



Needs Are Situational

# Needs Are Situational

When it comes to web development, developers experience myriad frustrations, which can represent opportunities for browser vendors to address that frustration and thereby fulfilling a need.

While this report focuses on the top ten needs of web developers, there's more nuance to the frustrations than a top ten list can convey because needs are not static. At any given time, a web developer may be in a situation that affects what they find frustrating, why it's frustrating, and how it causes frustration. Four main situations emerged from the pilot interviews.

- Being a part of the broad web development community.
- Working on a project for oneself without external stakeholders.
- Working on a project with requirements.
- Adapting to work cohesively within an organization or as part of a team.

Let's consider some examples. First, we know one of the top 10 needs was, "Keeping up with a large number of new and existing tools or frameworks." Why and how this is frustrating is experienced differently in different situations.

## **Working on a project for oneself without external stakeholders.**

*"When I write code for myself I'm often doing something that's simple so I don't need frameworks, but more than that, I like to understand how stuff actually works and frameworks hide the implementation details."*

## **Adapting to work cohesively within an organization or as part of a team.**

*"When I go to work, we use Angular and it's fine. If I was working somewhere else that used a different framework, I would have to adapt and learn that new framework. It's a byproduct of what management*

*or senior developers dictate. I work with developers that are not willing to make a change or try different frameworks. Their preference is to stick with something that's tested or something that's already been approved by legal. For instance, the team talked about using jQuery to build a web app. We could do that, but for efficiencies sake Angular or React would be better. Legal got involved and wouldn't let us use React due to concerns about licensing. We stuck with jQuery."*

## **Being a part of the broad web development community:**

*"I have a community of web developers. We all have different perspectives about what's popular at the moment, what's worth learning or not. One friend encouraged me not to learn Node JS because he felt it was too specific. But, I've also read many times that it's useful and a good thing to learn. Angular is another example of something that I didn't even know existed six months ago and I'm already getting competing opinions about whether it's worth the time to learn."*

The type of project a developer works on can create or eliminate frustrations because of the project's requirements.

*"Our software only uses anonymized data that we received from a third party, so [privacy] is not very frustrating for us. [We] don't actually have any identifying data."*

*"I'm happy if my stuff runs well in Chrome. I'm not...trying to target old versions of Internet Explorer or something like that. So I don't really care [that much about browser compatibility]. But it is important, and I would expect it to be important to somebody who hired me to build something."*

*"I haven't produced real marketable stuff [where I] had to worry about [accessibility]...I do a lot of this stuff for my own amusement. I'm playing with it so it just hasn't been an issue for me. I recognize it's import-*

ant, but I haven't had just haven't had to deal with it."

"Well, I think of available tools that you could I'm thinking of like dev ops, which I don't really need. So I don't know if that really applies to me there. Because I'm not doing anything complex enough to require, like any deployment, to, you know, more than one place or any real integration, outside resources. So I think I'm going to leave that alone."

"I'm basically trying to run a web server host some HTML, JavaScript, CSS, and deliver this to my clients. So just keeping a single web server up and running. It's easier than having a complex framework of micro-services powering my application."



Broader Picture

## Externalities

The evolution of the web has led to the state of affairs where developers can pay for services, such as hosting, which reduces the need to own and maintain all of the infrastructure that was previously required to put something online. However, each piece you don't own creates a dependency on someone or something external to your work. Dependencies present risks, and developers have to decide whether the tradeoff is worth it or not. While the upside is that making things for the web is much faster and easier, there's a fear that their work might become obsolete should something happen to those who they are externally dependent on.

*"...owning the entire spectrum is excessive, it's just ensuring that those trade-offs are either things that you [can] control the copy of that you're using, or that you're paying for service that you can take your money elsewhere."*

*"Unless you are starting with HTML, CSS, JavaScript and writing your own Ajax requests, using the breadth of the browser object model, XML HTTP request API to shoot code back and forth...you're always giving up some control over your code...In terms of importing code dependencies, that's the cost of doing business. It just makes writing human usable applications so much faster. But the act of hosting is following that same path...the system of hosting is also being abstracted away to third parties. There are horror stories every week of somebody whose business was just stuck...years of work [gone] because they had built everything into Google Cloud Platform and one of the developers that they had hired on to make some tweaks to their business application was using a blacklisted Gmail account. And because of that, the app certifications were all revoked....that's a huge externality, you have to think about...We're hosted on Heroku, we have stuff on AWS if by some chance X service provider disappeared, would your company also disappear? Like, if Amazon just poof was taken away by aliens tomorrow, our company would not exist...we have backups, we could spin up something else but there would be a week to two-week-long period*

*where we would just be building everything back up for availability."*

*"...my brother [a developer] feels he is at the whim of the creators of browsers...the concern is at some point, something that his work relies on will be obsolete by a decision that Google makes...or a decision that Microsoft makes...it's a little scary that things advance quickly...the thing that concerns me about changes [is that] updates would have to be made on account of third parties, such as the creators of the browsers... you don't have as much control over, for example, like the version of a compiler that you use, which is static...I'm referring to the obsolescence of technologies that people at one point relied on heavily...it's hard to predict which existing technology...will end up being the one in which a flaw is uncovered and a great deal of work left to be done to move to something else."*

## Browser Reborn

Developers have noticed that browsers are getting better, with Internet Explorer (IE) as the exception. Developers realize this wasn't just a kind move on the part of browser vendors to make developers' lives easier; they understand browsers have grown in monetary importance for their creators.

*"[Browsers have gotten better] because the industry started to care... ten years ago...the market wasn't as competitive, the browsers were not actually...trying to do a good job in implementing things apart from Firefox..."*

*"[Browsers had] been left as a second or third class citizen for ages. Now...Google is pushing on it because they live on ads so they need the browser to do whatever. Microsoft went back and started doing their own browser and it was actually good, but probably took them too much time to get up to speed and now they have to change because Edge was not as good as it should have been. Mozilla, well, Firefox since they changed to Quantum it's... it was not getting much love before...even Apple now is doing good things on the web, even if they're doing it at their own pace with their own reasons, but that's Apple you cannot expect too much..."*

Better browsers is a good thing, but the consolidation of browsers concerns to some. The first concern is about a lack of competition between browser vendors leading to a lack of technological advancements. Consolidation also has implications for compatibility, which manifests in different ways.

One view is that, as each browser vendor tries to outshine the other, those advances will create differences that may result in less compatibility. However, it's for the sake of advancements, so it's something developers with this view can live with.

On the flip side, some developers see competition creating more

compatible browsers because each will be trying to match at least what the other can do. But, as competition decreases, so might compatibility. Microsoft's decision to adopt Chromium is perceived as reducing competition, leaving Chromium with a larger market share. There is a fear that it will become harder for other browser vendors to compete with Chromium. The browsers with less market share may not invest as much in making their browser better, leading to it becoming outdated and causing compatibility issues.

*"Browser compatibility...might lose importance because browsers are getting so much better...the further we go the less time I have to think about it because I know that everybody's competing..now that Microsoft decided to stop developing their engine I'm worried that Chromium is going to get too much market share and that is not going to help with browser compatibility."*

*"I think [people moving to Chromium is] a good thing as a designer... but as someone interested in the further development of technology, it's not the best thing because considering everyone will start using Chromium, there won't be a competition left. If there is no competition, the pace of development...decreases...there should always be some competition. I think Microsoft declared that they're shifting to Chromium as well, so if they do that competition will decrease, it will be hard for Mozilla to sustain with their engine. It's a good thing in terms of users or developers, but in terms of technological development of that particular set of a browser or how a browser evolves, it's something bad."*

*"...I would love to see Mozilla and Chrome competing with each other to improve the web, I see that the two are basically trying to improve the standard, I like it very much that there is no fixed standard anymore."*

## Knowing How Things Work

Web developers understand the Web, and the surrounding ecosystem is continually changing. With that comes pros and cons. Considering, in most cases, it is their livelihood; there is a strong desire to maintain some semblance of control and understanding of their work. One way developers do this is to invest time in understanding how things work — beyond the abstractions that modern conveniences such as frameworks present. Understanding the basics boosts developers' confidence that they can handle something when issues arise.

*"I think [frameworks] are not that important because the fundamentals about how things work are more important than the specific frameworks, [but] for the web in general, and other developers...they allow so many more things to be done for the web."*

*"A basic knowledge of programming is really important because you will use basics even [as a] senior...it's really easy to make a basic mistake...it takes a lot of time to build the logic of basic programming knowledge."*

*"If I am writing code for myself, I generally don't use frameworks. I just go hardcore...[so] I know how it works, and then when I go and use it in a framework I get what the framework is doing. Whereas I think a lot of people coming in to front end, they learn the framework and they don't know the JavaScript that's going on underneath, which probably gets you by 80% of the time but then when something scary happens, it's like, 'Why did that happen? This is magic.' I did the same where I jumped into Angular and I'm like, 'No, this is way too much magic. I need to know what's actually going on.'"*

*"...in terms of obscuring what's actually happening...frequently, you don't need to know it's enough to say, 'Oh, in React land I'm passing this state property as a prop down to this child component,' and that's good enough, but what's actually happening underneath the hood..."*

*when it is time to do that digging, it can be really unclear, messy if you spent all your time just living in frameworks land..."*

*"If I learn something like basic JavaScript, then I think I can switch to different frameworks."*

## Experienced Developers Recognize Improvements

Much of this report's focus is on the needs of developers and what causes frustrations. These frustrations are representative across many years of experience from a beginner (less than a year) to seasoned (ten or more years). Developers who fall into, or near, the seasoned camp recognize that things on the Web are better, compared to when they first started. Improvements run the gamut from tooling, deployment, debugging, layout, privacy, ES6, etc.

*"I think we have really great tools now to do web development compared to many years ago. The tools now, I mean, they can always be better, but it seems like we have really good tools [they] weren't so common 8-10 years ago."*

*"Testing is hugely important for making sure that things work as expected. There's a bunch of really great tools available to help with that now, which I find super helpful...It used to be super frustrating, but the tools are getting better and better."*

*"Deployment is really important, obviously, because you need to be able to put your app out there but it has become so much easier as well. Now we have services like Heroku where in just two clicks, you can deploy your app, like if it was magic. I remember in my first year we have our deploy and build scripts, they were written in Bash..."*

*"The new Grid and Flexbox [are] so good. And Firefox in particular, has really amazing support for being able to debug those layouts and see how they work...the grid Inspector, it's so good, so helpful because you can actually see what's going on. It's like, 'This is where the grid is, this is where the gaps are, this is why that grid is over there.' They've got the Flexbox inspector as well. It's like, 'Why are my things all over on the left? Oh, because this is what's being set, this space is empty gap, or this space is actually an element that turns out to be way bigger than I thought it was.' Getting people at work to move to that way of doing things instead of floats left, it's been fun."*

*"When functions first came out, they were impossible to debug in the dev tools because, I don't know what was going on it was just really hard. Whereas now the debugging tools have gotten way better and it's way easier to do that kind of thing. As JavaScript got more asynchronous debugging, got a lot harder. And now it's getting easier again, so that's been great...those tools that are in the dev tools have been amazing to help with debugging."*

*"I think there are great tools for debugging...I use the debugger in Chrome, mostly for JavaScript. For server side development I use the Ruby debugger. I also use logs. And just in general, the web console in Chrome as well, and the Network tab. All the developer tools they help with their debugging a lot."*

*"I think layout has become much less of a pain point. Layout has been more difficult in the past, designing all these huge tables to have to achieve a certain layout...I think performance has improved dramatically, I think the tools have improved dramatically, too. I think the pace of change is also higher than ever before, I think debugging has been harder in the past. I think all these tools have improved tremendously."*

*"Firefox or Safari implemented a really good Do Not Track engine. Safari doesn't let me know exactly which MacBook you are using, it's just going to look like a generic one. They do have special ways of changing the tracking cookie so it's not going to always be the same, and that's something that they just improved. And now Safari and Firefox are using the restrict mode, so if the user wants you can prevent people from tracking what you do. I think we're going in a better place in terms of privacy, I'm just optimistic about it."*

*"I used to be more frustrated and stressed out about frameworks because there are so many and they change so much, but I'm not*



*frustrated anymore about that. Since JavaScript ES6 everything has become standard and I feel like using vanilla JavaScript is so much nicer than before. And I don't think using a specific framework, knowing a specific framework is so key [as] it was in the past."*

Keep in mind that while there are recognition and appreciation for how things have gotten better over time, it does not mean things are perfect. Some improvements are what lead to the current environment where developers are facing trade-offs when deciding how much to divest their control into external, third party products and services. That and the pace of change can be overwhelming.

## What's Missing From the Web

The same group of seasoned developers had thoughts on what is currently missing from the web. These are similar things to what was captured in survey findings and are included here to provide verbatims with more context. In addition to what's missing, this group had thoughts on opportunities for improvement.

Web components and privacy could be better executed.

*"Web components have a great opportunity, but they wasted so much time with things that...in terms of the choices that were made, didn't pay off like HTML imports. They we're not the best way to approach it because all of the things that they didn't allow in terms of good optimization."*

*"I think [how to overcome privacy frustrations] is the \$1 million question, right? Because I don't think there is this something where there is no good solution currently to it..."*

Access to hardware is something that the web is missing, as well as push notifications and serverless communication.

*"Hardware access...more device API's would be helpful. And I think, of course, also push notifications are a very important aspect. So if these two would work more reliably, or let's say would be better supported, I think I would be very helpful"*

*"This is a very personal opinion from a computer science guy...a more event-based browser to browser model, like in the web RTC environment. If I could place anything on the list, I would say, browser to browser communication, serverless communication applications. This is something I would like to see in the future so that basically you don't need a web server, just bring two people together like web RTC."*

# The Future of the Web

Not only do these seasoned developers recognize areas of the Web that have improved, could still use improvement or are missing entirely, they also had some thoughts to share about the future of the Web.

When it comes to the future of the web, these seasoned developers are excited for:

- WebAssembly
- Progressive Web Apps
- WebAuthn
- Speech to Text

*"WebAssembly is going to change the web. This guy is an insane level of importance. It's going to change the web from being a mostly consumption light editing environment to a fully-featured editing environment. It provides an increase in speed that is unheard of for the web."*

*"I think if you allow web applications to be more like native apps, then you can basically replace many of these native apps with web pages, which gives them a simple deployment model and development model for web developers."*

*"I just saw, now, they finally decided on the spec of WebAuthn. So once browsers [are] actually going to implement it is going to be much easier and safer for users to authenticate...that's probably me hoping for a better future."*

*"I would like to see more is text to speech and speech to text...I think this is a very important topic for the future of the web...I'm not sure if, in a few years, we will still use graphical user interfaces to interact... And I think that web applications should also be able to run headless, without having or without being forced to have a user interface, graphical user interface as their main interface towards the user. I think speech is something that should get more attention..."*

Though there are things to look forward to, there is a concern, and it's not clear how widely this concern is shared. The concern is that the next generation of web developers might not see the importance of learning the fundamentals of computer science as it's so easy to just run with examples.

*"My hardest [job as a teacher], what I'm trying to do...is to convince [my students] not to go to W3schools.com and just do an example-driven web development. My students typically think they can Google every solution, while I try to explain to them that they should first use their brain before they do something. I think they would tell me that the web is some copy and paste place and they would throw all these computer science rules overboard and just try to copy from existing projects."*