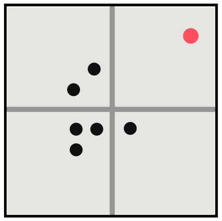
Outliers

Some of the pilot interview findings shed light on needs that are outliers, which were not tested in the survey. We include them here as they are things that some web developers have to wrestle with, and that can cause frustration.

Ops



When referring to Ops, we mean anything related to creating the build and deploying it.

The audience who participated in the pilot interviews found Ops to be a source of frustration due to a lack of familiarity and practice. Most of the participants were not directly responsible for Ops or their projects don't warrant that level of complexity.

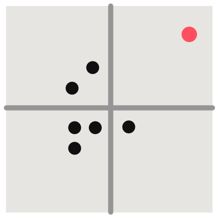
> *"I recognize that, for example, Ops, if I were more familiar or if I had enough repetition to get familiar then I could see that going down [in frustration levels] I think part of it is investing the time to get familiar with whatever the stack [is] or [the] core technologies that someone might be working with..."*

> *"I started deploying on Heroku nowadays, I'm not very familiar with the continuous integration and deployment. I guess, maybe that's why this is a problem...I don't really understand the systems oftentimes."*

> *"...deployment, I wish I knew more...I haven't had much opportunity to work with them and thus find them unclear. I don't know what a coop cluster is. I have the most basic familiarity with docker images. We run a couple things on docker containers...I would say they're incredibly important, at least to my job's codebase..I put [my] frustration [with it as] lower...because I haven't worked with them much. They just are opaque to me or unknowns."*

> *...if your Ops are fine [then] handling many, many deployments is not going to be too stressful because if your operations are fine, deploying a lot...it's just not stressful, it's just something you have to get comfortable doing."*

Tools



Tools is a broad subject with ties to other areas of web development such as debugging, testing, and browser compatibility. Developers are frustrated by the pace of change in the tools arena. It is difficult to keep up with what exists. Even if they feel up to date on their options, deciding which tool to use can be a chore.

> *"The situation that frustrated me about tools is more just me not knowing what's out there..."*

> *"An amazing number of [tools] are available, like tools to minimize images, to clean up the code, or to modify the code...there's so many of them and they're incredibly useful, but sometimes it can be difficult to decide which one to use."*

> *The reasons for choosing tools are whether or not something is supported, commonly used, or easy to use. That latter reason is a way in which developers measure how good a tool is because they don't want to spend all their time learning how to use a tool. Commonly used tools offer developers a level of assurance that it will likely continue to be supported into the future.*

> *"...before we start using any open source project, we carefully evaluate whether or not we think that this is a viable and supported project, or else this will just become more technical debt for us to handle in the future."*

> *"We try not to add lots of little libraries because we try not to have too many dependencies in our system. That tends to be where we have problems going forward. Like, we've integrated certain libraries into our system or use certain tools that are no longer supported. So we either have to start supporting them ourselves or abandon them."*

> *"...in terms of tools there seems to be new tools that come out. I err on the side of whichever is common...that's the tool that I should be well-versed [in]... if I were to look up how to do something it has to go beyond what the documentation is willing to provide because, you'd have, for lack of a better term, superstar web devs, 'Oh, they use this particular editor,' if they use that then you know you have resources that you can go to."*

> *"... the frustration [with tools] is the ramp up time to be brought up to speed and the level of importance is because the tools that you interface with you want to spend most of your time getting to the end point rather than learning the tool."*

> *"I would choose a framework if it has a lot of libraries and tools that I can instantly use...and if it's already tested, if it's been there for some time. I would go with an old framework instead of new framework that is still developing...If there is a framework that's been in the market for more than five years, I think that will have more tools, more testing tools, they are time tested, but a project that is still developing may not have [that]."*

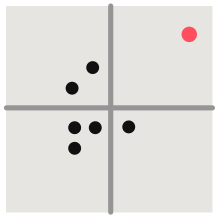
Developers do their best to avoid tools that have quirks, bugs, or perhaps don't run on Linux (if that's their preferred operating system).

> *"I've been using Linux for a long time on the desktop and on servers. I'm used to things not working out of the box and [even] so the things that [still] frustrate me are [the tools] that don't work on Linux."*

Obviously, [tools are] important because I'm using them all day, every day. They can be a bit frustrating because often the tools can't keep up with the rate of change. Also, I find that they can be a bit flaky, stuff that just doesn't quite work properly, which I think is partly just because getting things to work properly in JavaScript is hard. Little issues, like DevTools in Chrome, for ages you couldn't resize the windows properly and it just seems to always be, I guess because they're not selling the tools...they seem to break a lot.

> *"The rest of the DevTools in the browsers are the most important thing I need, and they seem to be flaky a bit...little user interface issues that just make it frustrating...They're definitely getting better and it feels bitchy to complain about them when I'm getting them for free. The browser manufacturers are companies that are trying to make money just like everyone else, and the browser is in a market. I guess the DevTools tools are there to encourage developers to develop for their platform a bit more, but it still feels like it's a nice thing they're doing for us, rather than a core thing...I probably should expect them to do it if they want me to their platform."*

Working With Others



Some of the pilot interview participants work on a team with others, which can cause frustrations. This is not unique to web developers, but how it impacts them is that it can create a more complex environment for managing code, and it can also dictate project requirements. There are also perceptions of different types of developers which color team dynamics. Most of these came about during deployment.

> *“Frameworks are the byproduct of what management or senior developers...dictate...I work with developers that are not willing to make a change or try different frameworks, because why [not] go with something that’s tested?...that’s when you start to get into red tape...I was trying to pitch possibly migrating to React...back in 2016...long story short, it wasn’t something legal was okay with due to licensing. That in itself is frustrating, but there isn’t really anything I could do about that“*

> *“...in the company it is understood that [code architecture] requires time and that by just going fast all the time, sometimes you have to, because, ’hey, market, you know, if you have to do something for your product you have to, but as long as the company knows that if you go faster than you have to pay the price and do it again. [With good] code architecture the level of stress level can be reduced so much. There’s nothing you can do about the importance because in the end, it’s what is going to help you so much along the way...[it can] reduce debugging a lot because if you can separate your logic in nice pieces, you’re not going to mix concerns. And so when something’s not right, you know exactly where to look.”*

> *“I’m committing more often [than I would if I were working alone] following the guidelines of what the team set. Typically the way we do it is there’s the master branch and then we have our own topic branches that we develop...in order to have a successful coding review, you have to commit and push to your topic branch. Once the team is met with consensus, like, ‘Okay, I think we’re ready to go to the next version,’ we then would merge all of our code to the new would be master.”*

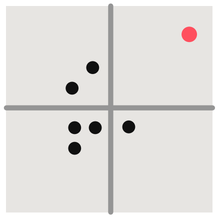
> *“I’m managing all these tools, along with one of the other developers, it can get a bit frustrating. For instance, yesterday, the other developer upgraded our PHP on one of our systems and Jenkins broke. We couldn’t deploy any of our software...it took us three hours to figure out how to get our deployments working again.”*

> *“...code quality, that’s something that becomes more and more important the more people working on [a] project. Same thing with architecture. The problem with all of those is they could be anywhere from less important, low level frustration to more important, high level frustration the more people there are in an organization...Code quality shows the respect you have for other developers, I think, because you’re not the only person who’s going to work on something. You have to make sure that you leave the code in a place that is going to make your life easier and everybody else’s.”*

> *“[Ops is] getting the code out there and making sure it doesn’t fall down. As a front end dev, it’s mostly someone else’s problem, not because I don’t want to be involved, but I find that in a lot of places I’ve worked there’s this attitude that front end devs don’t know anything about Ops. It’s weird, because it’s like, ‘Why would you expect the Java Devs to know how their stuff is deployed, but not expect me to know how my stuff is deployed?’ Which I find frustrating...there was a bit of a lack of respect for the front end, which is one of the reasons I left.”*

> *“Debugging, I don’t have a ton of tools based frustration or product based frustration with it. Mostly just reading other people’s, and I count [me] from six months ago as other people, their poorly commented spaghetti code.”*

Hiring Web Developers



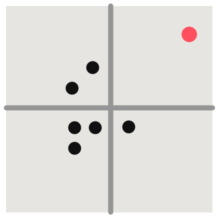
What we heard about hiring can be summarized with two additional need statements:

- Finding developers with the right skills
- Vetting the skills of potential hires

We learned during the pilot interviews that different regions of the world have different skill sets to offer. Hiring managers evaluate regional skill-sets when making decisions about where to post and advertise a job. Vetting the skills of developers, prior to offering them a job proves difficult, and if not properly vetted can lead to a poor hiring decision.

> *"...we've hired three developers that didn't work out...some of them... lied on the resume. They said they were good at certain aspects of technology where they weren't. Because of that...we had different expectations of them and they weren't very motivated to learn."*

Infrastructure Variance



This refers to intermittent power outages that affect business decisions, e.g., where to locate data centers. We experienced the impact of this first hand while trying to conduct the interviews. One of the participants was located in an area where they were experiencing intermittent power outages and was unable to complete the interview.

> *"Early on, there was a decision not to deploy into the cloud because our parent company has his own data center, and the data center is located in [a country where] we have issues that sometimes the power goes out, or sometimes there is a network outage. It is a developing country [it] doesn't have the same stability as you find in North America or Europe. We are looking into alternatives and whether or not we should be moving more of our services to the cloud, and how we can make things more stable."*

Overcoming Frustrations

Overcoming Frustrations

After exploring the topics that cause web developers frustration, but are also important to their work, the conversation shifted, focusing on the resources they use to overcome these frustrations. Two dominant resources help developers overcome frustrations. The first is a community of peers, typically personal acquaintances or acquaintances with few degrees of separation. The second is tapping into the resources available online.

When developers seek input from their community of peers, they usually have a sense of the skills and talents of others they are close with. They call upon people to help with a tactical problem such as troubleshooting an issue or help them with choices that will help them grow as a developer such as advice on what tools or frameworks they should invest their time to learn.

> *"I think the best way that you can learn how to do anything in this space seems to be to have a group of people who automatically or implicitly or without much effort on your part keep you aware of news that you would otherwise not see about the state of things."*
> *"In the case of Ops, my friend is good at it. He just does it for me... that's literally how I solved by DevOps problems. I just tell him, 'Hey, this is my problem, can you help?'...he just tells you what to do next. And that's very useful."*

> *"There are local developer communities. So it's very easy...[the] people are really helpful. I just say, 'Okay, I've done this, and what should I do next? What would you advise I do?'"*

> *"...we have a few people who know [security] or have trained in sales and security, and we continuously go back to them...they're just friends."*

When developers tap into online resources, they are using the usual suspects (listed in alphabetical order): freeCodeCamp, GitHub, Google, MDN, Reddit, Stack Overflow, W3Schools, and YouTube. Part of what makes these resources so valuable in troubleshooting and overcoming frustrations is that they provide developers access to a community beyond that they have cultivated in their networks. This broader community has very likely experienced a similar situation and can help resolve the issue, whether that is shared through forums or videos. These sites also offer official documentation, which is a key resource in troubleshooting.

> *"The web is just an incredible resource because it's almost always the case that somebody has the same question you have..."*

> *"The primary point of contact [for troubleshooting] is, how do I form a Google search to get to the information?"*

> *"[To overcome frustrations with something I don't know] I will just Google and find information or YouTube. [Some common sites that I find to be helpful] are Stack Overflow, and of course, Mozilla, W3Schools...[the sites are] easy to access, easy to find information on what I need, well organized, visually attractive, and easy to understand."*

> *"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."*

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."