

# DON'T BE FOOLED BY JAVASCRIPT'S UBIQUITY: How to Avoid Classic Web Development Pitfalls

## WEB DEVELOPMENT



Spaghetti code is a recipe for disaster. And modernizing a large, complex web application can feel like rewiring an airplane in mid-flight. The key to success is having a clear strategy to manage the process. And when it comes to web development trends, the old adage of “the more things change, the more they stay the same” really does apply.

---

Developers today have a wealth of choices for web-based programming languages and frameworks. The capabilities of React and Angular to create custom functionality — the former in building interactive UI components for example, compared with the latter in building enterprise web applications — are immense. [Stack Overflow's 2023 developer survey](#) lists no fewer than 34 web frameworks and web technologies. Still, most of the code running today is likely built on older technologies.

---

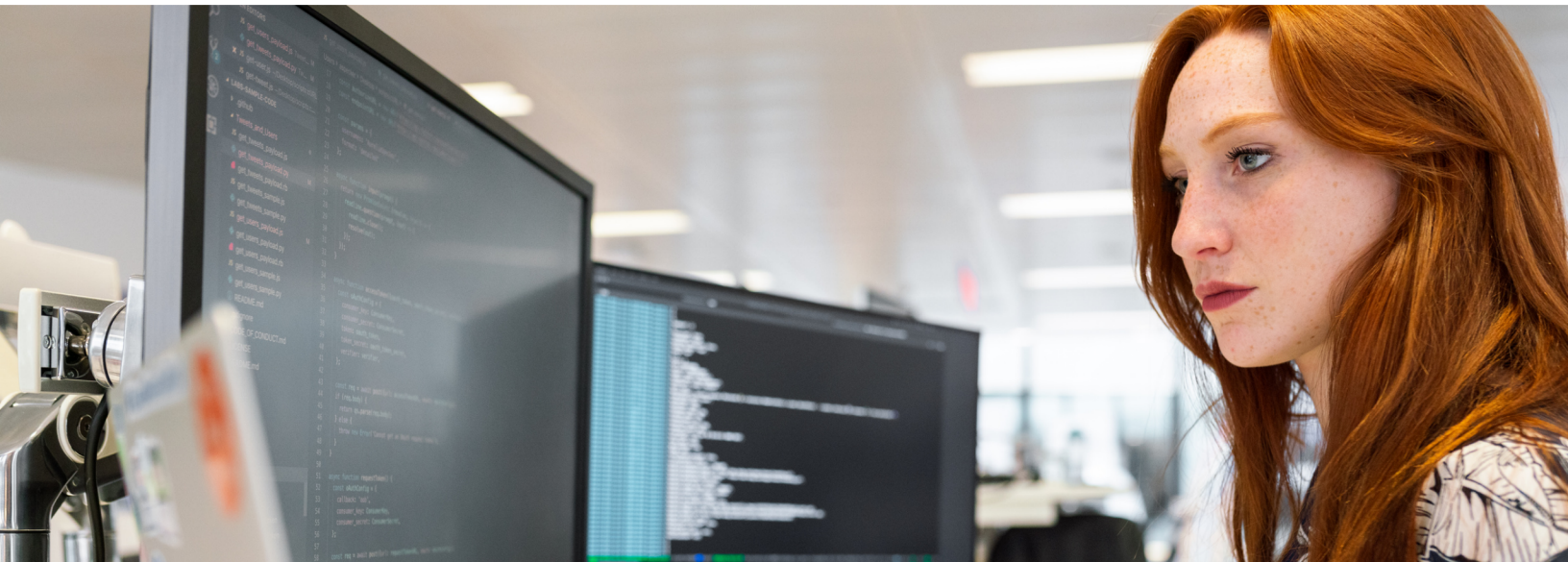
This understandably has an impact on framework adoption. Node.js and React.js remain by some distance the most common web frameworks used by developers of all industries and experience levels. Taking markup languages into account as well, the Stack Overflow survey showed a clearly enduring appeal. Users of JavaScript, TypeScript and HTML/CSS all selected those three languages as the top three they want to use the following year. JavaScript remains the most popular language for web development; it was the “[near-universal choice](#)” for developers with 96% community adoption, although object-oriented TypeScript (a transpiler for JavaScript) is coming quickly.

---

## So with this in mind, why are there so often pain points for organizations looking to harness the power of custom web applications, from classic web pages and site design, to sophisticated browser based apps?

To get the answer, let's look at a longer-term client project and how a trusted technical partner like Growth Acceleration Partners (GAP) is able to navigate the occasionally choppy waters from legacy technologies and “bits-and-pieces” projects into a modernized application stack.

One particular GAP client in the food and beverage industry has commissioned multiple projects, with some overlapping elements. One was a major intranet project with many different facets, from human resources, to product management and branding. The original stack presented by the client was outdated; front-ends based on the jQuery library and “vanilla” JavaScript, written in an antiquated way. With the project lending itself very well to a stateful framework such as React or Angular, the obvious solution was to rebuild. However, the client preferred to modernize what already existed over a complete rewrite.



Therefore, the first job was to make the code manageable. From a technical perspective, this was not as easy as it sounded. It meant digging through repositories that had been maintained over the years by dozens — if not hundreds — of developers, with their own styles and idiosyncrasies. And it included auditing all the different modules of JavaScript code so anyone could go in, find something, and fix or enhance it where necessary.

The next step was to, over time, build new modules for the client. These modules, with the architectural shackles off, would utilize much more modern tools; they were built from scratch in Rails and with React. Eventually, the process began moving toward a robust revamping of the client's intranet, taking the existing code and — either one module at a time or paired, if they were similar enough — rebuilding them.

---

**One of the key challenges with building newer modules and integrating them as part of a wider application is to keep the user experience seamless. You do not want the end user to suffer latency or other performance issues as they move between the different modules.**

---

The benefits of libraries are well-known, from saving time, to greater flexibility — enabling complex applications — to easier and more efficient problem solving. For example, the plethora of open source plugins enables the feature-rich jQuery to be a very useful tool, particularly with regard to animation and handling events. Similarly, React's reusable components are a perfect fit for dynamic and responsive web applications with interactive UIs. But it is not a one-size-fits-all.



A frequently cited weakness of jQuery, for example, is in its unsuitability for complex projects due to limited features. By comparison, React is less suitable for small projects — a single web page, or if the user is not required to change anything — due to its steep learning curve. The right library can open up an immense world of possibilities for your application; so long as you pick the right one.

While you can get ideal results when starting from a blank page, this is not always the case. With some projects, you have to go about things a little differently. It is not uncommon to have an application start off small, whatever the library, and gradually increase in complexity without the foundations being fortified. In this case, the code is being refactored. The result can often be spaghetti code.



Many times, a client has presented a project that has either been orphaned, or has limped badly, because of decision-making that has not been forward-thinking. A good example of this are companies who have a WordPress site but lack the technical expertise to maintain it. They then add plugins and libraries to assist functionality, which then have to be maintained until they are deprecated or reach end of life.

It is worth noting as well that, like any project, time and budget will not be unlimited, and trade-offs are required. In this instance, there may not be time to write a front-end interface, or the client is unable to pay for it. So the decision then becomes: do you want the trade-off to be writing the software yourself, or maintaining a library you did not write?

Part of working with a partner such as GAP is the collated experience on client projects and ability to get these kinds of decisions right. To bolster their existing expertise, in 2023, GAP acquired Mission Data, a full-service digital product development firm with experience and success across web development.

When engagement pitfalls and practices that create unsustainable environments are avoidable, you can rely on the expertise of a skilled web development team to handle those projects while your organization focuses on what it does best. By using the right tools, libraries and frameworks, you can get your web application to do pretty much anything you want.

01022024

To find out more, please visit [WeAreGAP.com](https://www.WeAreGAP.com) 