

RESILIENT WEB DESIGN

C H A P T E R 7 :

Challenges

THE FOURTH ANNUAL CONFERENCE ON hypertext took place in San Antonio, Texas in December 1991. Tim Berners-Lee's World Wide Web project was starting to take shape then. Thinking the conference organisers and attendees would appreciate the project, he submitted a proposal to Hypertext '91. The proposal was rejected.

The hypertext community felt that the World Wide Web project was far too simplistic. Almost every other hypertext system included the concept of two-way linking. If a resource moved, any links pointing to that resource would update automatically. The web provided no such guarantees. Its system of linking was much simpler—you just link to something and that's it. To the organisers of Hypertext '91, this seemed hopelessly naïve. They didn't understand that the simplicity of the web was actually its strength. Because linking was so straightforward, anyone could do it. That would prove to be crucial in the adoption and success of the World Wide Web.

It's all-too tempting to quickly declare that an approach is naïve, overly simplistic, and unrealistic. The idea that a website can simultaneously offer universal access to everyone while also providing a rich immersive experience for more capable devices ...that also seems hopelessly naïve.

This judgement has been handed down many times over the history of the web.

“This is too simple”

When the Web Standards Project ran its campaign encouraging designers to switch from TABLES for layout to CSS, it was met with resistance. Time and time again they were criticised for their naïvety. “Sure, a CSS-based layout might be fine for a simple personal site but there’s no way it could scale to a large complex project.”

Then Doug Bowman spearheaded the CSS-based redesign of Wired.com and Mike Davidson led the CSS-based redesign of ESPN.com. After that the floodgates opened.

When Ethan Marcotte demonstrated the power of responsive design, it was met with resistance. “Sure, a responsive design might work for a simple personal site but there’s no way it could scale to a large complex project.”

Then the Boston Globe launched its responsive site. Microsoft made their homepage responsive. The floodgates opened again.

It's a similar story today. "Sure, progressive enhancement might work for a simple personal site, but there's no way it could scale to a large complex project."

The floodgates are ready to open. We just need you to create the poster child for resilient web design.

“This is too difficult”

Building resilient websites is challenging. It takes time to apply functionality and features in a considered layered way. The payoff is a website that can better react to unexpected circumstances—unusual browsers, flaky network connections, outdated devices. Nonetheless, for many web designers, the cost in time seems to be too high.

It’s worth remembering that building with progressive enhancement doesn’t mean that *everything* needs to be made available to *everyone*. Instead it’s the core functionality that counts. If every single feature needed to be available to every browser on every device, that would indeed be an impossibly arduous process. This is why prioritisation is so important. As long as the core functionality is available using the simplest possible technology, you can—with a clear conscience—layer on more advanced features.

Still, it's entirely possible that this approach will result in duplicated work. If you build an old-fashioned client-server form submission process and then enhance it with JavaScript, you may end up repeating the form-processing on the client as well as the server. That can be mitigated if you are also using JavaScript on the server. It's theoretically possible to write universal JavaScript so that the server and browser share a single codebase. Even without universal JavaScript, I still think it's worth spending time to create technical credit.

The UK's Government Service design manual provides an even shorter form of the three-step process I've outlined:

1. First, just make it work
2. Second, make it work better

The design manual also explains why:

“*If you build pages with the idea that parts other than HTML are optional, you'll create a better and stronger web page.*”

This kind of resilience means that the time you spend up-front is well invested. You might also notice an interesting trend; the more you use this process, the less time it will take.

Moving from TABLES to CSS seemed like an impossibly idealistic goal. Designers were comfortable using TABLE and FONT elements for layout. Why would they want to learn a whole new way of working? I remember how tricky it was to make my first CSS-based layouts after years of using hacks. It took me quite some time. But my second CSS-based layout didn't take quite so long. After a while, it became normal.

Designers comfortable with fixed-width layouts had a hard time with responsive design. That first flexible layout was inevitably going to take quite a while to build. But the second flexible layout didn't take quite so long. After a while, it became normal.

It's no different with the layered approach needed for building resilient websites. If you're not used to working this way, the first time you do it will take quite some time. But the second time won't take quite so long. After a while, it will become normal.

There may well be situations where the three-step approach won't work, but they're not as common as you might think. If you're building a 3D game in a web browser, the simplest technology capable of providing the core functionality will still be pretty complex. That said, I'd love to see a text-only adventure game get enhanced into a first-person shooter.

“How do I convince...?”

The brilliant computer scientist Grace Hopper kept an unusual timepiece on her wall. It ran counter-clockwise. When quizzed on this, she pointed out that it was an arbitrary convention, saying:

“*Humans are allergic to change. They love to say, “We’ve always done it this way.” I try to fight that. That’s why I have a clock on my wall that runs counter-clockwise.*”



Commodore Grace M. Hopper,
United States Navy.

Behaviour change is hard. Even if you are convinced of the benefits of a resilient approach to web design, you may find yourself struggling to convince your colleagues, your boss, or your clients. It was ever thus. Take comfort from the history of web standards and responsive design. Those approaches were eventually adopted by people who were initially resistant.

Demonstrating the benefits of progressive enhancement can be tricky. Because the payoff happens in unexpected circumstances, the layered approach is hard to sell. Most people will never even know whether or not a site has been built in a resilient way. It's a hidden mark of quality that will go unnoticed by people with modern browsers on new devices with fast network connections.

For that same reason, you can start to implement this layered approach without having to convince your colleagues, your boss, or your clients. If they don't care, then they also won't notice. As Grace Hopper also said, "it's easier to ask forgiveness than it is to get permission."

Tools

Changing a workflow or a process can be particularly challenging if it clashes with the tools being used. A tool is supposed to help people get their work done in a more efficient way. The tool should be subservient to the workflow. All too often, tools instead dictate a preferred way of working. Whether it's WYSIWYG editors, graphic design programs, content management systems, or JavaScript frameworks, tools inevitably influence workflows.

If you are aware of that influence, and you can recognise it, then you are in a better position to pick and choose the tools that will work best for you. There are many factors that play into the choice of frameworks, for example: “Is it well-written?”, “Does it have an active community behind it?”, “Does it have clear documentation?”. But perhaps the most important question to ask is, “Does its approach match my own philosophy?”

Every framework has a philosophy because every framework was written by people. If your philosophy aligns with that of the framework, then it will help you work more efficiently. But if your philosophy clashes with that of the framework, you will be fighting it every step of the way. It might even be tempting to just give up and let the framework dictate your workflow. Then the tail would be wagging the dog.

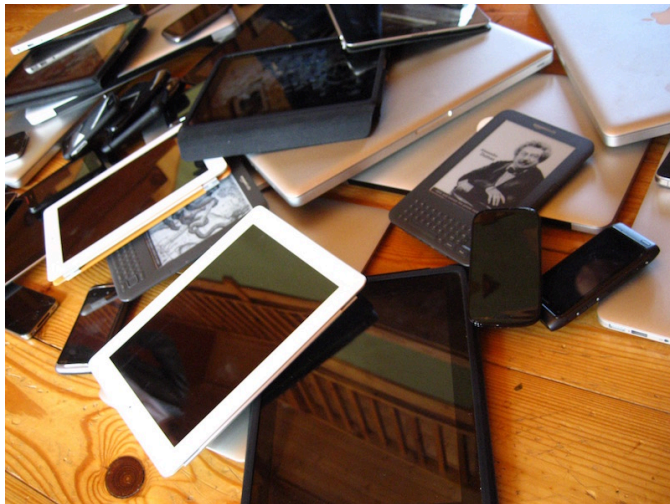
Choose your tools wisely. It would be a terrible shame if you abandoned the resilient approach to web design because of a difference of opinion with a piece of software.

Differences of opinion often boil down to a mismatch in priorities. At its heart, the progressive enhancement approach prioritises the needs of people, regardless of their technology. Tools, frameworks, and code libraries, on the other hand, are often built to prioritise the needs of designers and developers. That's not a bad thing. Developer convenience is hugely valuable. But speaking personally, I think that user needs should trump developer convenience.

When I'm confronted with a problem, and I have the choice of making it the user's problem or my problem, I'll make it my problem every time. That's my job.

Future friendly

In September of 2011, I was speaking at a conference in Tennessee along with some people much smarter than me. Once the official event was done, we lit out for the countryside where we had rented a house for a few days. We were gathering together to try to figure out where the web was headed.



A small sample of internet-enabled devices.

We were, frankly, freaked out. The proliferation of mobile devices had changed everything. Tablets were on the rise. People were talking about internet TVs. We were hoping to figure out what the next big thing would be. Internet-enabled fridges, perhaps?

In the end, the only thing we could be certain of was uncertainty:

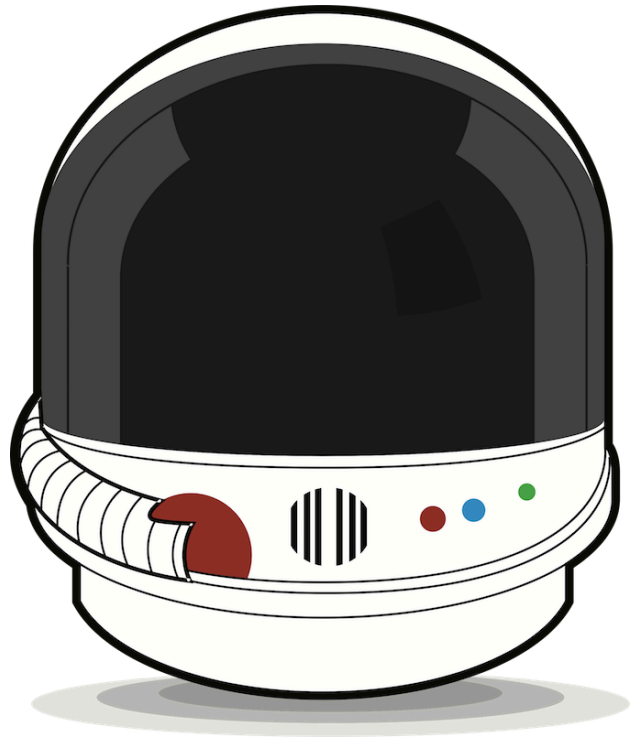
“*Disruption will only accelerate. The quantity and diversity of connected devices—many of which we haven’t imagined yet—will explode, as will the quantity and diversity of the people around the world who use them.*”

That isn’t cause for despair; it’s cause for celebration. We could either fight this future or embrace it. Realising that it was impossible to be future-proof, we instead resolved to be future-friendly:

1. Acknowledge and embrace unpredictability.
2. Think and behave in a future-friendly way.
3. Help others do the same.

That first step is the most important: acknowledging and embracing unpredictability. That is the driving force behind resilient web design. The best way to be future-friendly is to be backwards-compatible.

The Future Friendly symbol.



Assumptions

“We demand rigidly-defined areas of doubt and uncertainty!” cried the philosophers in Douglas Adams’ *Hitchhiker’s Guide To The Galaxy*.

As pattern-matching machines, we are quick to identify trends and codify them into assumptions. Here are just some assumptions that were made over the history of web design:

- Everyone has a monitor that is 640 pixels wide.
- Everyone has the Flash plugin installed.
- Everyone has a monitor that is 800 pixels wide.
- Everyone has a mouse and a keyboard.
- Everyone has a monitor that is 1024 pixels wide.
- Everyone has a fast internet connection.

The proliferation of mobile devices blew those assumptions out of the water. The rise of mobile didn't create new uncertainties—instead it shone a light on the uncertainties that already existed.

That should have been a valuable lesson. But before too long the old assumptions were replaced with new ones:

- There are some activities that people will never want to do on their phones.
- Every phone has a touch screen.
- Everyone using a phone is in a hurry.
- Every browser on every phone supports JavaScript.

These kind of assumptions always remind me of old physicist jokes. “Assume a perfectly spherical web browser...”

Assumptions are beguiling. If only we could agree on certain boundaries, then wouldn't web design be so much easier to control?

As tempting as this siren call is, it obfuscates the true nature of the ever-changing uncertain web. Carl Sagan put it best in his book *The Demon-Haunted World*:

“ *It is far better to grasp the universe as it really is than to persist in delusion, however satisfying and reassuring.* ”

The future

I wish I could predict the future. The only thing that I can predict for sure is that things are going to change.

I don't know what kind of devices people will be using on the web. I don't know what kind of software people will be using on the web.

The future, like the web, is unknown.

The future, like the web, will be written by you.