

Audiences & Artifacts

by Nathan Curtis

Years ago, Jakob Nielsen introduced us to the notion that users don't read on the web (www.useit.com/alertbox/9710a.html). They still don't.

At EightShapes, we spend much time talking with clients about their user experience documentation practices and the role that artifacts play in the design process. What did we find?

People don't read deliverables either.

The truth is that consumers of your documentation scan deliverables. They'll refer to deliverables; they'll look for those nuggets of information they can use to complete their own work. Bottom line? They don't read deliverables, but most of the time they try to use them.

Therefore, we're constantly inspired, driven, even forced to improve our design communications to be more effectively usable and efficiently produced, much like how we approach user-centered design itself. Where should we start? This article considers the artifacts we produce, who we produce them for and some techniques we can perform to improve our design communications.

The Audiences and Artifacts

In user experience design and, more specifically, activities for information architecture and interaction design, we communicate design solutions to audiences that include executives, product managers, site strategists, visual designers, design technologists (HTML/CSS/JS), developers, quality assurance specialists (QA), copywriters, publishers and – don't forget – ourselves. Artifacts include strategy documents, concept models, maps, flows, storyboards, wireframes (basic to annotated to detailed specs), mockups, style guides and more. In fact, this artifact list is just the tip of the iceberg (for more details on key artifacts for information architects, refer to Dan Brown's "Communicating Design").

Nathan Curtis is a founder and principal at EightShapes, LLC, a UX consulting firm based in Washington, DC, and blogs on design and documentation at www.nathancurtis.com.

Information Architecture

At a high level, different audiences respond to and inform different artifacts in different ways. Executives and product managers talk business objectives and high-level requirements and may be more responsive to strategic discussions, storyboards and (unfortunately given the effort to create them) high-grade mockups that present the experience concretely. On the other hand, visual designers, developers and QA often prefer rich, annotated wireframes and specifications to complete their own work in sync with the detailed behaviors and structures we create.

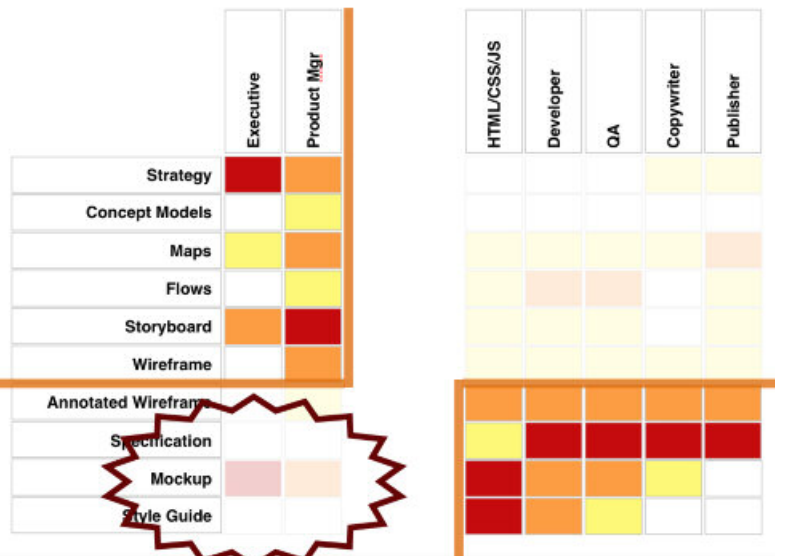
The following diagram represents an interpretive heat map of document relevance across audiences and artifacts based on our research thus far. While not objectively quantitative, the table enables us to think and discuss the rationale for utilizing specific artifacts to communicate in specific ways to specific audiences (Figure 1).

FIGURE 1. Audiences and artifacts

	Executive	Product Mgr	Site Strategist	Visual Designer	HTML/CSS/JS	Developer	QA	Copywriter	Publisher
Strategy	Red	Orange	Orange	Yellow				Yellow	Yellow
Concept Models		Yellow	Yellow						
Maps	Yellow	Orange	Orange	Orange		Yellow	Yellow		Orange
Flows		Yellow	Orange	Yellow		Orange	Orange		Yellow
Storyboard	Orange	Red	Orange						
Wireframe		Orange	Orange	Red	Yellow			Yellow	Yellow
Annotated Wireframe		Yellow	Orange	Orange	Orange	Orange	Orange	Orange	Orange
Specification			Orange	Yellow	Yellow	Red	Red	Red	Red
Mockup	Red	Orange	Yellow	Yellow	Red	Orange	Orange	Yellow	
Style Guide			Yellow	Yellow	Red	Orange	Yellow		

CURTIS, continued

FIGURE 2. The threshold



Here, a threshold emerges: a level of detail beyond which some audiences may not respond, but which other audiences may demand. Commonly, this threshold begins with anything beyond light annotations on wireframes. With such a threshold in mind, we could consider communicating via different, targeted deliverables for two (or more) segments of our audience base (Figure 2). Sure, the mockup problem persists, but by educating our stakeholders on the value and role of different artifacts, we can minimize the impacts of such issues.

Clearly, not all artifacts are for everyone. In numerous surveys across audiences in different organizations, there are clear distinctions between the value placed on different artifacts by information architects and the population to which they communicate. Sure, most often both IAs and their audiences find significant value in wireframes. However, what about other artifacts? IAs use concept models as a mechanism to clarify objects and relationships within a design solution; however, most audiences require non-trivial education about the role and value of concept models and how they differ from other familiar artifacts like site maps. Even with that, the perceived value of concept models is much higher to IAs than other individuals.

On the flip side, consider change histories, which are records of how a deliverable evolves. Such a history can be very valuable to consumers like developers, visual designers and QA. Trouble is, IAs don't place as much value on them since all that information is in their heads, so why record it? Therefore, change histories most often end up being poorly recorded, with pithy comments like "Wireframes updated" if anything at all, leaving others in the lurch and dependent on verbal communications.

And site maps? Surprisingly, consumers who are not IAs often described to us a significant desire to understand the site structure via an artifact akin to a site map (if not visual, still an organized and coded list or spreadsheet). With that, other teams can plan and execute their work using a common language and organization. Most admit looking to the information architect to set that tone. Sure, some of us still produce site maps, but they are seen less often these days and not structured enough with categories, numbered references and other techniques to serve as a foundation for remaining project work across disciplines.

Deliverable Life Cycle

As a deliverable evolves, it typically moves through a series of stages before converging on a final version with a stable and mature structure. Ideally, artifacts are instantiated from some template, with starter content sourced from other existing deliverables. That said, over time the deliverable matures along with the design itself from conceptual presentations and strategies to additional variations to more and more detailed annotations.

Through this life cycle, many factors can disrupt or shift the direction of the artifact's growth. Design iteration can cause large chunks of the design to change – along with any documentation of those chunks already produced. Descoping can result in areas of a design being completely removed from a project. Should this documentation be thrown away completely, just removed from this deliverable or sustained with descoped markers? Executive input can shift significant design progress in completely new directions. Should the documentation have been exposed to such an audience, and what do we do with it now?

Also, the artifact's audience base commonly grows over time from a

CURTIS, continued

narrow team of a few individuals to across an organization. Is your document maturing to address those audiences? And depending on your workflow and organizational structure, does your artifact clarify the design enough so that those audience members you communicate with directly can represent the work to the individuals you actually never meet or talk to?

Such considerations are critical for the composition and maintenance of a successful design artifact. What can we do to address this need? The remainder of this article touches on specific topics and suggestions to stimulate your creativity and improve your design communications.

Be Structured

Let's start with the document's foundation (Figure 3). Many design meetings begin with discussions like "Ok, is everyone ready? Is everyone looking at eCommerce.Wireframes.1.9.pdf?" Creating a platform for structured discussion around an artifact is therefore critical. Document metadata gives us the opportunity to do that, resulting in precise and unambiguous discussions based on specific and easy to understand structures and references within our design documents.

Document structure starts with information suitable for presentation across all pages via header, including document title, organization (typically via logo), author, author contact information (usually email address), version number, publication date, page number and overall number of pages. Interestingly, our deliverable review activities across many clients reveal that very few documents produced

FIGURE 3. Document foundation



include standardized page headers including all these attributes. The result? Confused consumers leaf through printed or electronic documents without a sense of place, suffering without key signposts and information to guide them through the basics. One would think we're beyond such trivial inadequacies of our design artifacts, but research proves otherwise.

With standard wayfinding resolved, designers can begin to layer additional structures into an artifact. Depending on your tool, produce a table of contents and other reference lists in an automated way using styles or other techniques so that you can set the tone at the outset of a review and create a baseline for future usage. Utilize numbering systems to organize items within chapters, properties of a page or component, and other types of requirements, ideally through automated paragraph styles and systems. And record the evolution of a design over time via change histories, both summarized via generalized lists or tables as well as integrated into specific annotations via colored highlights and additional content. Other opportunities to add context to your artifact include lists of reviewers, tracking open questions about the design, a glossary for project-specific concepts and/or jargon and more.

Be Focused

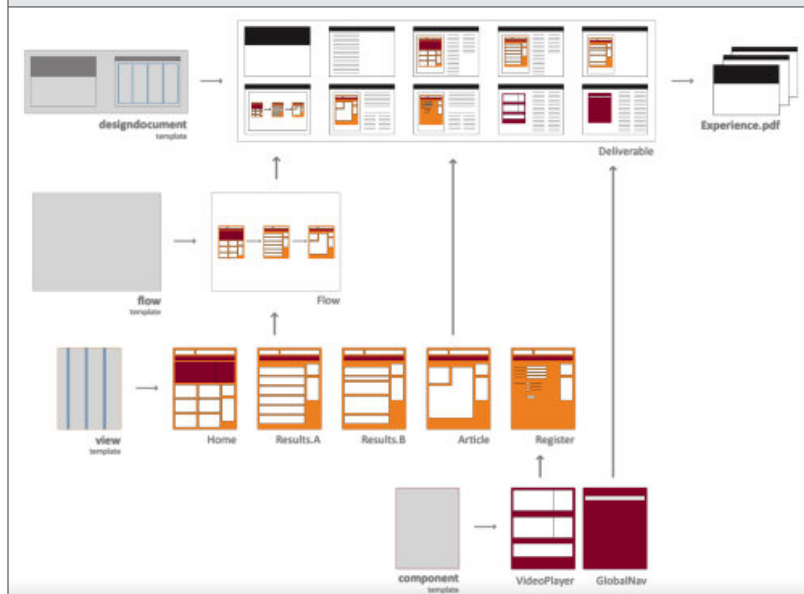
One implicit weakness of most deliverables is that they try to be everything to everyone. Designers may describe problem statements, personas, business objectives, data formats, editorial guidelines and behavior specs all on the same page. Even worse, such descriptions may end up intermingled without a coherent structure that facilitates scanning, retrieval and understanding. Sure, you've included the content that each audience needs, but each consumer ends up seeing a jumbled mess from which they must search and identify nuggets that are important to them.

Our research shows that, more often than not, each audience selfishly wants to see what they care about without so much other clutter. The eyes of product owners may glaze at the sight of detailed states and variations. Quality assurance may feign interest in personas, but when probed further, admit such documentation contributes little to structuring their test case.

To address this factor, consider moving away from a "one-size-fits-all" composition strategy and create 2+ artifacts tailored to specific audiences.

CURTIS, continued

FIGURE 4. Modularity



If you are authoring artifacts from a single file (such as produced by Visio), you can utilize layers to show and hide various annotation attributes when printing the document or exporting to PDF to share with others.

On the other hand, you could consider modularizing your artifacts into numerous, reusable chunks of artwork and annotation. Using linked files in a product like Adobe InDesign CS3, you can reuse wireframe or mockup page and component artwork repeatedly in multiple deliverables and flows while managing changes and evolution from the same files. That way you can create one deliverable PDF for product owners and executives that covers the highlights and strategy, while producing a design specification PDF with more variations and details for developers and QA, but both from the same linked artwork stored modularly apart from either deliverable.

Be Predictable

Since deliverable audiences will learn and adapt to your document structure within and across projects, it is important to create learnable,

consistent page structures. One technique we've employed over the past year is the application of page patterns to pages within our deliverables. In short, a page pattern is a combination of predefined shapes and text placeholders in a layout that enables a design to place artwork and compose annotations. Such page patterns are akin to our user interaction design patterns, in that they provide a consistent yet flexible solution starting point for common deliverable problems such as varied techniques for annotating a wireframe, flow, storyboard or other typical presentation.

For example, suppose you were introducing a wireframe and chunking the page into segments that you intend to annotate later in the document. Then you could utilize the "Wireframe (Intro & Chunked)" page pattern that includes (a) a frame for the wireframe artwork, (b) a descriptive intro paragraph that you can type over, (c) a list of components where you can name the chunks and (d) markers so that you can connect that list with highlighted areas on the wireframe artwork.

In effect, these objects and placeholder text areas prevent you from having to reinvent the wheel every time you want to introduce a wireframe and also result in a pattern your audience can quickly recognize and understand.

Better yet, you can organize these page patterns into a library and share them with teammates. Such patterns can run the gamut of your deliverable landscape, from capturing requirements and conveying project strategy to preparing research reports like a competitive analysis to annotating wireframes and visual design artifacts. Once organized, you can create a foundation where resource can easily retrieve and apply those patterns in their own deliverables.

Be Prepared

The response to the question: "How do you get started on a new deliverable?" is almost always: "I open a previous deliverable, erase irrelevant content and jump right in." As designers, our instinct is to solve a design problem, not think about how we're going to document that solution. We start by throwing visuals and annotations into a document without considering the long-range goals or life cycle of the document over weeks or even months.

CURTIS, continued

Document “recipes” (Figure 5) afford us the opportunities to consider and plan a deliverable from the outset. By talking about documents in terms of recipes, designers can discuss a document’s purpose, audience and process of production. Such a process addresses what gets added – the ingredients (through the language of page patterns) – and how those ingredients may change over time.

You can concoct a recipe via a casual five-minute conversation or a deeper, more formalized planning session that includes numerous document consumers. The discussion enables individuals or groups to clarify the direction of a deliverable through exercises akin to outlining, chunking out segments and even assigning responsibilities for composition across teammates. And with a page pattern library as a baseline for common understanding and vocabulary, designers can quickly describe the tone, level of detail and breadth of coverage across a document.

Additionally, larger teams can use recipes to standardize expectations for documents produced frequently by numerous resources. For example, a team can structure a competitive analysis to have an executive summary, table of

contents and then chapters for approach and strategy, annotated examples and findings and recommendations. Such a recipe could be described in a single page with an introduction, document outline (via page patterns) and process in which the content is authored. Later, the designer can use the recipe to quickly produce an effective design communication that meets team expectations.

Be Investigative

When designing, don’t you always try to include the voice of your consumer? You can apply user research techniques to your deliverable design too.

Recruit the participants, write the test plan, reserve the testing facility and... well... perhaps not. We need not put our developers and quality assurance professionals in a usability lab to determine if our artifacts are usable.

But we can apply simple and quick techniques to get valuable feedback on our design communications. Without a doubt, my greatest improvements in design communications have come from asking questions of my artifact’s consumers. Here are some ideas for getting that feedback:

- **Plan** the deliverable collaboratively with your audience from the outset, insuring what you’ll produce meets their needs
- Sit down beside artifact consumers and **test** the deliverable informally. Learn if your audience can use it without you reading it to them, line by line. For example, if you’ve annotated a behavior, ask them if they can find, understand and explain the requirement without leading their answers.
- **Iterate** and improve your deliverable based on feedback during design reviews, proving to your audience that you are listening.

Be Mechanized

For much of our user experience design work, our design concepts arise within the broader context of a defined or burgeoning design system of grids, styles and – ultimately – broad collections of structured components that can be reused across pages, sections and projects. Each component is a page’s chunk, with variations accommodating different states and scenarios and intentionally allowing specific page location(s), usage criteria and design objectives.

FIGURE 5. Document recipe



CURTIS, continued

Large user experience design teams go to varying lengths to codify a component library for extensive reuse across user experience resources and throughout the development process. Therefore, our documentation systems, such as templates and libraries built within a tool like Adobe InDesign or Microsoft Visio, will include libraries of predefined, classified component artwork. Such components can be applied to create wireframes a million times over without ever creating the same wireframe twice, all within that recognized system.

As wireframes become increasingly formalized via components, they are increasingly congruent with the codebase of HTML, JS and CSS that design technologists create for efficiencies in their own work. However, with improved mappings between disciplines from wireframes to code, new expectations begin to arise: How synchronized are the two representations of the design system as each matures in parallel? What is the process to co-manage the libraries? At what cost are the dual artifacts maintained? Such issues may present challenges, but benefits have proven to outweigh the drawbacks.

In fact, mechanization has even extended to experimenting with automated wireframe creation of frequently used page-level starting points by combining (a) a formalized wireframe template, (b) XML page definitions

based on components and (c) XML description of a design system's page grid. Relatively straightforward scripting enables quick, efficient production of starting points so that information architects need not begin by staring at a blank canvas each time. With components in hand, there are numerous directions we can explore to make our process more efficient, consistent and scalable to a large team.

Conclusion

Our hope is that information architects continue to polish their design communications just as they do the designs they create and strive for artifacts that are more structured, focused, predictable, prepared, usable and mechanized. By formalizing our approach to common documentation problems, we create more opportunity (and time) to creatively address the interesting design problems we face every day. Ultimately, such polish and professional communication lends credibility and influence to our discipline so we can contribute meaningfully to a positive user experience.

This material was presented at the ASIS&T IA Summit 2008 in Miami, FL, and associated slides from that presentation can be downloaded from nathancurtis.com. ■