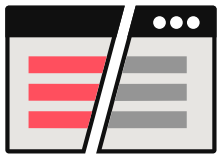
Pilot Interview Findings

Enhanced View of the Top Ten Needs

Browser Compatibility



In the full list of needs browser compatibility was the dominant theme. Four need statements made the top ten:

#1 Having to support specific browsers (e.g., IE11).
#3 Avoiding or removing a feature that doesn't work across browsers.
#4 Testing across browsers.
#5 Making a design look/work the same across browsers.

Of all the pilot interview findings, browser compatibility was the topic discussed using strong language especially if IE was at the heart of an issue.

While IE11 is decreasing in share as Microsoft shifts users to Edge (and the forthcoming Chromium-based version), it continues to be a large source of frustration to developers due to being over-represented in enterprises, legacy line of business apps, and certain industries. The prevalence of users accessing the web through IE is beyond both web developers' control and Microsoft's. In some cases, it's the management of a company that dictates what browsers should be supported, ignoring data that suggests users are accessing a site through browsers other than those mandated.

> *"I don't know why people still use IE. I find [browser compatibility] very frustrating, especially as a front end designer. IE is such a curse my god, why do people use it?"*

> *"Browser compatibility, by and large, shouldn't be an issue and for us specifically is a nightmare...we have so many people accessing our application from their office, which means they are running IE11. Which is something that I don't care about, except I needed to code for it to keep my job. It's just a hair-pulling frustration...it is tragic."*

> *"[Browser compatibility] is important, but I think people put importance on the wrong parts of it. Everywhere I've worked they say, 'It has to work on IE11,' and I'm like, 'Really because I think it needs to work on mobile Safari but not IE11.' It's frustrating trying to get stuff working on IE...the manufacturers have said stop using that."*

> *"...the most annoying thing for me is...browser compatibility, to feel like something is simple enough to where, it shouldn't render differently in Edge, but it does. That's the big one, especially because [in my industry of] oil and gas, most people aren't very technically inclined..."*

> *"... this [doesn't] strictly [relate] to accessibility it more relates to browser compatibility. But Edge is not particularly accessible. It's optimization for JAWS, which is the only popular software for professional blind users is abysmal...it's recommended that Edge is not used at all, which makes users divert or revert to older versions of IE, which is a mess."*

Browser compatibility is getting better though, at least some see a light at the end of the tunnel.

> *"There are not many problems we have with browser compatibility... every once in a while, we're supporting IE11, and that's the only old browser we're supporting. Even that we're noticing that less than half a percent of our users are using IE 11..."*

> *"If you think about the amount of weird tricks [you had] to pull off ten years ago to make sure that your website looks good in every web browser, and the amount of things you have to do now is kindergarten level. You just have to do a few things every browser now supports a decent enough level of features."*

While it might seem easy to point fingers at IE, all browsers have

their quirks, and developers have had to learn those quirks and adapt to make their site work well enough after tediously testing across browsers and, in some cases, having to remove a feature.

> *"...it's hard to achieve cross browser compatibility so everything works...it's a lot of testing...to do with all these browsers, which [isn't] a lot of fun..."*

> *"Modern browsers are supposed to support same-site cookie attribute. But it turns out the latest version [of Safari] has a bug...it not being compatible with this specific version of Safari means that we cannot use them in general...all this is really frustrating."*

> *"...you have to make sure that whatever you develop works as expected. There are fringe cases, say, if I were to make a web form [and it] works fine and dandy in Chrome, but for whatever reason, maybe it's not behaving in another browser. Then you go down the rabbit hole of trying to debug or trying to research what the root cause is. Sometimes it's due to an unsupported API or feature that's being used, and that being the hallmark feature of the app..."*

> *"...I understand the need to cater to browser compatibility, for example, you can pull up the metrics of any given project or site, and then you can look at the technology or the browser that's being used to visit your site...you can make a case to say, 'Well, a lot of our users are using a specific browser, we need to ensure that our project looks [in] the best light. That's what can withhold certain features that you implement on your project."*

Though not captured as a need statement by itself, at the crux of browser compatibility is the users' experience. If it doesn't work across the browsers your users use, then they aren't going to be accessing your site, which has business implications.

> *"I think it's very important that your audience can use your product no matter what device or browser they're using, they should be able to use your service. I think this is very important to be able to deploy your product across a range of browsers."*

> *"[Browser compatibility is] an implicit point where it is so important because everyone should have the same experience."*

Documentation

Survey participants rated aspects of documentation as the second most frustrating need:

#2 Outdated or inaccurate documentation for frameworks and libraries.

Documentation is so important, and hence leads to such high frustration because it is one of the things that gives developers an accurate picture of how things should work. It is also one of their go-to resources for overcoming frustration, which is why developers find inaccurate or outdated documentation so frustrating.

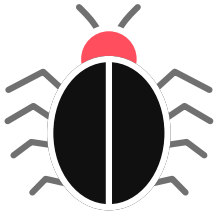
> *"...[documentation] really gives you the right picture of how things should be working...if you aren't sure how things are supposed to work it's so hard to get them right or it's so hard to identify possible issues with what you're doing."*

> *"Documentation is [the central thing for helping me overcome frustration] partly because...you think you knew something but then after reading up on said feature or said documentation, it's like, 'Ah, I should be implementing it this way,' rather than a learned know-how, so to speak."*

Documentation was one of the reasons why developers dislike frameworks. They find there is not enough information about how to use a framework optimally.

> *"...nobody tells us what is the optimal way to use [a framework]. It took me a lot of time to understand how to format my bootstrap file...this is the tree I have to make to make sure that it is optimal and it can be used by multiple different devices. That is something nobody tells us, exactly...That's something I really like about Vue.JS they tell how to use it optimally."*

Debugging



In the full list of needs, debugging debuted at #6:

#6 Discovering bugs not caught during testing.

The frustrations developers experience with debugging are more about the time it takes and the fact that if you're debugging, then something is wrong. Trying to find the bug and how it manifests in a browser is also a source of frustration.

> *"Debugging is always irritating...[It's] a never-ending process. I can have a hundred test cases, I can debug [them all]. I'll come back, one user will call, 'This particular thing is not working,' so I have to go back again, and realize, 'Okay, this is again a problem because [of] something so debugging is frustrating."*

> *"[The] level of frustration [for debugging] would be high up there... debugging a layout it isn't so much of a mystery if it's HTML and CSS...in the context of JavaScript I feel like you're always debugging throughout the process, making sure that it hits certain breakpoints. If I hit all the marks...for me, then I'm satisfied. Debugging should definitely be more along the lines of ops [and] code versioning because I feel like that's where a lot of that happens in tandem."*

> *"...the only reason that [debugging is] frustrating is because of the multiple places to go to do it when you're actually having to confirm how it appears in a browser."*

> *"Debugging is just frustrating because if you are already debugging, you've done something wrong. You shouldn't write to the point where you debug, you should just test so that you don't waste time debugging...it's important because if you are debugging that means there's something that's not working right that's bothering you. Hopefully, you catch it before it's live."*

Frameworks

One of the unknowns stakeholders had about web developers is why they like frameworks — what about frameworks do they like? It turns out, the relationship with frameworks is more of a love/hate meaning developers don't necessarily like frameworks. In the survey, frameworks made the cut in two of the top ten needs:

#7 Supporting multiple frameworks in the same code base.
#8 Keeping up with a large number of new and existing tools or frameworks.

Some developers envy the server-side world where there's more stability and only a few leading frameworks, whereas, on the front end side, developers feel they are getting whiplash just trying to keep up with the many frameworks that exist.

> *"I think it's frustrating that there's no good way to manage your dependencies of applications on the web, compared to Maven in the Java world, which has been a long, steady, common tool which has been available for many years."*

> *"It could very well [be]...frameworks are entering a maturity where you'll have a few, React and Angular, in perpetuity, the same way that the server-side frameworks have settled down into three or four that everyone uses. But it's not there yet, and it's a lot of whiplashing and eye-rolling...but it also keeps me employed."*

Some developers try to avoid frameworks altogether mainly because it's too hard to keep up with new and existing frameworks.

> *"...the pace of change in the framework arena has a high level of frustration, so I try to avoid them."*

> *"I'm not a big fan of frameworks... I don't feel like they get you that much...[frameworks] always look more complicated than they ought to be [but] they're presented [as] the greatest thing since sliced bread. I'm not convinced."*

On the flipside, frameworks offer efficiencies allowing developers to reuse something that's already been solved for.

> *"I use the term framework to mean a new set of modules that have been prebuilt to trivialize the tasks...if there's a complex series of blocks of code that you need...in order to get around a certain issue of rendering something or processing something, then somebody is going to create a library to make that more efficient."*

> *"React is a joy to use, Angular is a very solid one. You can pick and choose whatever you want now. Each one of the big ones, it's really well made...allows you to do whatever you need. [Frameworks are] important because it does allow many companies to create something good with not much effort...it's to the point where it's nice, but it's somewhat in the background because everybody got so good at it."*

Participants mentioned browser-based APIs, web components and object-oriented JavaScript as potential solutions or methods for reducing the need for frameworks.

> *"...in the last few years, the browser's did enormous development...all these browser APIs are good enough in order to avoid specific platforms. What I've learned from the last 10-15 years is that every, let's say, half a year, some kind of framework is hyped. People tell you, 'It's the best that they've ever seen.' Half a year later, something else comes around, and you have to change your code base. The most reliable project I've seen that works over many years, just purely use browser based API...I resorted to using pure browser, cross browser supported*

APIs...they work for many years"

"...a lot of the problems that frameworks are trying to solve are artifacts of the non-object-oriented JavaScript, the older JavaScript way of doing things... it's my hypothesis that a lot of that goes away if you just take a more modern design approach to developing for the web,... I've been playing with web sockets...it's just a shared object. That's one way of doing it. [Another way is] with remote method indication."

"I believe in web components. It will be nice if [frameworks] are not going to be as important as before."

Privacy

In the full list of needs tested in the survey, there were two need statements that related to privacy. One made the top ten list, coming in ninth:

#9 Managing user data to comply with laws and regulations.

Depending on the part of the world a developer resides, privacy is seen as getting better (GDPR) or worse (Australian Assistance and Access Bill). These laws and regulations have implications on both protecting a developer's privacy as a user of the web, how they approach their work, as well as their career trajectory. When laws and regulations were mentioned in conjunction with privacy, there was a belief that maintaining privacy, whether their own or users, is beyond the scope of technology, meaning Governments need to step in to hold technology companies accountable for their use of private data. However, in some countries, it can be the Government who is exploiting data.

> *"I saw [privacy] more in terms of [the] end customer instead of me [the] developer... me as a user I'm feeling much safer now, in terms of privacy than a year and a half ago because...if you're in Europe, GDPR is going to keep your data in a much better place."*

> *"It feels like many services and apps...are designed to mislead people that don't know about the technicalities. I think [privacy] is super important and it's frustrating that it's hard to protect users, but I think this should improve with regulations...with laws like GDPR from Europe. I mean, it's the only way really. Tech solutions like the Do Not Track setting in your browser...I'm sure people wanting to track you [will] exploit your data and everything. They will always find ways unless it's really regulated. I don't think that technology alone can protect people."*

> *"...the impact [of GDPR]...we have to do the work to ensure that when people ask for us to delete the data, we need to ensure that we are deleting all the data that we have from them...that means deleting anything for logs, making sure that everything is deleted from all databases...before this we were still doing that kind of work but we didn't have the same pressure I suppose...and it wasn't clear for users that they had the right to exercise that..."*

> *"privacy...that depends on the company where you work honestly. In Europe, [privacy is] much easier...because you have to specify everything you do and they have to be able to delete everything. GDPR helps so much."*

> *"[Privacy] that's a big one...we just pass the Assistance and Access Bill, a huge blow to the Australian security and IT industry...you can be instructed to make changes to your app so that the government can get access to it, and you're not allowed to tell anyone. It's at the forefront of everyone's minds here...if I were to go look for another job, there's plenty of companies who said they don't want to hire devs in Australia anymore because we're suddenly a security risk."*

Beyond the laws and regulations surrounding privacy, the pilot interviews revealed that privacy is personal. One thing top of mind when discussing privacy was protecting their privacy as a user of the Web, followed by the role of privacy in their work. When it came to protecting their own privacy and the privacy of their users, developers felt it is hard to do. There is a belief that maintaining their privacy is their right, and it is something they value, so they try to keep their users' privacy safe too. Maintaining privacy comes with tradeoffs in terms of the features you're able to access.

> *"You're asking me why privacy matters? It's my right."*

"When I use the web, I feel uncomfortable. I ask myself, 'Is this secure enough or will I have to use incognito version?' I'm not sure how much I can protect my information and security on the web...what is important for me is that I keep these values in doing my development."

"...it's quite frustrating to protect my privacy on the web, but it's really important for [users] to keep their privacy safe. I think most developers should consider this when they build a website. As a consumer, I think it's really important to keep my privacy safe. I don't want to give my specifics or I don't want to receive any ads based on my activity on the web."

"...being a German [privacy is] very important to us...I think this is the hardest part to achieve typically because you want to have all these cool features, want to use all these platforms, so it's hard to achieve privacy, but still, at the same time it's very important to us...this is one of the most difficult issues and that's why I'm using Firefox and not Chrome for instance today."

"...I'm thinking that mostly as a user...you install an app and it wants permissions. I'm always like, 'Why does it need that, it doesn't need that,' you don't want to read [the privacy policy] you just click okay...it's a problem...you can't trust these apps...they can access all your data and all with any webpage you visited...I find that very frustrating as a user it ought to be more precise about exactly what it's going to do and what it won't do...it's just creepy."

Security

Rounding out the top ten needs is security, coming in at tenth:

#10 Understanding and implementing security measures.

What did we learn about security from the pilot interviews that lead to this need? First, let's look at how people defined security. Developers used terms such as authentication and privacy in their definitions.

> *"Security means that no one can get access to any part of the system that they are not authorized to access, and no one person can see any data from the system that they're not allowed to access."*

> *"[When I think about security] I think about authentication...how do you successfully guard against malicious login or malicious attacks on things that are supposed to be secure."*

> *"...[a site] is considered secure [if] it's working in the way you intended, respecting all the access rules...when your users are aware of what's happening with the data, [and] when the data users submit is only used for what they intended [it] to [be used for]."*

> *"Security to me is like the authenticity of data, authentication of users, and being sure that the other guy I'm exchanging data with is the guy he pretends to be...it's very important to have trust so that your users know that they receive data from you and send data to me. I think authentication here is key."*

> *"Privacy is about keeping people's data safe whereas security can have further implications like your app being brought down or somebody hacking it [so it] no longer does what you're expecting it to do. Like proper password protection and stuff to stop people's accounts from being able to get hacked, and then that information being used for other stuff...so privacy is a part of security."*

Beyond the definitions of security, the main thing we heard is that web developers don't know enough about security which creates a feeling of concern and frustration that someone with more knowledge can find and exploit security flaws. This feeling causes stress, especially because security is viewed as very important.

> *"Security is one of those things that I'm sure we should be doing better on, we haven't had any issues yet, but all you need is one issue..."*

> *"I always have concerns that there are security issues that I haven't figured out yet. I've thought very hard about whether or not it's possible for malicious third parties to authenticate as someone else...I haven't figured [it] out yet but I feel like I'm not quite smart enough in this area, to know whether or not that's possible."*

> *"...[with security] I always have this fear...I use JWT to authenticate my request after initiating. I always have that doubt that it's not unbreakable...a good hacker might be doing that...I think there are a lot of things to be learned and steps to be followed. I have not done that. Maybe it's my ignorance on that side."*

> *"Security is very important for any of our customers at any time for anyone to be honest...you have to make sure that [user] data is contained with us...applying security it's very frustrating because I myself am not very good at security and I personally feel that there are not enough online resources that actually explain why your security might be flawed or what are you doing wrong that might lead to flaw in security. Mostly, I don't really understand security."*

We also heard that the ways to implement security feel like a hack

in and of themselves, which is a result of the evolution of the web. As security became important, new ways of ensuring security evolved out of necessity, but those were reactionary developments.

> *"Security is so important...having bad security can be catastrophic for your business and for individuals. It's one of the most important things right now...every other day there is some breach and a lot of user data is exposed. It's so bad...it's so frustrating because it's so hard to get it right. There are so many things on the web that are broken from the beginning because the way that the web has evolved...it's really hard to get it right. Many of the things that we've used in the past for security to protect things like tokens...they feel like a patch or a hack on top of the whole system...feeling that it's impossible to get your app to be truly secure [because] that doesn't exist. There will always be vulnerabilities in everything in protocols and fundamental libraries...it's so hard and so important at the same time."*

The third thing we heard about security is a two-way relationship with users. If you get security wrong it adversely affects users. However, you may implement the best security features you know how to, but if users access a site through passwords, as a web developer you have very little control over whether or not they choose a secure password.

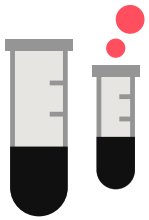
> *"Security is essential...it requires so much attention, and it can be frustrating...because it's something that you don't want to get wrong. It's going to stress you out because getting it wrong might be a really bad thing for your users."*

> *"...I'm trying to make sure that people [use] better passwords for our system...we give them logins for the production server and also a staging server...'123' is their password. I have to make sure that they change that but sometimes I don't know what password they're using because it's their password, I'm not supposed to know it. What we find is that we're only as secure as the passwords that our users use."*

As you can see with the perception that security feels like a hack and being tied to customer logins, security is closely related to needs or frustrations that might arise due to the pace of change of the web and authentication.

Enhanced View of the Remaining Needs

Testing



During the kickoff meeting for this project, stakeholders shared their thoughts on what was already known about testing. What was captured is that automated testing is hard to get started and there are many different types of testing like unit, integration, and functional. What was unknown was the types of tests developers are running and how or if that causes frustration.

The pilot interviews revealed that some developers feel that it's frustrating to keep track of all the available tools to conduct a proper test and automated tests are not always an appropriate method for the web.

> *"The frustrating part [about testing is] not being aware of the tools available to conduct a proper test."*

> *"The web is not an automated test driven place because there's too much interdependency between JavaScript, HTML and CSS. I didn't really see the benefit of doing automated testing here...typically you react to user events and then change something on the UI...I think that is better tested by a human...while back end code that interacts with databases, servers, and stuff like this, where the logic is pretty clear that's easily testable using unit testing or something..."*

> *"I don't automate my test normally like you're supposed to...I like it to run on both the server side and the client side, and I can test it interactively. It's just not always possible to do things to automate the test... it's not always clear how to do that, and how to check the result...if I can play around with it, that's the way I like to test things. And it can be frustrating, it's always frustrating when things don't work."*

Participants felt testing can be frustrating because it can take a significant amount of time to write certain tests and the ease of writing those tests is related to a developers' experience with testing.

> *"A couple of reasons come to my mind immediately [about why testing is frustrating]. The front end testing it's not always easy to do unit testing. That's what I have felt. I have not written extensive test cases before...I know the importance of test driven development and we should write code for testing, and then only write the application so that it won't fail, and those kind of things. Also, I think [it's frustrating] because I haven't done it that much. I always used to manually test my code..."*

> *"Some projects you might not have time to do the unit tests, there is a time factor involved where you don't have the time to write first the test cases, then you write the code that is failing, and then you have to make it pass. We don't have that kind of time for the project, because it's a quickly moving product."*

> *"We also do system tests that are harder to write [than unit or integration tests] because there are more things to consider [and more] time consuming to write when considering all the interactions with the browser. That's why testing is not zero frustration."*

Opinions did differ, slightly. On the flipside of not having the time to test, is the belief that taking the time to run tests can reduce stress.

"Wow, if you're a developer, [testing is] your bread and butter. It shouldn't be stressful to test, it should be something that you want to do. Because you know this is going to actually reduce stress."

The types of tests developers run varies as do the tools they use to test.

> *"[I do] unit testing, but it's component level testing. A single compo-*

nent testing TypeScript and HTML together...we used to test JavaScript, whereas now we try to test the whole component together...the back end devs do some kind of back end testing. We'll run automated end to end tests, you login and go through a flow on a page. We were using Protractor and Selenium. Now we're using Test Cafe and we do a heap of manual testing. We have as many testers as we have devs, and they do all of the cross browser testing, cross device testing, all that kind of stuff."

"[I run] as many tests as possible. I'm a true believer of the green-red testing, then integration tests to make sure things work, and then use Cypress or Puppeteer...maybe your QA is going to use Selenium...you'll have to cover the critical path of your website, at least with end to end testing...there's so much choice...you can test it by checking the values, you can abstract other things in a snapshot testing. As long as you are able to check your volume on your logic, and then check that your components are well structured, you should be in a good position to iterate quickly without worrying too much about what could break."

"...on the testing side, all of the angular testing tools they just flatten out all of your asynchronous code, and magically make it synchronous so that you can understand it and follow the flow more easily."

Authentication

Authentication was often described as difficult and frustrating for a variety of reasons, but mainly due to the complicated nature of it and the threat that exists if you get it wrong. However, authentication can be easier if a developer is willing to divest control to a third party.

"It's hard to write a secure API for mobile apps to authenticate against. There are flows but they are very convoluted. [It's] not nice for users and not nice for developers to implement. The whole authentication mechanism with passwords and everything that feels like it should be much easier, especially with mobile where you have other ways to authenticate...like using your fingerprint or using your face. I hope it improves with standards, but it still feels very frustrating in the sense that is so hard to get it right."

"OAuth2...it's a complicated sort of thing...the thing that bothered me about it more than anything was that [logging in with Google or Facebook] even though they use the same app, the code that was used [was] messy. Even though they were using the same protocol, they weren't really the same, they were just really different...I started downloading...a piece of suggested code to run on the server...even though they're using the same protocol they were so different as far as actually setting them up on my server and how to handle it...it was just confusing, it all [should] be simpler than that."

"I think the whole the whole notion of authentication and authorization it's just fraught with difficulty...you've got people out there all the time, they're trying to screw with it, trying to break it. It's difficult and frustrating and important."

"Authentication is generally a nightmare...I'm not that comfortable when there's opaque layers of abstraction, and authentication very rapidly becomes like, 'Oh, there's this encrypt hashing algorithm,' I don't know what that is and frankly, neither do any of my buddies who actually got bachelor's degrees in computer science. It rapidly becomes impenetrable save for certain people. And that's even setting aside the issue of, 'Are you storing this as a cookie, or are you storing this as a JSON Web Token?' Well, that's a new technology that has a ton of pointy edges. And now the tide is moving against JSON Web tokens..."

"Setting up authentication is either a bit of a nightmare if you're doing it yourself, or it's incredibly simple, if you're willing to just divest yourself of control over everything, just a leitmotif, a theme in web development."

Performance

Performance was not often mentioned as one of the top items that causes frustration, however, it is something that developers find important. It's important because it directly impacts the users' experience on a site. Performance also ties back to a broad view of accessibility where a site should perform across a range of devices.

> *"Performance can affect a lot of users' experiences [and whether] they stay on the page...they need to see something...with information as soon as they land. It's one of the things to keep the user on the site or the application."*

> *"Performance that should be your first things together with [privacy, testing, tools, and accessibility] because you want the user to be able to reach your website anywhere, no matter the connection no matter the device."*

> *"[Performance] is closely tied to accessibility in a lot of ways, which makes it quite important. If your app is huge and slow then it's annoying for someone on a brand new MacBook, but it's unusable for someone on a low end phone, and wasting people's bandwidth and download and stuff. So it's important in that respect..."*

> *"Performance in terms of user experience, I think it's very important to ensure that your app or your project is performing the way [your users] expect it to..."*

> *"...the page should load quickly and shouldn't keep the user waiting."*

Developers referenced performance as a source of frustration because it's hard to find the time to make a site perform well on low powered devices or difficult to pinpoint what is causing the performance issue. This fits with the survey results where, "Pinpointing existing performance issues," ranked as the twelfth need, out of 28. Another reason performance was mentioned as a source of frustration is the nature of the business itself. Businesses that experience rapid growth in a short amount of time strain systems and businesses must adapt.

> *"...it's hard to find the thing that makes your application slow. And it's hard to debug, sometimes how or what affects performance. We have to isolate everything and try to iteratively find out what's degraded performance."*

> *"I don't feel that frustrated with performance in general, I think that are a lot of tools both server side and client side that help...in many cases it's just hard to find the time to do the work and to make sure that your app works on not as powerful devices."*

> *"We're doing a lot better. Six months ago, we were doing a lot worse with performance. The problem was that, [and] this is a good problem to have, we were constantly getting new [users]. We're probably about 100,000 unique [users] every month [whereas] four months ago, we're probably about 30,000. We've tripled in the last three months. That has put a lot of strain on our systems. We've had to work hard to... parallelize our computation."*

Developers referenced many server side and client side tools that help them resolve performance issues. Most often referenced were network debugging tools, developer consoles, Lighthouse, and New Relic tools.

> *"...with my current app that I'm working on, the performance is so bad that I just run it through Lighthouse, and that's enough to say, 'Hey, it's taking eight seconds to load or 20 seconds to load sometimes... there's some pretty obvious things that we can do.' In my previous*

place, we got beyond that, where we needed to look more into the details of what we could work on, again Lighthouses was mostly what we used because it does give you a breakdown of stuff like that. There's other Google tools that we've been using that I do not recall the name of...we've got monitoring and stuff that I can look at. Everyday, I get warnings, 'Oh, the app is taking more than two seconds to respond...yep, that's pretty standard.' Those are driven from New Relic."

"I'll go into the developer console and look at the network tab, and see how fast things are loading, how many round trips it's making just to load the first page...you can fix by bundling stuff together into one file that loads instead of having to make twenty different trips to the server to get the pieces to put the page together...there's a lot of different little tricks that you can do to make things load more quickly."

During the kickoff, stakeholders had questions about what it means to be performant. The pilot interviews did not reveal any one measure, instead it was unique to the developers and the types of projects they're working on. Developers define their own performance metrics as needed in a way that made sense for their work.

"I measure [performance] in terms of payload, time to interaction, and amount of requests. Of course, this needs to be set to comply on rules that change from project to project."

"In my case, I [measure performance] on what is an acceptable response time...since I work mostly in the server. I just look at graphs and try to see if most requests are within acceptable response time."

"Well, of course, you will always [have] some kind of [performance] metric...sometimes it's very easy. If you have a video conversation...you need to have some very hard limits on round trip or on frames per second. I typically try to define [a] performance measure, which gives me an indication of what good actually means...I try to always measure it in some way."

"[When] we're talking about the entire system as a whole, they are different aspects of performance...from a web developers point of view, we need to make sure that [our users]...believes that, even if it takes longer than it should, [it] doesn't feel like it takes a very long time."

Localization

The frustrations developers experience with localization expand beyond the traditional notion of accounting for the visual impacts of different languages, although that can be a source of frustration. Localization can mean understanding cultures and preferences in those cultures in terms of how people formally write their names so that you can account for that when designing input fields or understanding date and time format preferences. Getting these things wrong can lead to a bad user experience. Localization is also seen as something that is not just in the realm of responsibilities for a developer, it's something developers and designers have to work on together.

> *"Most of my work has been aimed at Australian customers, [but a] related thing we did deal with at my last job was people's names and things like the idea that everyone has a first name and a family name and that they should be in that order, which is not even a little bit true...It's a localization [issue] because the people were in Australia but they were from somewhere else. It's a problem because we're doing identity matching so we need to be able to match people's names across different identities. Basically, the answer was, 'If your name isn't in a standard format, you have to go into a shop and deal with it,' which was not ideal...the people who originally designed the system were all white, Australian men...because it was built that specific way in the first place, it was hard to change it later. Whereas if they had people from more diverse backgrounds to start with, then it would have been built in from the beginning instead of tacked on later."*

> *"[I was] using an app today that wouldn't let me change the date out of American format. So I'm really frustrated about localization. That was as a user...It's so annoying, especially living in Australia. You're using stuff from overseas almost always, and if you can't localize it, it's really noticeable all the time...it's super frustrating, because it's just like, 'My God, everyone's not where you are.'"*

> *"Localization is just hard to get right. It's not just a developer issue it's something that developers and designers have to work together to manage. It's just stressful because, in many organizations, there is not a real care for it. If you do localization as a second thought your interface is going to look horrible and then you have to fix it. But you didn't start with a user interface that was enabling the ability to do it, so then you're just going to waste time."*

> *"Localization can be difficult because sometimes you have to render the page from right side to left side, it's not left to right. There are several steps involved. Typically that can [create] a little bit of frustration if you want to do localization in a language that you don't know well. For example, a person who knows Spanish can do the localization well if he knows English and Spanish. If they don't know Spanish, and if they had to do it, I think it there might be a little bit of frustration."*

Pace of Change

Developers acknowledge the web is an environment that changes, often quickly. It's a reality that they've come to accept — almost with an air of resignation.

> *"...we're pretty good about keeping on top of things...whatever happens, happens."*

> *"As a developer, I think I should just accept sometimes the reality is the way it is so... if the environment changes quite fast...no matter how fast or how slow, I need to adapt."*

> *"Trends it's important to follow them because it's nice to see where the industry is going...they're not as frustrating as I initially thought because, in the end, it's just something that is always going to change. They're going to be frustrating, or less frustrating, every few years because they're going to be different...maybe you're not going to like all of them, but they're not even something that you have to follow strictly. So if it's something you don't really like, you can still rely on something else."*

> *"Every time I try to learn something new, I always think it is more complicated than it needs to be, and I think everybody has that feeling."*

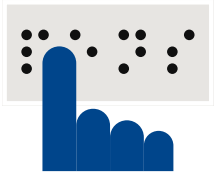
Not all change is bad, as change can lead to advancements in technology and more efficient ways of doing things but the frequency is difficult to keep up with and has implications for their work. Mainly, there's a fear that as new technologies and tools become available, it will require work to migrate over and may break things in the process.

> *"...the pace of change in the web arena is quite high. That's frustrating somehow because you have to migrate all your projects and from one to the other tool."*

> *"Pace of change is frustrating and very important especially when it comes to updates, and how fast things change and things break. Sometimes you don't know how, or why. And it's so hard to keep up, especially when you are trying to maintain a different set of apps and some of them are legacy apps...because in many cases, it means your apps break."*

> *"...frameworks that are used for web development, in large part have to do with the efficiency of doing certain tasks that are currently popular and have not been efficient before... like encryption, were the MD5 checksum was discovered to be imperfect, at some point, cryptologically speaking...anyone who used it for extremely secure purposes needed to stop, which is stressful...that had to do with more of an advancement in math, which wasn't anybody's fault...I worry that an existing technology would be obsolete artificially quickly. If it was used in a form, or if your projects were dependent on it."*

Accessibility



Similar to authentication, accessibility, developers described it as difficult and frustrating. Developers understand the steps to follow to make a site accessible, but sometimes those steps are hard to get right, and the standards leave a lot to be desired. Often, a developer might be in a situation where their organization does not prioritize accessibility because, perhaps, it hasn't come up as an issue, yet. The awareness of the need isn't there. In these cases, accessibility is seen as less important than the time and effort required to implement it correctly.

> *"Accessibility this is so important...it's frustrating because it's hard to get it right. I don't think most web developers, myself included, put enough effort to make our apps accessible because it's hard. You need to use these tools that simulate how it will be to use our app [on] a screen reader. The tools are there, we just don't use them. I don't think it's fair to say I'm frustrated with accessibility because it's just a matter of putting in the work to get it right. But it's definitely super important because there's so many things when you are developing a web app you just assume people are going to use it in the way you use it...when you don't see the problem yourself, you don't think of fixing it. So maybe this awareness what is missing in general?"*

> *"Our application is not as accessible as it should be. I'm going to put [accessibility as] a low level of importance, not because it really should be that low but because we haven't prioritized it...we sell our software to companies that provide our software in an iframe to their customers, and they've never once mentioned accessibility to us. In my previous company accessibility was very important, we made sure that everything accessible. But my current company, it's it hasn't once been mentioned, and I'm keeping in the back of my head that at some point, we're going to need to work on this and make sure that our software is accessible, but it has not been mentioned by one of our customers."*

> *"[Accessibility] not that I want it to be an afterthought, but it's the byproduct of the teams that I've been in. I do recognize how important it is, but it's just not something that's heavily invested upon, I think."*

> *"I would say the level of frustration that [accessibility] generates is greater than its level of importance, and it is important. Without minimizing its importance, it is important, you should be able to access web technologies, regardless of whether you have a visual, physical, or cognitive impairment. Currently, ARIA and web accessibility specs are a mess. We get accessibility audits that will be contradictory between each other. It is important and the way it is currently implemented... and the way the specs are interpreted (or not interpreted) and are clear (or not clear) is just a nightmare."*