

(Not) Designing for Ourselves

A Case Study from the U.S. Web Design Standards

Visit the site standards.usa.gov

Download our slides github.com/18F/wds-ia-summit-presentation

18F

Colin: Hi, we're [who we are] from 18F, [intro 18F], and we're going to talk to you about [what we're going to talk about.]

1/ Intro

18F

Colin:

Imagine Joanne, a young Veteran who is looking to make use of her VA GI Bill Benefits and apply for Federal Student Loans to attend college.

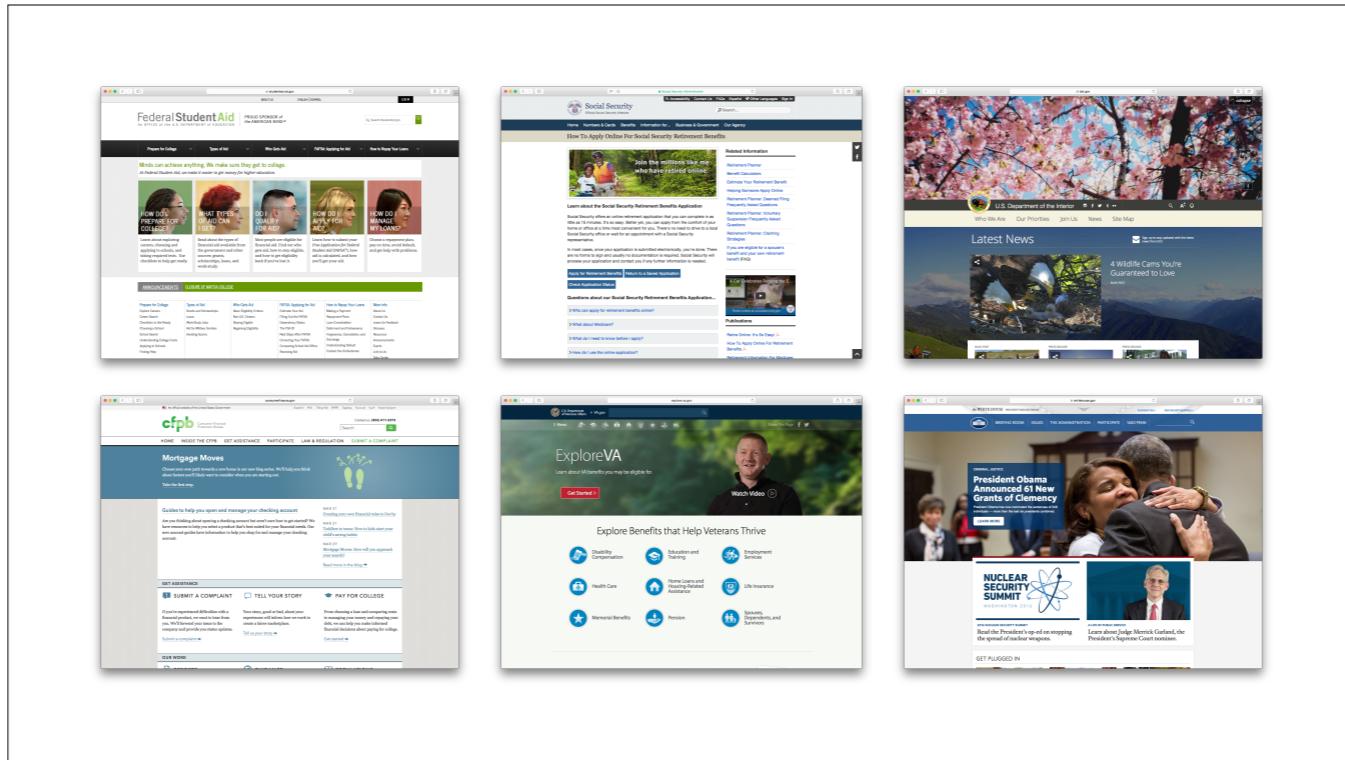
To apply for Federal programs that allow Joanne to afford college, she has to navigate multiple agency's websites to understand the requirements and application process. In her initial search, she is bombarded with information. She finds dozens of government websites which all seem related to what she's looking for – studentloans.gov, studentaid.ed.gov, direct.ed.gov, fafsa.ed.gov, benefits.va.gov/gibill, explore.va.gov, consumerfinance.gov/paying-for-college ... the list goes on.

After wading through a flood of words at every turn, Joanne is confused. Are these programs related to each other? Are they even all a part of the federal government? Are some of them a scam? Everything looks different. Some of the sites seems like they haven't been updated in 15 years, and just seem kind of sketchy to her. She tries to make sense of it all, bookmarking the sites to read on the bus on her commute to work, but half of the sites are impossible to really use on her phone.

Carolyn:

If we want to make Joanne's experience better, we also have to think about the people who build federal websites. Now, everyone on our team happens to be people who build federal websites, but not everyone who does so works in a large, digitally focused team.

So let's imagine Roy and Jennifer. Roy and Jennifer are real people who work right here in Atlanta – they are a two-person web design and development team for the entire Southeast Region of the U.S. Fish and Wildlife Service. Among other responsibilities, they manage their region's website of over 4500 pages. Content is contributed to these pages by subject matter experts, many of whom are biologists who are literally working in the field.



Carolyn:

In terms of building digital tools, a lot of progress is happening in the federal government. But most of what is happening is happening in silos – within individual agencies or even within separate teams in those agencies. So while any individual agency can innovate, it's hard to create consistency across sites.

Colin:

Some agencies are developing their own pattern libraries, but there was no unified and shared library for the entire government.

Why not use [insert existing UI library]?

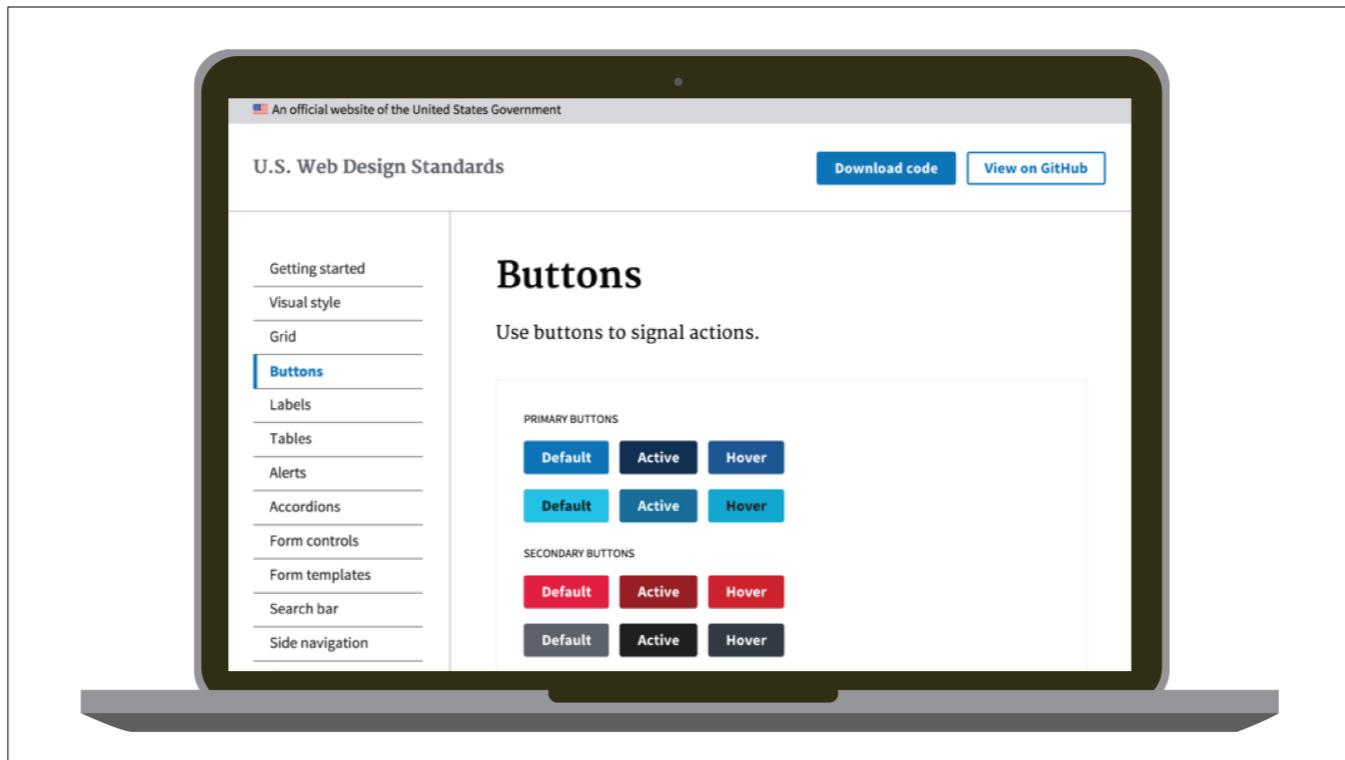
Colin:

Why not use an existing, free, open source pattern library?

Accessibility is often an afterthought in these frameworks (like Bootstrap) and they require refactoring both the visual design and code. That adds extra cycles of work or failing to deliver fully accessible digital services.

Many federal agency teams were looking for something to help them quickly put together polished, credible prototypes to easily test with end users.

Not all teams can afford the time and resources to build and maintain their own front-end design systems.



Carolyn: That was the impetus behind creating the Draft U.S. Web Design Standards.

The Draft U.S. Web Design Standards are open source set of UI components and a visual style guide to create consistent and positive user experiences across U.S. government websites.

Essentially the Standards are a Pattern Library of the United States, or as I like to say, PLOTUS ...

Why should you care?

Colin:

One of the most interesting challenges in this project is that agencies aren't required to use the Standards at this point. So we need to make the Standards a tool that people want to use and can use. We want to "make the best thing the easiest thing."

Why should you care?

When designing the standards, we're designing for multiple agencies that have different needs, cultures, vocabularies, and which serve different users. Your large organization might look a lot like this too.

Maybe your company can make your pattern library mandatory, but wouldn't it be a lot easier if employees also wanted to use it? If they believed in its purpose and mission?

We want to share what we learned through user research about building a pattern library that people across a diverse organization *want* to use.

2/ Finding things

18F

Carolyn:

There are two primary actions on our pattern library. The first one is finding things, and the second is using the thing. So finding things is all about navigation, and we tried a lot of different solutions.

Government Wide Pattern Library

Composer

Accordions

Use

This is the usage center for accordions.

Footers

Visual Style
Layout System
Elements
Components
Accordions
Footers
Forms
Headers
Navigation
Search Bar
Search Results

What we tried:

Carolyn:

A lot of existing libraries categorize the patterns they offer into categories such as “elements” and “components,” which seem to parallel the atomic design metaphor by Brad Frost of starting with the smallest pieces (atoms) and then combining them into slightly more complex items (molecules), and so on.

This made sense to us, so we tried out a navigation scheme that followed that basic principle. And you can see this in an early prototype we built, where we have four top level categories and different sections organized within each one.

What we heard: confusion

Colin:

But when we tested this with users, we found out that this metaphor didn't work so well. People could pretty easily find buttons in the elements category, but struggled with items like tables, which could be elements (they are HTML elements!) but also seem like more complex molecules.

“A component feels like it could be made up of many things, not just elements but lots of stuff.”

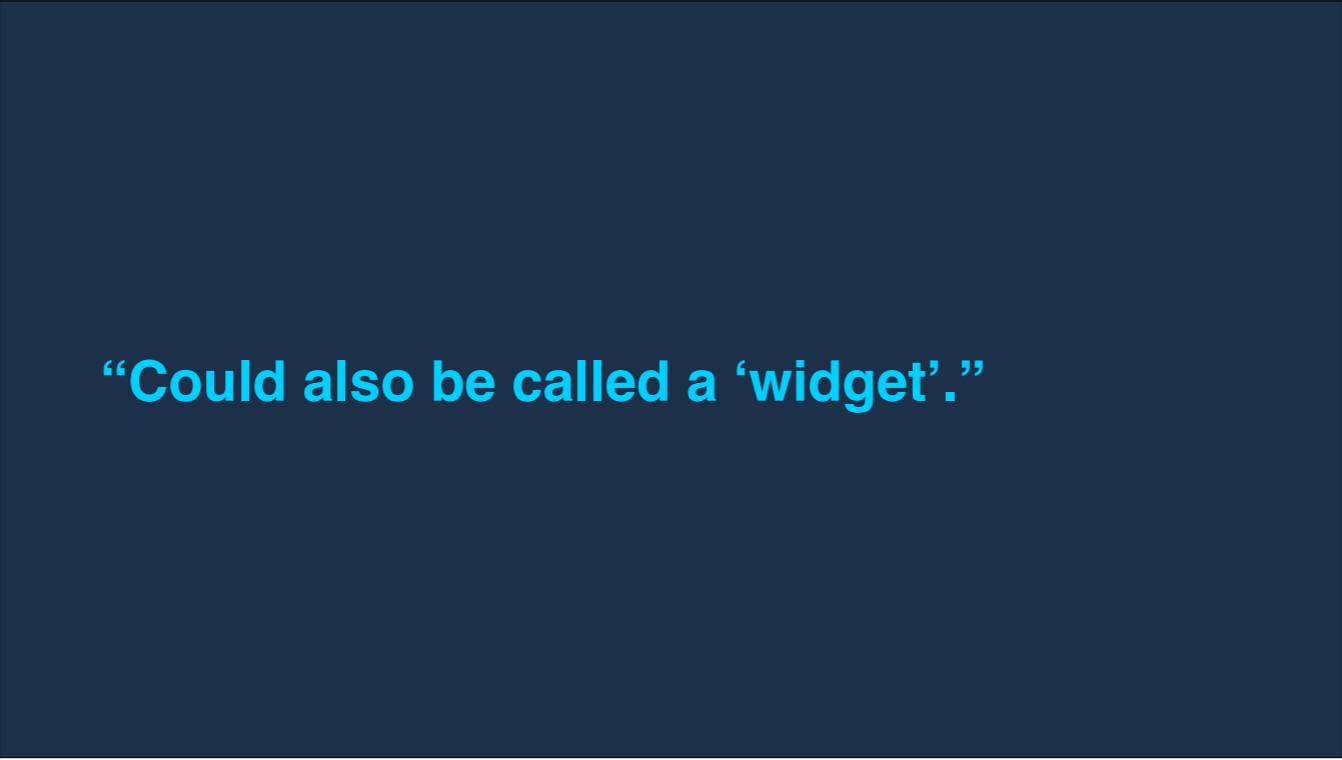
Colin

That was because people couldn't really agree on what the difference between a “component” and an “element” was. One participant said: “A component feels like it could be made up of many things, not just elements but lots of stuff.”

“Components and elements are one and the same.”

Colin

And another said: “Components and elements are one and the same.” Um. Ok.



“Could also be called a ‘widget’.”

Colin reads.

And then another said “could also be called a widget.”

Carolyn swears.

What worked:

The screenshot shows a web page titled "Draft U.S. Web Design Standards". At the top, there's a banner with the text "An official website of the United States Government" and the American flag. Below the title, there's a sidebar with a vertical navigation menu. The menu items are: Getting started (which is highlighted in blue), Visual style, Grid, Buttons, Labels, Tables, Alerts, Accordions, Form controls, Form templates, Search bar, Side navigation, and Footers. To the right of the sidebar, the main content area has a large heading "Getting started". Below it, there's a paragraph of text and a link: "The Draft U.S. Web Design Standard set a new bar for simplicity and consistency across government services, while providing out-of-the-box design and code." and "Learn more about why designing consistent digital government services is important in our [blog post introducing the standards](#)". Further down, there's another section titled "For developers" with some descriptive text.

Carolyn: So we went back to the drawing board. After also conducting some card sorting exercises we realized that there weren't really any consistent patterns to how people were organizing the parts of the pattern library. Certain things – such as form elements and patterns – were usually grouped together but beyond that there wasn't much consistency. So we went back and created a mostly flat navigation scheme that you see here. Subsequent testing showed that users were able to browse the website much more easily with this scheme.

3/ Using things

18F

Colin:

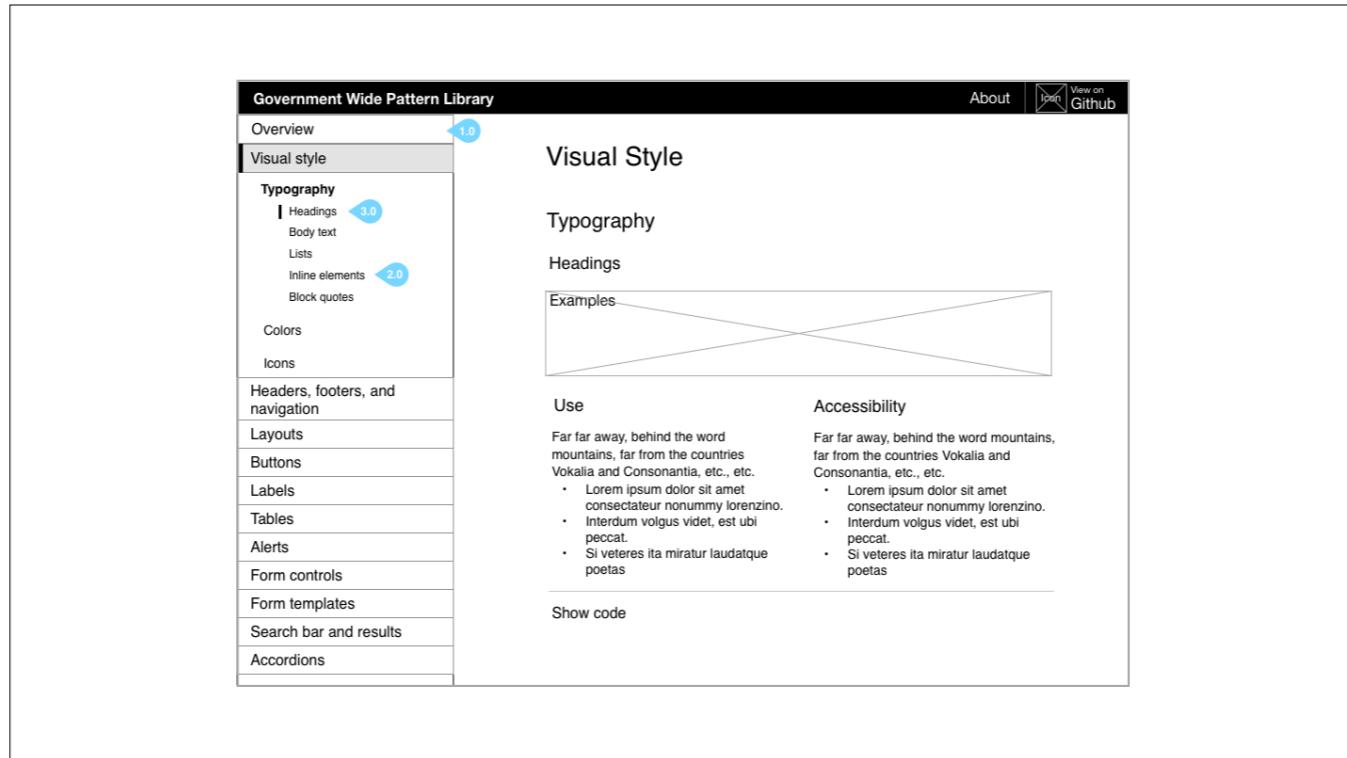
Once people are able to find things in the library, they are going to set about using them. We also wanted to research this process. Specifically, we were interested in learning about how our colleagues preferred to download the code, but we found out a lot more in the process.

Expectations

- 1**
Find the thing
- 2**
Get the thing
- 3**
Use the thing

Colin:

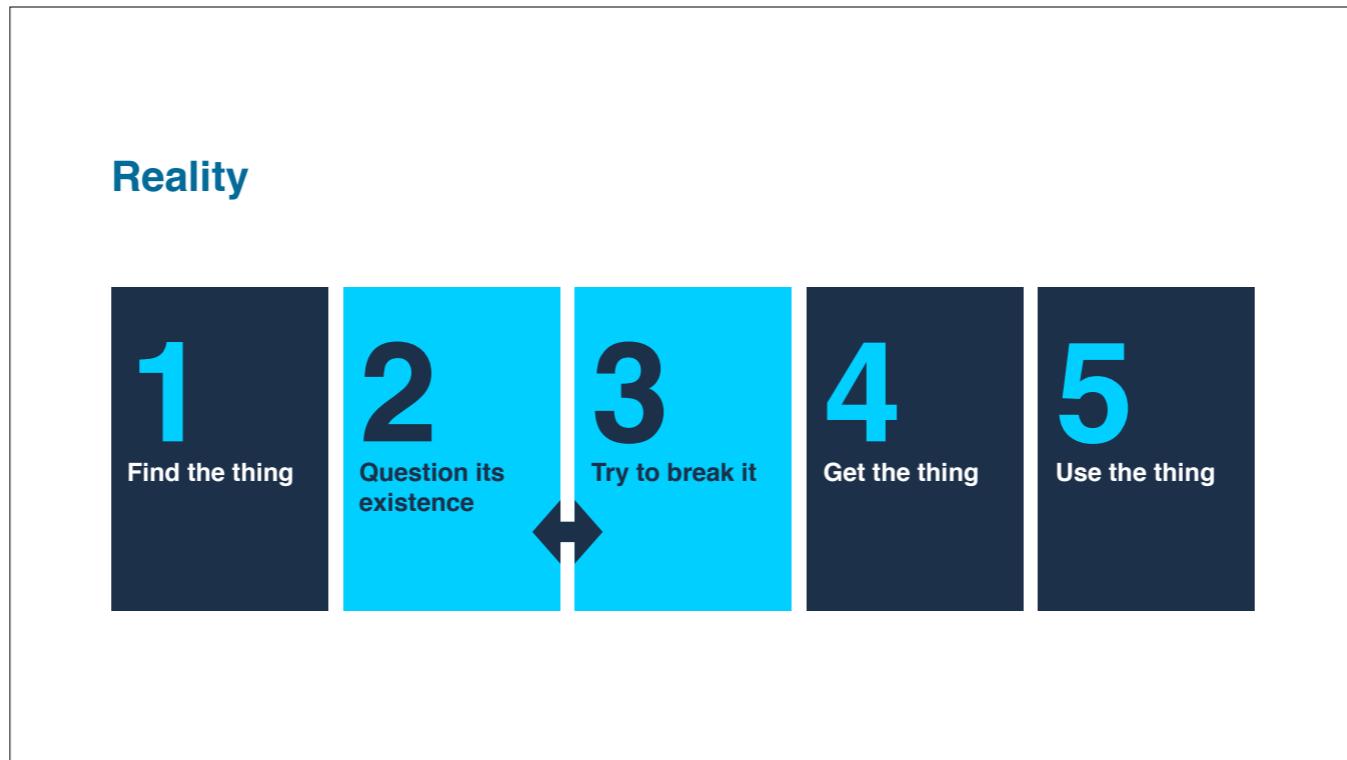
This is the general flow we expected to see. Very straightforward and linear. <read the flow>



Colin:

And so this is an early wireframe based on those expectations. We included many of the things you'd expect to see in a pattern library – a live example, some words about it, and some code.

We also expected this to be a fairly linear process – a person would find a pattern, learn a little bit more about it, and then grab the code.



Carolyn:

But when we actually went out and tested with our colleagues, we found out it's a little more complicated.

Both designers and developers are a bit more skeptical than we had anticipated, and rightly so. Designers wanted to know why we had made the design decisions we had. What research did we have to back it up?

Developers wanted to 'kick the tires' of the components in a way. They would get into the web inspector and start tweaking the component on the spot, trying to see if it would break in ways they've seen components break before.

Everyone we talked to indicated that they wanted to be able to evaluate the components before committing to download something.

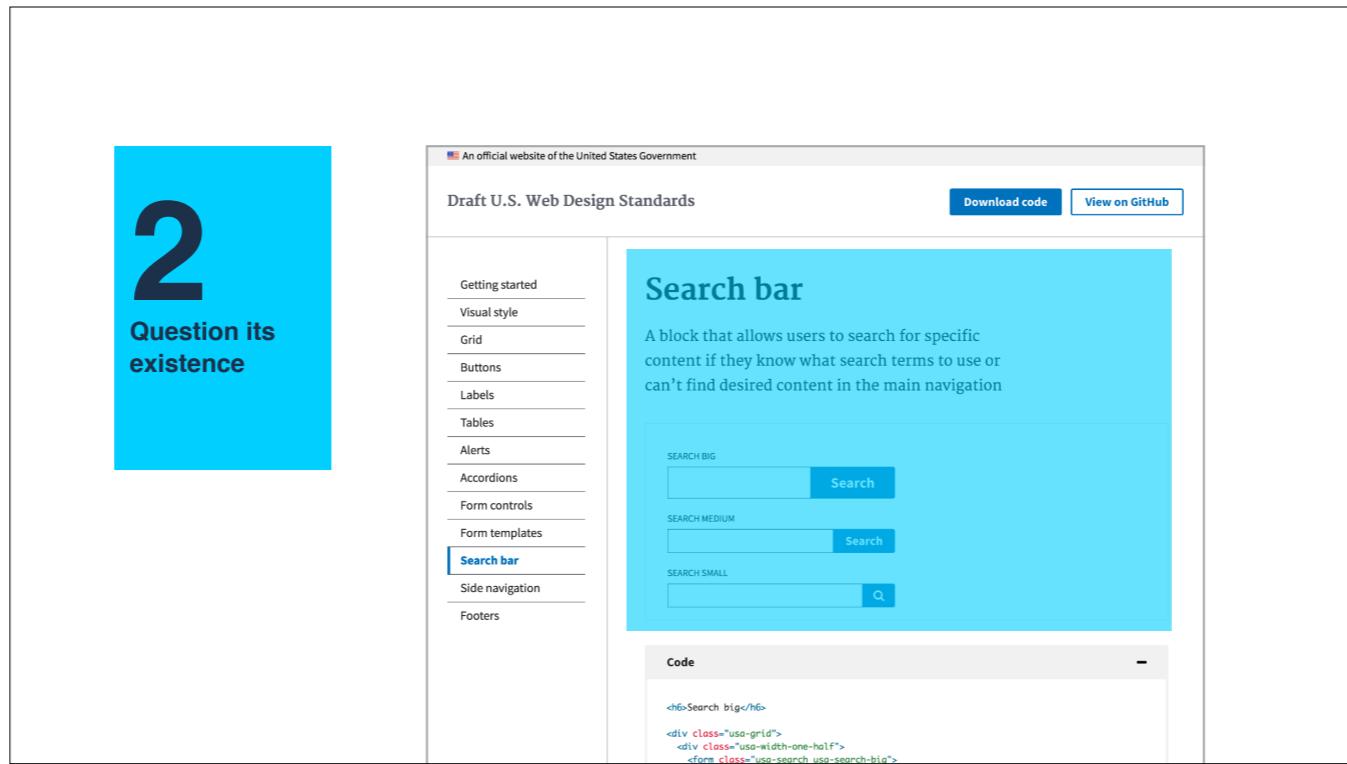
Only then when a component passed this exploration phase were our colleagues willing to download it to play further.

So let's look at how this process plays out in the page layout.

The image consists of two side-by-side panels. The left panel is a dark blue square containing a large white number '1' and the text 'Find the thing' below it. The right panel is a screenshot of a website titled 'Draft U.S. Web Design Standards'. At the top, there's a navigation bar with links for 'Download code' and 'View on GitHub'. Below the title, there's a sidebar with a list of design components: 'Getting started', 'Visual style', 'Grid', 'Buttons', 'Labels', 'Tables', 'Alerts', 'Accordions', 'Form controls', 'Form templates', 'Search bar' (which is highlighted in blue), 'Side navigation', and 'Footers'. The main content area is titled 'Search bar' and describes it as a block that allows users to search for specific content. It shows three examples: 'SEARCH BIG' with a large input field and a 'Search' button, 'SEARCH MEDIUM' with a medium-sized input field and a 'Search' button, and 'SEARCH SMALL' with a small input field and a magnifying glass icon. At the bottom, there's a 'Code' section with a collapse button and some HTML/CSS code.

Colin:

First we have step one: finding the thing. This is the navigation part of the page, which we already talked about.



Carolyn:

The second(ish) step involves questioning the component's existence. Users want to know why we're recommending this particular pattern, and why is it the way it is? Does it match their own expectations for how it should appear and behave?

The live example supports this exploration as it allows people to play with the piece and see how it behaves in real life.

The screenshot shows a component library interface. On the left, there's a sidebar with navigation links: Labels, Tables, Alerts, Accordions, Form controls, Form templates, **Search bar** (which is highlighted), Side navigation, and Footers. The main content area has a title "can't find desired content in the main navigation". Below this is a section titled "SEARCH BIG" with a search input field and a blue "Search" button. Underneath it is "SEARCH MEDIUM" and "SEARCH SMALL", each with their respective search input fields and "Search" buttons. At the bottom, there's a "Code" section containing the HTML code for these three search components.

```
<h2>Search big</h2>


<form class="usa-search usa-search-big">
  <div role="search">
    <label class="usa-sr-only" for="search-field-big">Search big</label>
    <input id="search-field-big" type="search" name="search">
    <button type="submit">
      <span class="usa-search-submit-text">Search</span>
    </button>
  </div>
</form>


<h2>Search medium</h2>


<form class="usa-search">
  <div role="search">
    <label class="usa-sr-only" for="search-field">Search medium</label>
    <input id="search-field" type="search" name="search">
    <button type="submit">
      <span class="usa-search-submit-text">Search</span>
    </button>
  </div>
</form>


<h2>Search small</h2>
```

Colin:

We moved the code snippet to appear right underneath the example, because we found that developers in particular evaluated the example of the component and how it's been coded simultaneously. Having the two next to each other on the page makes this process easier.

The screenshot shows a documentation page for the 'Search' component. At the top, there's a large blue callout box containing the number '2' and the text 'Question its existence'. Below this, the page is divided into several sections:

- Side navigation**: A sidebar with a 'Footers' section.
- Code snippets**: Two examples of search form code, one for 'medium' and one for 'small' search fields.
- Documentation**: A main content area with the following sections:
 - Accessibility**:
 - A note about customizing the form template to follow accessibility guidelines.
 - A tip about including the word 'search' in the button element for screen readers.
 - Usability**:
 - A note about users who are confused by navigation systems.
 - When to use**:
 - A note about users who would benefit from being able to search the site.
 - When to consider something else**: No content is shown here.

Colin:

The documentation also helps with this process. We found that many users really value the accessibility guidance because they are less familiar with best practices in this area, so we moved that guidance to the top.

Users seem to refer to the documentation more often when they are already building out their own projects, and want to be sure they are implementing things correctly. Contrast that with people who are standing up prototypes very quickly, who don't have time to come back and check the documentation. Moving this to the bottom of the page gets it out of the way for the clock racers but still makes it available for those who want it.

The screenshot shows the 'Search bar' component from the Draft U.S. Web Design Standards. The sidebar includes links for Getting started, Visual style, Grid, Buttons, Labels, Tables, Alerts, Accordions, Form controls, Form templates, Search bar (which is selected), Side navigation, and Footers. The main content area displays three search bar variations: 'SEARCH BIG', 'SEARCH MEDIUM', and 'SEARCH SMALL'. Each variation includes a search input field and a 'Search' button. Below each variation is a 'Code' section with the following HTML and CSS:

```
<h2>Search big</h2>
<div class="usa-grid">
  <div class="usa-width-one-half">
    <form class="usa-search usa-search-big">
      <div role="search">
        <label class="usa-sr-only" for="search-field-big">Search big</label>
        <input id="search-field-big" type="search" name="search">
        <button type="submit">
          <span class="usa-search-submit-text">Search</span>
        </button>
      </div>
    </form>
  </div>
</div>

<h2>Search medium</h2>
```

Carolyn:

The third(ish) step is trying to break the thing. This is the part where people would look at the code and evaluate whether it seems to be semantic and clean. They will play with the browser width to see how it responds, and even dig into the web inspector to see how easy it is to tweak and modify the component. The example and the code snippet primarily serve this need (another reason that these two are together).

The image consists of two parts. On the left is a large, bold blue number '4' centered on a dark blue rectangular background. To the right is a screenshot of a website titled 'Draft U.S. Web Design Standards'. The top navigation bar includes links for 'Download code' and 'View on GitHub'. A sidebar on the left lists various design components: Getting started, Visual style, Grid, Buttons, Labels, Tables, Alerts, Accordions, Form controls, Form templates, **Search bar**, Side navigation, and Footers. The main content area is titled 'Search bar' and describes it as a block that allows users to search for specific content. It shows three examples: 'SEARCH BIG' with a large input field and a blue 'Search' button; 'SEARCH MEDIUM' with a medium-sized input field and a blue 'Search' button; and 'SEARCH SMALL' with a small input field and a blue magnifying glass icon. Below this is a 'Code' section containing the following HTML code:

```
<h2>Search big</h2>
<div class="usa-grid">
  <div class="usa-width-one-half">
    <form class="usa-search usa-search-big">
```

Colin:

The fourth step is actually downloading the code. Since this clearly comes after the evaluation phase, we wanted to make it easily accessible but out of the way. Putting it at the very top means that it's easy to find, but it's not interfering with the other parts of the process.

Find the thing

Get the thing

Question its existence and try to break it

Question its existence

Question its existence

Carolyn:

And this is how the entire thing fits together on the page.

The navigation and the download button – the only two distinct phases are separated out from the main display page. And the parts of the page that help with the evaluation part of the process are close together, and in an order that makes sense for the mental process we observed in our users.

3/ Takeaways

18F

25

Colin:

So what did we learn from all this?

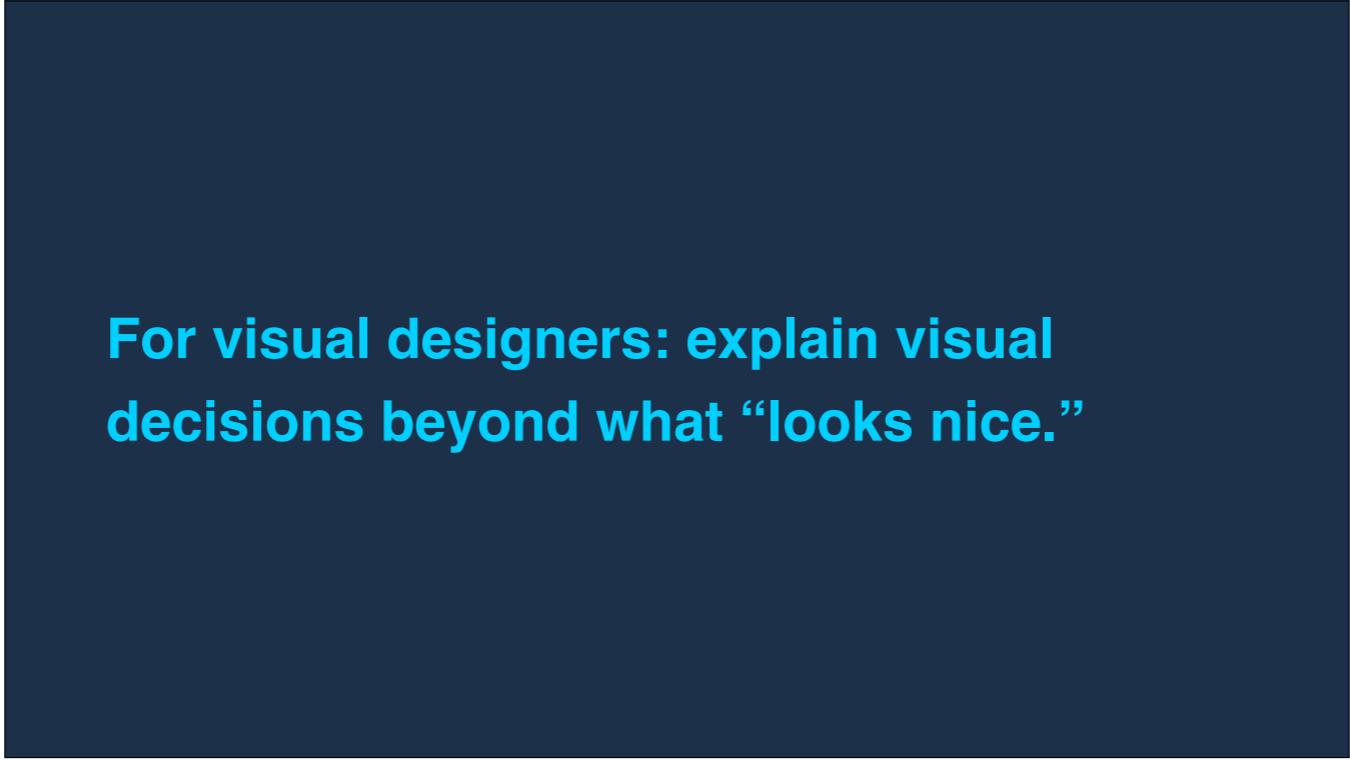
Elements/components are a useful metaphor for building a pattern library, not for organizing it.

Colin:

First of all, thinking in elements and components may be a useful metaphor for building a pattern library, but it's not so useful for organizing it. Even if these terms seem intuitive, our implementers had lots of different ideas about what belonged in each category.

For developers: offer multiple ways test out components.

Carolyn: Allow developers to evaluate your components in many different ways. Developers will want to test out the example and see if it behaves as they'd expect. They'll want to look at the code to see if it's clean and semantic. And then they'll want to tweak that code and see how easy it is to modify, or if the component is going to break in ways they've seen before.



For visual designers: explain visual decisions beyond what “looks nice.”

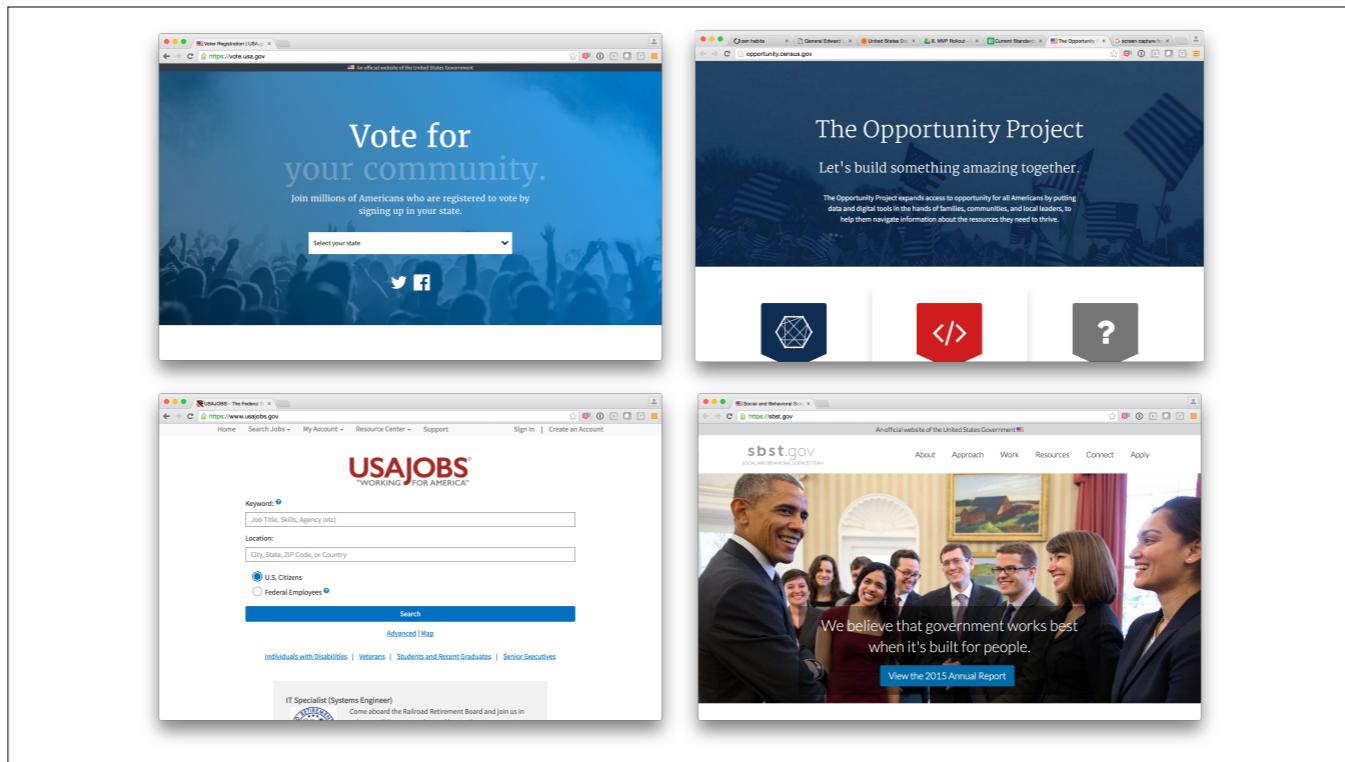
Colin

Designers also evaluate each component but in a different way. For visual designers, explain the visual decisions beyond what “looks” nice. For us, that meant using our documentation to explain why we made certain visual design choices and what they intended to achieve.

For UX designers: give the reasons and research for your decisions.

Carolyn

UX designers want to understand the reasons behind your design decisions, and want to know what kind of research led you to these particular recommendations.



Colin: And above all, remember to not just design your library for yourself, but for all the designers, developers and others that might want to – or have to – play with it. If you take the time to make your pattern library a delight for others to use, not only will more people use it, but they'll use it in better ways!

Thanks!

Contact 18F@gsa.gov

Follow [@18F](#)

Work together github.com/18F

Visit the standards standards.usa.gov

Download our slides github.com/18F/wds-ia-summit-presentation

18F

