

How We Prototype

[Web Smart Newsletter](#) by Eric Holter, December 2007

How We Prototype

We kicked off our 2007 newsletters with "Why We Prototype." It was a short re-telling of how we stumbled upon our grayscreen prototyping process. The longer version is recounted in our book, [Client vs. Developer Wars](#).

We'll finish the year off with the subject of prototyping again--but this time we'll talk about how we prototype. We've also produced a ten minute video demonstration of our prototyping process.

Our January newsletter was the story of how we discovered our grayscreen prototyping process. We learned the hard way that paper-based sitemaps and wireframes are not effective for communicating the subtleties of a website's **structure, content** and **functionality**. Paper leaves too much up to the imagination. Websites are non-linear, complex objects. They can't be reduced to linear, simplified, paper documents. And what's worse, paper-bound specifications leave both clients and developers under the illusion that they have effectively communicated. In reality, all they've done is produce a document whose meaning can later be argued about when the final results fail to match un-communicated expectations. Paper allows assumptions and misunderstandings to creep in. We lived this web development horror show from 1995 through 2000. But then we discovered grayscreen prototyping.

A grayscreen prototype is very much like a detailed paper-based wireframe, except that a grayscreen prototype is not presented on paper, it's presented online, in a browser, where it can be clicked and experienced **as a website**. This subtle difference captures insights and understanding in ways that paper documents can't produce.

My previous newsletter, and our book, go into the details about why grayscreen prototypes make such a difference. So I won't cover that ground again. Instead I'll focus on how we go about building prototypes, our process, our thinking, and our execution.

F.A.Q.

Let me answer a frequently asked question at the outset. Many web developers and designers follow this newsletter. So whenever I talk about prototyping they often ask what software we use to build them. We built a custom prototyping system using the NewfangledCMS, for our internal use. However, the principles for grayscreen prototyping are in no way dependent on our software. You can use a WYSIWYG tool like Dreamweaver, or any other CMS to build a grayscreen prototype. I have been told there are a couple software developers that have made tools for prototyping. [Jumpchart](#) and [Intuitect](#) are the two I've heard of. I have not used them but if anyone has, please use the comments below to let us know about your experience. Likewise, if you know of other systems let us all know via the comments.

The goal of a grayscreen prototype is to thoroughly represent the **structure, content** and **functionality** of a website.

Prototyping Website Structure

STRUCTURE:

Starting a Prototype

Every web development project starts with different initial input. Sometimes an existing website is to be redesigned. Sometimes there's a detailed Request For Proposal, and sometimes there's a **sitemap**. Sitemaps are my enemy. They're such illusionists. They're so limited in how they define categories and relationships between pages. They fall far short of communicating the details of the pages themselves.

Building a website based on a sitemap is insanity.

I'm also uneasy with notorious wireframes. They're slightly more sophisticated than sitemaps--offering a lot more detail. The problem is, they're still presented on paper. And paper just doesn't work. In fact, the detailed nature of a wireframe, I believe, contributes to illusions of communication even more than sitemaps. That's because they look so thorough and detailed. The detail makes those who review them feel like the site has been thought out. And they probably have been thought out--by the developers who make them. But as a vehicle for communicating this thinking, they fail miserably.

So when we begin a web project, whether we're starting from an existing website, sitemap or wireframe, our very first step is to put all the input into a clickable, web-based, grayscreen prototype. Then we burn the sitemap or wireframe.

Sorting Out The Puzzle Pieces

Every website is a puzzle. Some are kid-simple 100-piece varieties, others are 5,000-piece brain busters. Regardless of how complex the puzzle, first you need to turn over all the pieces and start looking for edges.

Defining the "edges" of a website consists of working out site categories, grouping and labeling subpages, and differentiating main navigation from universal navigation (such as home, search, contact and sitemap, sometimes also referred to as "site utilities").

Website Categorization and Navigation

Books on information architecture suggest a maximum of seven main site categories. That's a good rule of thumb. Some main navigation categories, like "About Us" are common to most websites. They typically contain pages like History, Mission, People, and Employment. News, Events, and Calendar are also common site sections. But sometimes grouping and naming a site's main categories is trickier. Consider a software company whose main product is called "CyberTrac." A software product section would normally include pages for product downloads, support, developer resources, demos, and documentation. But should we group these pages under a menu labeled "Products"? Or do we name the main navigation tab after the product and label it "CyberTrac"? Getting the product name into the main navigation could be good for branding, but would visitors understand that "CyberTrac" is where to find all their product information?

There are no absolutes. Factors (like the number of products) bear weight in deciding how to group and name such a category. The name of the product is another factor. In this case "CyberTrac" kind of sounds like a software package so it might work as a label, but if the product were called "Centra," it could sound too abstract to function as an effective main navigation label.

Navigation Flow

Navigation flow is very important. We keep an eye out for navigation items that click to other areas of the site that are not actually under the tab where it was listed. For example, "Downloads" is a popular main navigation tab for software companies but it also makes sense to include a link for product downloads under the "Products" tab. Since there is just one Download page, the page listed under Products could redirect to the main "Downloads" tab. But this is always disorienting to site users. Cross-linking is a common problem and usually betrays a failure in overall site organization and categorization.

Navigation Balance

During our initial prototyping we evaluate the overall navigation balance. If one section has ten drop-down choices, while the others have only one or two, the navigation is imbalanced. This also usually means there is a problem in overall categorization.

Steve Krug's book [*Don't Make Me Think*](#), and [*Information Architecture*](#), by Peter Morville and Louis Rosenfeld are helpful resources for learning more about information architecture and usability design. The website BoxesAndArrows.com is another good resource.

Information architecture can be expressed using a paper sitemap or wireframe. But in a grayscreen prototype clients and developers get a feel for how their choices work in a browser. Cross linking problems, for example, are felt much more acutely with the clicking experience than they are when navigation is represented with lines between boxes on paper.

Prototyping Website Content

CONTENT:

Moving From Edges to Details

Working out the structure, categorization, grouping and labeling is actually the easy part of website information architecture. Representing content details on each page is much harder. Of course the final site copy is never ready at such an early planning stage, so we use Greeking copy (Lorem ipsum dolar) in place of final copy and blank images as place holders for final images. However, we do represent the intent, length, and detail of all content expected for each page. Some pages may just have a sentence or two, others may have multiple paragraphs with images, captions, and tables.

It is extremely important to represent content in detail, albeit without the real words and images, even on simple pages. Take a "jump off" or what we sometimes call a "table of contents" style page for example. Let me explain why a "table of contents" style page is sometime necessary. Some main site tabs exist primary as containers for the subpages under them. A travel site, for example, might have a section called "Destinations." Its subpages could be broken up by type: Island Getaways, Ski Resorts, Cruises, and City Tours. Each of these subsections would have considerable content, but the "Destinations" page itself might not have much else to say other than "Here are the pages that talk about our destinations."

The content on these kinds of pages is often overlooked in sitemaps and wireframes. But since the page exists, it will have to have something on it. So when we prototype, we make our best guess at how such pages should be formatted. We'd probably show a few lines of introductory "lorem ipsum" text and then list each destination as a link with short description and a "more" link. Maybe we'd throw a thumbnail image next to each item too.

By detailing the content on this simple page, the client and project manager will stop to consider it. Had the prototype merely displayed a few links that read "subsection one," "subsection two," and "subsection three," the client and the project manager would fly right past the page during review.

But with detail to respond to, their review and discussion can lead to important discoveries that might save time and frustration. Here are a couple of ways such a conversation could go.

Conversation Possibility #1:

Project Manager: "We've represented the destinations page with a few lines of intro copy and links to the four subsections, each one having a short description and thumbnail image. We call these kinds of pages "table of contents" pages because they don't serve much purpose other than directing people to the content that lives under them. In most cases, because of the way drop down menus work, most users won't actually go to these pages. They click through directly from the menu bar to the subpages. Do you feel this format will work for you?"

Client: "Well, actually we want people to see this page. Our packages have seasonal trends so we want to promote certain ones at various times of the year. We thought the main destinations page would be a good place to do this." In this scenario we might discuss ways of working a destinations overview page as the top option in the drop down menu so people would be less likely to miss their important destination overview content. We'd go back and add more detail into the prototype to represent how they might promote their destinations on this page. This conversation might also lead to ideas for including promotions on the home page, or functionality requirements like date-based control over when promotions display.

Conversation Possibility #2:

Project Manager: "We've represented the destinations page with a few lines of intro copy and links to the four subsections, each one having a short description and thumbnail image. We call these kinds of pages "table of contents" pages because they don't serve much purpose other than directing people to the content that lives under them. In most cases, because of the way drop down menus work, most users won't actually go to these pages. They click through directly from the menu bar to the subpages. Do you feel this format will work for you?"

Client: Well, if this page is not usually seen, I think we can ditch the thumbnails. We don't want to have to pick images for a rarely viewed page. And could we just pull the first few lines from the subpage's main content as the abstract so we don't have to write text for all of these table of content's pages? In this case we would actually simplify the prototype and minimize the amount of content the client would ultimately need to work into the final site. We would add a programmer note to remind the developer, when it comes time for coding, to display an abstract for each destination from the destination subpages.

Devilish Details

I've used a simple page as example of how detailing content on every page can either help refine or simplify the site. These details need to be considered sooner or later. Otherwise the developer might not build the pages right, or the client might get hung up in content development. When these elements are glossed over they don't go away; they just surface much later. As a result many pages get re-coded, usually under the pressure of a looming go-live date.

And if simple pages have potential for illusions of communication, how much more complex aspects of e-commerce, account management systems, and customized applications?

Prototyping Website Functionality

FUNCTIONALITY:

Showing How it Will Work

Working the edges of a site's structure and diving into the details of its content is essential for communicating effectively. But websites are often more sophisticated than that. E-commerce, personalization, advanced applications, integration with third party services, community features, complex forms, and surveys add intricacy to today's full featured websites. Therefore, grayscreen prototypes must include functionality mock-ups and technical notes that describe exactly how custom web applications, or e-commerce implementations, will work. If a normal web page can contain many subtle considerations, a web application or transactional site has far more. Click through screens that mock-up typical interactions helps to refine functionality, and reveal areas where important questions need to be answered before development.

So when building a prototype we mock up everything from simple comment forms and log-in procedures to more complex e-commerce and account management screens.

THE PROCESS:

Prototyping Procedure

Building a thorough grayscreen prototype requires several rounds of review. Each successive round refines the details and brings the prototype into focus. One of the advantages of building web-based prototypes is that many people can review them online and chime in on the work in progress. We like to include as many people in the prototyping process as we can.

The project manager and the information architect here at Newfangled take in all the initial information and begin turning over all the puzzle pieces. Once they've worked through the prototype, we do a broader internal review. I personally review our prototypes at this stage. We also invite review from a couple of other internal resources, who have no familiarity with the client or the prototype. Fresh eyes bring a level of insight that clients and the prototyping team don't see because they're so close to all the pieces. After our internal review, we invite the client to weigh in on the first round of prototyping work.

While we make our best educated guesses during our initial prototyping, we always need the client's feedback to move forward. This stage is not without tension. One of the classic mistakes in information architecture is not being able to see past internal ways of thinking about organization. Clients can get too caught up on their own internal jargon, or even get derailed by internal politics. That's one of the benefits of hiring an outside firm. You get an objective perspective. But reorienting your mind--when it's used to thinking one way about things--and fighting internal battles, can be daunting. Being able to show a working web-based prototype to stakeholders helps demonstrate the benefits of one solution over another.

Once a prototype gets close to completion, we bring in the designers and developers. They will inherit the prototype as their blueprint for design and coding, so their understanding of its features, and their insight on its formulation are extremely valuable. We make sure that there is still time so that whatever insight designers and developers have can be integrated and evaluated by the team before the prototype is finally approved. Everyone's background contributes to their perspective on a website. Marketers think one way; programmers think another; and designers have their own refined perspective. Ultimately, it's the site's users that we're trying to serve, and they're never there during planning. But by getting as many different perspectives on the prototype as we can, we will start to approximate the average user. (Of course there is no such thing as an "average user." Marketers, developers and designers are all site users too, so a variety of insight

from different perspectives usually results in an effective site design.)

Comments

One of the features we've built into our prototyping system is a commenting area at the bottom of each prototype page. Clients can note their ideas, change requests, and concerns, which become a part of the prototype.

A Note About Gray

One of the things we learned about how to prototype is how important it is to keep the prototype generic. Adding any design elements at all, including subtle color suggestions, begins to beg the question "Is this how the site will look?" We've actually worked very hard at making our prototypes look professional and clean, without looking "designed." If anything, we want them to look ugly, so that our clients don't get distracted by visual design.

"A Stitch in Time..."

We spend as much time as we need to thoroughly detail a site during the prototyping phase. We never watch the clock. After all, it's far easier to integrate changes, refinements, and ideas into a prototype than it is after a site's been designed or coded. There are many other benefits to grayscreen prototyping that we've observed. We invite you to read or download our book [**Client vs. Developer Wars**](#) to read more about how and why we prototype and the amazing difference it makes.