. The following setting options are available: show dial pad when docked, authentication credentials, notify low signal, screen timeout, doc configuration. These follow Android conventions.
Settings are easily accessible from the options menu and follow android conventions. In addition, there is a setup wizard allowing customization the first time the user launches the application.

While most users probably won’t need help in this app, designers still provide a thorough description of features and how-tos. A few of the topics described in help include: Mobile UC features, placing calls while undocked, receiving calls while undocked, placing and receiving calls while undocked, and transferring calls. Each topic is presented in the list with an arrow to expand and collapse the write-up, so users may read the instructions in place.
An on-device Help system is available allowing the user to understand how to use application features. Each section is expandable.

Mobile UC marries the power of the traditional office phone with the flexibility of the mobile phone to make for a convenient, reliable, and simple experience for business users who also have a personal life.

**DESIGN PROCESS**

**From Idea to Execution**

The journey from idea to execution followed an unusual path for this application. It started with concept work and design, and then was marketed within the company to convince stakeholders to say, “Yes, this is something we’re interested in pursuing and developing!” The designs the team developed were instrumental in achieving that outcome.

“Usually,” says Linda Borghesani, Manager, Device Marketing, “people come to us [the design group] saying, ‘okay now [much further downstream] we need your support.’ But in this case, it actually started with the graphics and the idea of doing this dual persona application work on a cell phone.

The project originated from a kernel of an idea that was being circulated through Verizon Wireless and the process of evangelizing the idea to key IT decision-makers at various organizations drove the need to create design comps as a way to better illustrate the concept.
“Humphrey, our director, was giving a lot of PowerPoint presentations, mock-ups of what it might look like, to IT decision-makers at various organizations. So there was a need for graphics just to imagine the product and make the product a little bit more real,” says Borghesani.

That need was a lucky coincidence for the design group, which jumped on the opportunity to flesh out the concept. This gave the in-house team a rare chance to lead the charge and see to it that the product development eventually followed the vision articulated through the designs.

“It was just a happy coincidence that we were on the same team that would ultimately be responsible for developing the design so it just got started in our department,” says Borghesani. “The early technical architecture discussions weren’t done in isolation from the design work.”

Designing the application in-house put the design team in the driver’s seat and ensured that they remained central to the process along the way.

“This [project] was much better because we had control and we did a lot of things in-house,” she says. “I worked on another project where we had one company do the wireframes and another company do the comps. We were just third party to everything and the comps didn’t match the graphics nor match what we thought the product should be like. So it was the fact that we were part of the core design team [for this project] that has made the difference.”

Having the entire design team in-house allowed much more control of the product, and helped avoid delays or costs associated with making design changes that were due to technical constraints or other issues.

**Usability Methods**

The design process involved a number of iterative steps that were required to both refine the user interaction layer of the application and at the same time ensure that the application could deliver functionality despite (and constrained by) the challenges of the multiplicity of PBX environments. The team used several methods to refine the application design:

- **Iterating wireframes:** The team began with very detailed wireframes that demonstrated task flows exactly, as in, “When you press this button this should happen.” The wireframes even included all the error messages that the user would receive when completing the task. While the application functionality was effectively illustrated by these detailed wireframe designs, the wireframes had to continue to be iterated over time to accommodate changes that were required for technical constraints as they were discovered. And this was a constant flow of changes, across the designs. So, adding a design layer to this meant keeping precise track of which wireframe aligned with which comp.

  “Then what we did was we had comps to go along with all the wireframes,” says Borghesani. “So we labeled everything by screen number. So [for example] this is undocked screen 621. Then we would create a comp that would be marked ‘U621’ so that they’d know exactly which comp matched up with which part of the wireframes. That was a nice syncing process.”
• **Heuristic Review**: True task-based usability testing was not possible because the design was a moving target. The design had to be refined to adapt as the team uncovered new technology constraints. To add to the challenge, the team was working with a portable PBX system to simulate the PBX environment of a company phone system. To work around these limitations, the team chose to conduct mini-heuristic reviews against the proposed functionality as it developed.

• **Usability Testing**: Once the design was stable, but before it was implemented, the team conducted a usability test with eight people who represented potential customers (Android users.) It was a traditional talk out loud session where participants were asked to interact with a paper prototype with screen variations. The product idea scored very high with participants and some improvements to the UI resulted from the usability testing. The feedback helped the team remove screens and simplify some of the interactions. This gives the user more feedback when he needs it, especially in using the transfer and conference functions. The testing also contributed to the following:

  o **Improving the nomenclature for buttons and features.** Borghesani explains: “For several features, we subjectively asked which wording users preferred and in other cases we tested several concepts. For instance, we probed about the wording to add a second call. One option was *Add Call* which confused users [too similar to adding a conference call to the current conversation]. In this case *New Line* was preferred and is the wording we used.”

  “Participants preferred seeing the names and pictures, especially during conference calls to easily see who they are talking with and for call control. In some cases, more feedback was requested, such as more indication when a line was on hold and adding a confirmation of the transfer before being brought to the call log, which is the native approach in Android today after completing a call. We added a message to inform the user that the call was transferred successfully.”

  Participants preferred to return to the phone’s home screen when calls ended rather than the call log (which is how the native Android O/S works.)

  o **Reducing taps**: Borghesani explains, “We worked creating a design that uses the least number of user actions whenever possible, and we also tried to come up with as many defaults as we could. We really tried to keep the application as simple as we could because it is a very complex application to begin with.”
• **Flattening hierarchy wherever possible:** Sysko explains, “We tested five concepts using an interactive prototype on an Android device which varied in how we presented options to the users. The designs we tested included a Full Carousel, Split Carousel, Up/Down menu, Full Menu, and Partial menu, where less frequently-used options were hidden rather than being presented on the screen all the time. Users preferred to have all options visible to reduce extra steps rather than a step to open and close a tab to see additional options.” Another example is the **Transfer** function. “We could have put the **Talk, Transfer, or Send Voicemail** on secondary screen after dialing; or we could have users choose the action first and then choose the extension. We decided let’s put it all out there for the user. Let’s let them dial and then figure out what they want do with the call, but let’s put it on one screen rather than displaying another pop-up to tell them what to do.”

**Designing to the Least Common Denominator (in PBX systems)**

The team had to design for a variety of PBX systems, each with its own set of technical constraints. While some of the PBX systems that the application works on offer functionality that is above and beyond what’s included in the application, it was a simple reality that the team had to draw a line in the sand somewhere, and decide what to include and what to leave out.

The team settled on a manageable set of functions that were common across all the PBX back-ends. This meant that there were some edge cases where this one PBX could do something but it was decided to not include it in the app because it would have added to customer confusion and wouldn’t have worked all the time, across platforms.

This pantheon of PBX systems, even if designing to the least common denominator, presented challenges for the team, especially during testing. They had to accommodate multiple vendors on the PBX side, some with multiple versions of each PBX. So there were many variables and everything needed to be tested with each of those variations.

“I don’t think anyone realized there were so many difference between them or the versions of the PBXs,” says Borghesani. “Even with just Cisco there are so many differences based on the version the company is using.”

**Examples:**

Sysko explains some examples of technical considerations that had to be resolved along the way:

• **Multi-person conference calls:** “We could have provided functionality to accommodate conference calls with more than four people in mobile mode but it was determined that this would be overwhelming and not used by the vast majority of users.”
• **Shared lines:** The idea of shared lines is another example of something that is supported in certain PBX systems. “There’s the concept where an administrative assistant can access somebody else’s line. That functionality is supported when the phone is docked and the plan was to support that when undocked as well, but the problem is not every PBX implements it in the same way or even has that capability. So that is still being worked on and may be included in a future version.”

• **Mobile dial pad:** “There were originally designs intended to replace the mobile dial pad. So in dock mode we would be presenting a new design for the dialer. In mobile mode, we originally had designs for that. It was decided as a combination between usability simplification and technical simplification to make use of the existing stock dialing experiences that the phones had.”

• **Reducing lag time:** “We worked on how long should it take for a call to be placed because behind-the-scenes what it’s really doing is two-stage dialing. So we set a limit of six seconds to be able to do that second stage dialing - that the call should not take any longer (between the time you dial and the time it rings) than six seconds. That required working with the PBX for the two-stage dialing.

• **Adapting to switching behavior:** “There are differences with the switching behavior of mobile calls that are regional within the United States. So, as with any system you have to integrate with other systems and that requires technical adaptations in the application. This was especially true with the conferencing feature, for example.”
# Project Timeline

<table>
<thead>
<tr>
<th>2010 May</th>
<th>• Product ideation and internal championing</th>
</tr>
</thead>
</table>
| June | • Marketing Service Requirements (MSR) created to describe the product  
• Series of discussions with Product Marketing to determine the scope of the features in addition, we also had specialists: one team member was a PBX expert, one an Android development expert |
| July | • Draft graphics and design concepts to be applied to use cases. These helped the team evangelize the product and what the app was going to “do.” |
| Summer | • Vendor RFPs issued. The team needed two vendors to build the product: software and hardware |
| August /September | • The usability team developed draft comps to support use cases and product functionality. |
| October | • Usability testing on specific parts of the product that the team thought would be most confusing to users |
| November | • Began detailed wireframing to determine what functions to include in the application and to match the user cases and design specifications |
| December | • Baseline wireframes completed (docked flows, undocked flows, docking, undocking, settings, web portal and the setup wizard) |
| January 2011 | • Official product announcement at the Enterprise Connect Conference where the application was named as a semi-finalist. An alpha version of the application was demonstrated on a working device with a (portable) test PBX. The demo was tightly scripted so the team could control what they wanted to show and how it was demonstrated. We received great feedback and enthusiasm from Enterprise Connect attendees. |
## March
- Designed final graphics, layout guides and animation guides
- Tested alpha software to ensure all interface elements displayed properly
- Tested the alpha software to be sure that actions and results matched what was specified in the wireframes (when you press this button, this should happen)

Conducted a pixel perfect UI review. This review happened in parallel with the wireframe revisions and consisted of overlaying the wireframes over the design comps to ensure that everything was laid out properly and would display well in the final product

## April 2011
- Showed wireframes of the provisioning tool to IT decision makers at corporations that were interested in beta testing this product to get their feedback on the system administrator tool for setting up and configuring the service and revised the wireframes as needed.
- Signed up these IT stakeholders at outside corporations as beta testers

## April-August 2011
- Testing and refinement of the Beta software for user interface and technical issues

## August 2011
- Designed new layout guide for portrait mode devices
- Designed User Guide and Online Help System

## September 29, 2011
- Official Product Launch

## October 2011
- Chosen as Best Enterprise Product at CTIA

### LESSONS LEARNED
Verizon team members share a few lessons learned from the redesign effort:

- **Create good wireframes and design against them.** “The wireframes helped so much,” says Borghesani, “and we had comps to go along with all the wireframes, and each box in the wireframe had a comp to go with it that was numbered and labeled.”

- **Create a generous timeline.** The team learned the hard way that application design is very time-consuming and without a generous timeline things like testing get eliminated. ”It takes a lot more time than you think and I wish there had been time in the schedule for usability testing. We used ourselves internally and would walk around to each other so we could ask somebody, ‘What do you think about this?’” says Borghesani.
• **Budget for usability testing and budget time for the inevitable changes.** This rule is especially true, depending on the complexity of the wireframes and the technical architecture. The more complex (and detailed) the wireframes, the more time required to adequately test and revise them.

• **It's hard to test a moving target.** The constant flow of iterative changes to the wireframes meant there was no time for formal testing. "If only I had known how much time it was going to take, especially to do all the revisions and the testing. It was a very aggressive schedule to begin with and I think that was part of the problem. This was the first time something like this was done [in our group] so there were a lot of things that we had to change midstream, either adding or taking away functionality that needed to be redesigned into the application. And we needed to figure out where else that might affect things, which is why the wireframes were so pivotal," says Borghesani.

**TEAM**

The team responsible for the application is in a user experience design group that’s part of device marketing at Verizon Wireless. This placement of the team within the company shifted as the project proceeded, but despite this shift, the team remained with the project to its conclusion. Both a PBX expert and an Android expert were added to the team to address technical considerations of the project. And as mentioned before, the team hired both a hardware and software vendor to realize the vision of the Mobile UC application and docking station.

The design team roles were as follows:
## DESIGN TEAM RESPONSIBILITIES

<table>
<thead>
<tr>
<th>Role</th>
<th>Responsibilities</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Graphic Designers</strong></td>
<td>• Design graphics, animation and layout (with a good understanding of technical issues, PBX, and Android)</td>
</tr>
</tbody>
</table>
| **UI designers**            | • Develop settings for: set-up, provisioning, docking and undocked state, and docking/undocking  
                                 • Create application logic and wireframes  
                                 • Test against wireframes  
                                 • Write screen verbiage  
                                 • Coordinate with vendors  |
| **Third Party testing company** | • Conduct product testing based on use cases                                      |
| **Product Managers**        | • Guide feature and functionality specifications  
                                 • Set project schedule                                                           |
| **Third Party Developers**  | • Implement designs  
                                 • Test and bug fix  
                                 • Design docking station                                                          |

Pictured: Members of the application design team manning the company booth at Enterprise Connect 2011 where the product was first announced: Stanimir Dinov, Samantha Kalita, Kathleen Murphy, Linda Borghesani, Yeharn Hwang.
Ten Application Design Mistakes

Among the many submissions we judged that did not make our short winners’ list were some truly wonderful applications. Also in this group were some applications with some severe design and usability issues. Without mentioning any specific applications, organizations, or features, the list below summarizes some of the top reasons why some applications did not make the winners list.

1. **STEPS IN A TASK OR PROCESS PRESENTED OUT OF ORDER ON A PAGE OR SCREEN**

Good screen or page layout is difficult for many reasons, namely it can be thorny to prioritize tasks, and there is rarely enough space to present everything you want to the users. Of course practical constraints can make layout particularly testing, but what should not be a contributor to poor layout is ignoring the chronology of events. If the order of tasks can match the user’s workflow or a known, logical, or obvious process, then tasks should be presented in that order. Arrange forms chronologically as you would a food buffet: plates to put the food on, the food, the condiments and accoutrements, and lastly the utensils (if they are not on the table already.)

2. **MISSING SIMPLE PATHS**

When people finish a step they should not only have some sense of what happened, but also where they might need to go next. Applications are getting better and better at giving users feedback, but they are still sometimes missing that extra step of leading the user to the next most likely choices and giving guidance for those options. Some organizations know what a user’s most likely choice is, but purposely make another button more attractive--usually one that leads to more conversions. In these cases sales or clicks may look positive in the short term, but that is not way to forge a long-lasting relationship with customers.

Focusing only on what you (or the business) want people to do, not what they want to do is a major issue. At a conceptual level some apps have major issue: designers or business owners clearly just haven’t thought about why people would actually use it. At best these app designs turn out to be something nobody wants to use. At worst, some apps actually mislead users and hide critical information in order to get people to take some desired action. These short term gains end up hurting the brand.

Finally, another element of poor paths for the user comes in the form of lacking affordances, or not communicating when items are clickable, moveable, or in some way possible to take action on. This is especially true on touch interfaces where controls are very subtle, too small, don’t look clickable, or are just plain invisible.

3. **INVISIBLE CONTROLS AND CRYPTIC ICONS**

Toolbars, icon bars, and menus with icons can certainly help people who can recognize and or recall the icon. Locating an icon can enable people to perform tasks in a single click, which is especially great for common or repetitive tasks. But when an icon has issues, it may as well not exist.

Additionally, showing controls or icons for a few moments than taking them away is a trend that seems to appear on mobile apps more and more, and is creeping in to
desktop apps. This kind of UI is confusing and makes it difficult for users to find commands.

Some related issues we saw include:

- Showing controls for a few seconds then hiding them
- Tiny graphics
- Blurry graphics
- Recognizable but confusing elements, sometimes overly using metaphors
- Poor contrast between the icon and its background
- Multiple similar icons
- Pretty icons but that don’t help support the user’s core tasks

Attractive as they may be, icons need to support core tasks. Designers sometimes didn’t seem to spend enough time on task analysis to figure out what steps people need to take, and instead added attractive icons which resulted in a misallocation of screen real estate. Users are much better off with a plain and simple interface where the navigation closely matches the tasks, they don’t have to spend time looking at the navigation, and the content is the star of the show.

On a related note, we are also still seeing foreign elements such as expand/ collapse arrows just hanging in space, and buttons that don’t look like buttons (clickable.)

4. HIDEOUS FORMS

Alignment and balance go an incredibly long way in form design and usage. So why do designers still come up with forms that look like a game on Jenga? Users have trouble dealing with forms with right-aligned labels, and fields that are also right aligned, not aligned at all, or arranged in multiple sloppy columns.

And while stacked fields are typically better than multiple columns of fields, and sectioned steps are better than a potpourri of fields, cutting a form into too many screens, like mini-forms, makes the process take longer and can make the user feel constrained. Some of the non-winning apps shied away for long forms and broke their questions into too many steps.

5. UNCLEAR DIFFERENCES WHEN COMPARING CHOICES

Especially in an e-commerce environment people should be able to immediately tell the difference between items. But even with compare-type features sometimes it is difficult to decipher the difference between items, as images look the same and small-print in lists makes it difficult to find the differences.

6. INDISTINGUISHABLE CURRENT SELECTION STATUS

From accordions to scrollbars to buttons to preview elements, rampant in application design is the lack of feedback about the currently selected item. The issues almost

---

5 From Wikipedia: Jenga is a game of physical and mental skill created by Leslie Scott, and currently marketed by Parker Brothers, a division of Hasbro. During the game, players take turns to remove a block from a tower and balance it on top, creating a taller and increasingly unstable structure as the game progresses.
always surround the visual representation of the selected item. Something as subtle as the color of a tab not matching the related content below, or an expanded accordion menu heading which has a line below it that visually separates it from the sub-menu choices can completely baffle users.

7. BRANDING FEATURES OR MENU COMMANDS
Sure, sometimes your content or a link name may need to match a popular advertising promotion at an organization, but making up catchy phrases for features that are part of a task makes them confusing for people to find, understand, and use. Leave the branding to the advertising campaigns and name features based on tasks and common terms people use.

8. ONLY THE PROMISE OF CONTENT
Portals and dashboards are meant to deliver information as well as links to more details, but some apps only provide the latter. On a page with several sections displaying information, users should be able to decipher something, some answers, even from the small amount of visible content. Designers must use discretion and not cram the area full of too much content for the space allocated. But giving people generic terms that constitute only the promise of content is not only boring but dangerous, as users may not click something they need because the content on the front page is too vague.

9. VIEWS AND MODES THAT TRAP PEOPLE
A view is meant to help people see information in particular ways and eliminate the information not needed at the moment. A mode allows people to focus on one task only, by restricting access to unrelated features. Designers employing either often have good intentions, and both types of UI’s can work quite well. But when users are not sure what a mode is or its purpose, they can have a terrible time. Also, if a person is not aware that he filtered out information or is viewing a subset, again things can get pretty messy pretty quickly.

Ideally, make modes and views understandable, easy to tell when you are viewing content a particular way or in a mode for a certain task, and easy to get out of and back to another state. Even better, allow people to choose commands within a mode and make the mode forgiving by taking people out of the mode and to the area they actually need to be in. Be very cautious about hiding important features while in a certain mode or view.

10. BAD DIALOG
Sometimes a dialog box or page was obviously created as an afterthought and nobody truly considered what was being said to the user, or what he is allowed to reply.

If you tell a person something negative in a dialog box but present OK as the only button, then you deserve whatever comes next. It’s incredibly annoying to users to have to click OK when things are simply not okay. Consider using Close in those situations.
About the Authors

Kara Pernice is the Managing Director at Nielsen Norman Group and is an expert with more than 20 years of experience in user experience (UX) research and design. The Wall Street Journal called Pernice an "intranet guru." Since joining Nielsen Norman Group, Pernice has led many major intercontinental research studies and authored many research reports and guidelines, and is coauthor of the book Eyetracking Web Usability (2009.) She has lectured around the world on a wide range of topics, and her client work spans many businesses and industries. Before joining NN/g, Pernice gained invaluable experience about building and managing UX in an assortment of development environments and established several successful user-experience programs. A champion for usability, Pernice chaired the Usability Professionals' Association 2000 and 2001 conferences, and served as 2002 conference advisor. She holds an M.B.A. from Northeastern University and a B.A. from Simmons College.

Kathryn Whitenton is a User Experience Specialist with Nielsen Norman Group. She works with clients to evaluate the usability and information architecture of websites in a variety of industries including technology, telecommunications, and media, as well as corporate intranets. She has conducted usability research, eyetracking user research, and studies of users on mobile devices in the United States, Europe, Asia, and Australia. Her user studies have included general audiences as well as specific consumer types, business segments, children, and seniors. She teaches seminars on Information Architecture, Principles of Human–Computer Interaction, and Fundamental Guidelines for Web Usability.

Prior to joining NN/g, Whitenton worked as a Usability Engineer with the University of Washington Libraries, where she led user research and usability testing for the Libraries' website. She also worked on developing the user experience for agile web application startup companies, and as a Psychology researcher at the University of Texas at Austin, where she managed a clinical research study funded by the National Institute of Health. Whitenton holds Masters in Library and Information Science from the University of Washington, and a B.A. in Psychology and Plan II from the University of Texas at Austin.

Patty Caya (www.pattycaya.com) is a multi-platform journalist, editor, and award-winning digital media producer. In her business writing, she specializes in topics relating to usability, social media, and the business and technology of the web. For NN/g she has co-authored the Intranet Design Annuals since 2008 and she wrote the 3rd and 4th editions of the report on Intranet Portals. She is the author of the report, Enterprise 2.0: Social Software on Intranets.

For over a decade, Caya has split her time between journalism projects and web consulting. She is an experienced content strategist and information architect. She has consulted for many of Boston’s top interactive agencies, leading web and intranet development projects as well as usability testing, research, and design initiatives. Her client roster spans a wide range of industries and includes leading national brands alongside mission-driven non-profits. She has a BFA from New York University’s Tisch School of the Arts and has completed coursework in the User Experience Program at Bentley University.

Dr. Jakob Nielsen (www.useit.com) is a principal of Nielsen Norman Group. He is the founder of the “discount usability engineering” movement, which emphasizes fast
and efficient methods for improving the quality of user interfaces. Nielsen, noted as “the world’s leading expert on Web usability” by *U.S. News and World Report* and “the next best thing to a true time machine” by *USA Today*, is the author of the best-selling book *Designing Web Usability: The Practice of Simplicity* (1999), which has sold more than a quarter of a million copies in 22 languages. His other books include *Hypertext and Hypermedia* (1990), *Usability Engineering* (1993), *Usability Inspection Methods* (1994), *International User Interfaces* (1996), *Homepage Usability: 50 Websites Deconstructed* (2001), *Prioritizing Web Usability* (2006), and *Eyetracking Web Usability* (2009.) Nielsen’s Alertbox column on web usability has been published on the Internet since 1995 and currently has about 200,000 readers. From 1994 to 1998, Nielsen was a Sun Microsystems Distinguished Engineer. His previous affiliations include Bell Communications Research, the Technical University of Denmark, and the IBM User Interface Institute. He holds 79 US patents, mainly on ways of making the Internet easier to use.
Acknowledgements

Thank you to all of the organizations and designers who submitted their applications for this content. Special thanks to the winners in this report, for their wonderful apps and for sharing them with us.

Also thank you to Susan Pernice for her assistance with this report.