GDEVS [SOTR071] SEARCH OFF THE RECORD - 71st EPISODE

[00:00:11:02] - **John**
Hello, and welcome to another episode of Search Off the Record, a podcast coming to you from the Google Search team discussing all things Search and maybe having some fun along the way. My name is John, and I'm joined today by Lizzi from the Search Relations team, of which I'm also part of. Hi, Lizzi.

[00:00:29:11] - **Lizzi**
Hi, John.

[00:00:30:17] - **John**
And today we have Rick Viscomi joining us as a special guest. Rick, would you like to introduce yourself?

[00:00:36:27] - **Rick**
Hi, everybody. I'm Rick from the Chrome Web Performance Developer Relations team, and I help the web get faster.

[00:00:43:17] - **John**
Cool. So you make internet cables?

[00:00:49:09] - **Rick**
Not quite. On the software layer, I think. I help web developers understand how to improve their Core Web Vitals performance.

[00:00:55:09] - **Lizzi**
Oh, okay. And is that what we're going to be talking about today?

[00:00:59:02] - **Rick**
I hope so.

[00:01:00:08] - **John**
I think that's a good topic. Yeah.

[00:01:03:11] - **Lizzi**
So what are Core Web Vitals, and why are they core, and why are they vital? And why is both of these words in this term? Core and vital. They sound very important.

[00:01:12:26] - **Rick**
Yeah, I think the word "vitals" here is meant to evoke that sense of like, "These are your core metrics to actually your health." Like, if you can take your pulse or your blood pressure, those are your vitals. When we want to look at the health of the web, we have metrics like how fast is the page loading? Or is it doing too many things visually to interrupt your concentration on the page? And so we have metrics like these that we use to measure whether the user's experiences are good. The reason why we call them the core metrics are because there are other non-core metrics implicitly. For example, Time to First Byte or TTFB. That's a really important metric that we can use to measure the difference between back-end performance and a front-end performance. But that metric alone is not necessarily related to the user experience. A user doesn't necessarily know, "Well, my back-end performance was slow, but the front-end performance was fast." It doesn't really matter then. They just care about, "Did the thing happen that I was waiting for when I navigated to this page?" So, for a metric like Largest Contentful Paint, it's actually made up of several sub-metrics like TTFB, but there's also more front-end-focused metrics that are useful as well. So the whole Web Vitals program is a collection of metrics, but the Core Web Vitals are just the three: Largest Contentful Paint, Cumulative Layout Shift, and now Interaction to Next Paint.

[00:02:41:05] - **Lizzi**
Ah, so what a good way to think about the core ones are the ones that are keeping you, like, alive, like my heart rate, like my heart beating is the one that is critical. And then maybe other ones are... I don't know which ones are not vital to me being a human and alive in the world.

[00:02:57:09] - **Rick**
Yeah, it's not that serious. Like, your site will not go down if it's not totally fast. And, of course, there are other metrics of web quality than just the performance ones. So it is a little overly dramatic, but it does mean that Google and Chrome, the teams who created the Web Vitals program, are giving the web community sort of benchmarks to aim for.

[00:03:20:11] - **Lizzi**
I see.

[00:03:21:14] - **Rick**
These are the metrics that, if you achieve the good thresholds for them, you can be reasonably confident that your site has a good user experience. And it's, of course, ever-evolving. There will be new metrics over time. They might be fine-tuned in a way, but it's meant to basically give developers confidence that their sites are performing well.

[00:03:39:08] - **Lizzi**
And, when you say performance, is it synonymous with speed or is this like a...

[00:03:44:12] - **Rick**
Yes and no? I mean, that's a good question because we're talking about web performance. LCP and INP are you can think of as web performance metrics, like a user does something, and something needs to happen within a certain amount of time for the experience to be considered good. But there is another metric Cumulative Layout Shift, where it's about things jumping around on the page. It's less about the speed of an interaction or how fast the page loads, but it is a measure of the user experience quality. So, in a loose interpretation of the word performance, it's like the page is performing in the way that the user expects.

[00:04:22:10] - **Rick**
It's a good quality of experience, and things aren't jumping around and confusing the user and getting in their way of doing what they came there to do.

[00:04:29:04] - **John**
Cool, and are these like obvious metrics for site owners? Like, if you pull up the tool and it says like, "Oh, LCP is this," do they understand what is happening there? Or are there things that people get confused?

[00:04:46:29] - **Rick**
Yeah. So web performance has been around a while, and it has started with a metric like the loading time of a page, like my page loaded in x seconds. I think people have an intuitive sense for when the page felt loaded, but it wasn't necessarily the case that this metric from ten-plus years ago was actually measuring that. In the name Largest Contentful Paint is clues as to what we're trying to actually measure. So working backwards from the paint part, something visually changed on the screen. This is what makes it a user-perceived metric, as opposed to something like on load, which would be affected by things like images finally loading below the fold and nobody can see that. So what makes it a contentful paint is that it is something important to the page contents themselves. This is your things like H1 headings on the page or that really big hero image. So, if you're trying to read a news article, the title and the image are pretty important to the user experience of feeling like you can start to read that in the paragraphs of text. So the largest piece of content on the page would be like that hero image, so that the user can really feel like, "I'm ready to go. This page is done loading." It's this one metric, that you can think of it almost like the lowest common denominator for most websites. It's not always the case that your LCP element is the thing that matters most to users. A good example I give a lot is YouTube. If you're watching a video, it's like the start time of the video, how quickly it can start playing, and also the experience of being able to watch it seamlessly. And it's not like stuttering. There are

other metrics that make a web page have good performance to the user, context dependent. But you can think of these metrics as really just like the ones that are applicable to most pages, if not all, in some common way.

[00:06:42:18] - **Lizzi**
So, for LCP, would that be something that every page would need to care about? And it's not necessarily like depending on whether it's a YouTube video that's loading first or a button or something, whatever the user is expecting to then click on or expect to happen, like the video autoplaying. Whatever that thing is, it should happen in a certain amount of time. It doesn't necessarily matter if it's text or video or a button. Would that be a correct assessment?

[00:07:08:04] - **Rick**
I think generally it's a good idea to optimize LCP, but it's not always necessarily the case. I think site owners should understand what their users are trying to accomplish when they visit their site, and optimize the things that are critical to that user journey. It's not always the case that the biggest piece of content is what they're there to do. Maybe it's like that really small add-to-cart button. There's a famous Amazon case study where just optimizing that button creates a lot more conversions. So it's a matter of knowing your site, knowing your users and knowing the correlations.

[00:07:42:04] - **John**
Okay. So it's probably something you would need a tool to measure, in that case, to at least measure objectively where you couldn't just say like, "Oh, I'm going to click Refresh and use my stopwatch and wait until things are finished spinning." It's like a more nuanced metric than just that.

[00:08:04:07] - **Rick**
Yes. If you're testing locally, your eyes will deceive you. You definitely want to have cold, hard data that you can point back to and say, "This is how long it took." But also your users are not running your websites with a stopwatch and emailing you later to say this is how fast or slow it was. So that's why it's really important to have APIs that are built into the browser, and you can pipe the data from the user's browser and their local experience back to your analytic servers. So that way you can see the full distribution of user experiences. And then you could say, for example, with Core Web Vitals, 75% or more of user experiences need to meet this threshold for the metric to be considered good. And so it's really about the full distribution of experiences rather than any one number.

[00:08:52:08] - **John**
Cool. Yeah, I think that's a good point. That's something that I sometimes hear from people externally, that they get confused about this where like, "Oh, but it's like Search Console says my page is slow, but when I try it, it's fast." It's like, why is Search Console wrong? And that sounds like that difference that you're describing, where it's like you're testing it locally versus what people actually see. Is that correct?

[00:09:20:20] - **Rick**
Absolutely. I think sometimes people take for granted that the machines we use as site owners or even developers to build our websites is not at all like the devices used by our real users, and we might be having users accessing our websites from really low-end devices or on really slow networks. And it's very important that we test our websites under similar conditions. It helps to build empathy. Facebook had a really famous program called 2G Tuesdays, where they would force everybody in the office to have 2G-like internet connection speeds, and that led to so much more empathy with the users who were accessing the website from such slow performing devices. And it led to faster Facebook website, which is great. I think developers need to do the same thing on their own websites.

[00:10:10:20] - **Lizzi**
Does that impact how people even design the layout of their page? Like, would they load text first or something that isn't going to be as heavy because of this? Would it impact even just the whole approach to how they would design a page?

[00:10:22:16] - **Rick**

It definitely could. There's a good case study on YouTube, again, where they redesigned the whole page with web performance in mind for a really low-end device type, where they just didn't have the capabilities to load the full experience with all of the comments, all of the suggested videos, and the expensive video player is like, "Okay, we can provide a pared-down experience for these users," and they measured the effects on web performance. And you would think, "Well, when you have a really basic web page, it's going to be faster, right?" Well, they looked at the data and it got slower.

[00:10:58:06] - **Lizzi**
What?

[00:10:59:11] - **Rick**
It wasn't because the page took longer to load. It's because the types of users that were now able to start using YouTube were coming to this pared-down version of the app, and it was a mixed shift effect of slower users starting to use it. Their A/B test wasn't fully perfect because they weren't controlling for the experiment group 100%, but it's a really great reminder that you really need to understand the data that you're looking at. But, to answer your question more directly, yes, you should be building your websites in a way that makes sense for the types of users. There are client hints that you can build for so that you know your user is arriving to your website with some sort of a limitation. But, in the general case, there are things that you could do. For example, there's an app-shell model where you can kind of like show the skeleton of the page, where things will be when they get laid out. You'll recognize that as those gray boxes. If you're loading a page and something's just sort of like there will be something that loads here, but not quite yet, it's better than looking at a blank page with a spinner, for example. It's less jarring to see.

[00:12:08:14] - **Lizzi**
They do this for the RSVP site for any of our Search Central events. When you go to register for that, you see those gray rectangles, and that website is really slow. I don't know, like when I see those, I don't know as a user how much it helps to know that, yes, there will be boxes there. I just know that it is slow and whether I see the spinning wheel or boxes like, ooh.

[00:12:30:14] - **John**
Well, I guess at least it doesn't jump around.

[00:12:33:25] - **Lizzi**
That's true. I think, if I had to pick one of the experiences or the Core Web Vitals, that is my most pet peeve one, is the one where you're like about to click something, and it jumps to another position. It, for me, is the most frustrating one.

[00:12:47:19] - **Rick**
That's right. If you can't have a fast-loading web page, at least make sure that, when the contents do load—

[00:12:52:03] - **Lizzi**
Yes.

[00:12:53:06] - **Rick**
--it's not going to be a totally disorienting experience. So that's Cumulative Layout Shift. That's one of the Core Web Vitals metrics.

[00:12:59:21] - **Lizzi**
Okay. So, for me, that CLS would be personally my pet peeve.

[00:13:03:21] - **John**
We recently announced a new one that's now a part of the Core Web Vitals: INP, Interaction to Next Paint. Can you

tell us a bit more about that?
[00:13:14:13] - **Rick**
Yeah, so it might be helpful to start from also First Input Delay, its predecessor. And then we can go on to the differences between them. First Input Delay measured the first time a user clicks or types on a web page. The amount of time that the browser has to start responding to that interaction. You can probably imagine what the limitations of that type of metric definition are. Only the first interaction. Obviously, if you're using a web page over a long amount of time, you're going to be interacting more than once, and the first may or may not be the slowest one. That might not be the one that matters most to what you're trying to do on the web page. Also, only measuring that delay portion of the interaction doesn't tell the full story, because once you are able to start processing it, then you have to process it. There's time just to handle it. And then, after you've processed it, you have to present the next paint to the screen to give the user some sort of visual feedback that things are happening. And so that's what Interaction to Next Paint does. It measures all interactions on the page. It measures the full interaction latency, and the number that gets reported back is approximately the slowest of those interactions. You can think of it--

[00:14:29:03] - **Lizzi**
Oh no.

[00:14:30:05] - **Rick**
"Oh no," as in it's not the slowest?

[00:14:31:27] - **Lizzi**
"Oh no," as in like, does that mean my stat is going to be worse because it might find the actual slowest one? Like, maybe the first one was fast and now maybe my FID score...

[00:14:43:18] - **Rick**
Yes.

[00:14:46:18] - **Lizzi**
FID compared to INP is going to be worse if I compare the two, because now this new score has uncovered potentially the worst one on my page.

[00:14:54:25] - **Rick**
I mean, almost everybody had good FID performance. 99.9% of websites' desktop experiences had good FID. I call it FID, and it was something like 95% of sites' mobile experiences had good fit, so it was an overly permissive metric. The good news is, with INP, now we have a view into real user experience issues that were affecting our users. They existed before. Just because we have a new metric doesn't mean that the websites are suddenly slower. They were always this slow and unresponsive. So having this metric now means we can identify the issues and fix the issues and push them out to users, so that way the user experiences get better. And then ostensibly, after all of that, they start to do more good things on the website, like converting more.

[00:15:49:17] - **Lizzi**
Does it also flag which one is the worst one, or do you need to then, I don't know, do some like investigative work to find out which thing is the slowest, the worst one on the page?

[00:15:58:27] - **Rick**
Yeah, it depends on what you mean by it. So, if you're looking at like PageSpeed Insights or Search Console, it'll tell you, "Your INP value is this." And it's one number expressed in milliseconds. And it's like, "Well that's interesting. Thanks. I have a performance problem." But then you have to go down and analyze and debug and diagnose what is actually causing that. And it could be a whole array of issues. It's not always just one interaction that everybody is getting tripped up on. Chances are there's something happening on the page, and there are many different types of slow experiences. So this is why it's so important that site owners collect data about their user experiences. And so the Chrome team has published a library called web-vitals.js. This is something that you can add to your websites and hook into your analytics. And with it comes attribution data. You could see what was it that the user clicked on?

When did they click it? And now with a new API called the Long Animation Frames API, we can get information like, what was the script that was running that took so long that caused this interaction to be so slow? So there's a lot more data that the developers can get out of their user data to help inform what went wrong. And then you can put those pieces back together and have a reproducible test case that you can look at locally and see the issues happening for yourself, and then go in and fix them.

[00:17:21:20] - **John**
So cool. Okay, so maybe we should tell more people about these libraries, that they actually implement them. Is that something you could tie in with Analytics or is that like completely separate that you have to implement it?

[00:17:34:13] - **Rick**
So the web-vitals.js library is totally independent. It's just like a few kilobytes of JavaScript. You drop it on your page, and then you can add hooks so that, once there is one of these Core Web Vitals metrics reporting a new value, you get to choose where to send that data. You can pipe it back to your Analytics. It works with Google Analytics. For example, I've created dashboards where you take the data out of Google Analytics and visualize it yourself. This is like a homegrown solution, but there are real user monitoring or RUM tools that site owners can use that take care of all this for you. Many of them are actually built on web-vitals.js, but some of them use their own data collection to measure the Core Web Vitals. But it's so important, no matter what solution you use, you're measuring it for yourself because the Chrome User Experience Report, which is what feeds into Search Console and powers the PageSpeed Insights dashboard, that dataset only tells you how fast your experiences are, and you need to know which sites to look up. Otherwise, it falls back to origin level if there isn't page-level data, but for the whole picture, you need to have the data that you can slice yourself, because that will tell you what your popular pages are, how fast they are, and then all of that attribution data that I mentioned.

[00:18:49:21] - **John**
Cool. Lizzi, have you checked our documentation with regards to Core Web Vitals?

[00:18:55:13] - **Lizzi**
I have, and you're not going to be happy.

[00:18:59:06] - **Rick**
What?

[00:19:00:09] - **Lizzi**
So the score, I ran two tests, and I got two different scores, which I feel like is a common story that you hear from people that run it. And I'm sure that if I ran it on the train coming home, I would have a different score. But, for PageSpeed insights, I ran it on the page experience documentation page just for fun, and I have a score of 45. It's red and scary looking. How upset should I be?

[00:19:27:02] - **Rick**
First of all, what you're describing sounds like the Lighthouse score of PageSpeed Insights.

[00:19:30:29] - **Lizzi**
It is.

[00:19:32:01] - **Rick**
And I think this is such a common cause of confusion because developers see one single number and it's red. It's scary. Do I need to panic? And I get this question all the time and I say, "What really matters is what your real users are experiencing." And so what might have happened in this case is that there might not have been enough real user data for that specific page to exist in the Chrome User Experience Report. And so in PageSpeed insights, it falls back to the Lighthouse data. And you can think of Lighthouse more as like a worst case scenario. So it's using like a throttled connection on an emulated mobile device, which may or may not be realistic for your users arriving to your

website and the performance metrics that come out of that, which is what that number 40-something is based on, doesn't actually interact with the page at all, so it doesn't scroll. It doesn't click. So you might not be experiencing, I'm telling you, it might be worse than that.

[00:20:27:13] - **Lizzi**
Oh my god.

[00:20:28:15] - **Rick**
Yeah. In some way, yes, because if you scroll down, you can incur a bunch of new layout shifts that might not have otherwise been seen by Lighthouse. If you didn't click on that button, we don't have any Interaction to Next Paint to report to you. So instead there's this like proxy metric. And this actually gets back to like Web Vitals versus Core Web Vitals because there's a proxy metric called Total Blocking Time, which is the lab equivalent of like First Input Delay and Interaction to Next Paint. And that tells you like how long the main thread was blocked with long tasks like doing a bunch of unnecessary work, but nobody actually clicked and got to feel any slow interactions in a test like this. So it's really just meant to be, "Hey, something could go wrong here if a user did click during one of these long tasks," but I would say focus on the field data for sure.

[00:21:20:02] - **Lizzi**
Okay. In terms of user clicking, what about Googlebot navigating my page and experiencing my page? Maybe this is a question for John, would this be like a good proxy metric for Googlebot or not?

[00:21:33:22] - **John**
I don't think so, because Googlebot doesn't really click on pages, because what basically happens is we load the page and we look for links, and then we load those URLs from those links separately. So it's not that Googlebot is navigating from one part of a page to a different page. It's basically acting like, "Oh, here's a completely new page, and I'll load it in a new browser to see what happens." So there's no connection there. I could imagine something like Largest Contentful Paint maybe interacting with rendering a little bit. But that is really hard to say because, with rendering, we're very patient with regards to the page loading. We load it like a user wouldn't load it because we load it in a very large viewport to try to get as much as possible of the page into the rendered view. We try to wait for the whole page to finish loading. Some pages never finish loading. That's kind of like, I guess, the rendering team's problem. But, for the most part, we don't really care if it's fast or slow. Googlebot is patient. A lot of the content that does get loaded is cached. I think it's a very different scenario there. I do see SEOs conflating this a little bit, sometimes when it comes to things like crawl budget, where they say, "Oh, my server is slow, and therefore, Googlebot is not crawling a lot," and server's slow is sometimes projected as well as like Core Web Vitals are also slow or bad, which is like I'm sure there's some relationship between the server being slow and the page also being visible slowly. But you can have a fairly slow server, and it still loads pretty fast.

[00:23:40:05] - **Lizzi**
So pinpointing what the thing is that's slow is maybe difficult for site owners, SEOs, everybody involved here.

[00:23:47:02] - **John**
Yeah. I mean, we do use Core Web Vitals in other ways in Search.

[00:23:51:16] - **Lizzi**
Oh, tell us more.

[00:23:53:18] - **Rick**
Tell us.

[00:23:54:26] - **John**
Tell us more. Tell us all the secret. Stop giggling.

[00:23:59:07] - **Lizzi**
Sorry. I'm just here to be the peanut gallery.

[00:24:04:02] - **John**
Yeah, it's something. With the INP change, we went back to the Search ranking teams and really went through all of the metrics that we have, I think, on the page experience page and really discussed how are you all using this and how does this play a role in Search? From that, we kind of went into this discussion and thought like, "Probably they're not using any of this anymore." It will be really hard to explain to people externally, but it turns out they are using Core Web Vitals. So that's something which we documented a little bit more explicitly in the documentation now, that we do say we do use this in our ranking systems or in Search systems. The details we tend not to go into. We don't go into thresholds or anything like that. Similar to how we don't talk about how many words on a page you have to have or all of those details, which, from my point of view, are almost secondary. I think a big issue is also that site owners sometimes over-fixate on the metrics themselves where they see some number and it's like, "Oh my gosh, I have to get this to like some other number, some higher state." And then they spend months of time kind of working on this. And they see this as they're doing something for their Search rankings. And probably a lot of those incremental changes are not really visible in Search. I think that's really challenging because, on the one hand, there are very few metrics with regards to SEO that you can look at explicitly say, "Oh, it's like 17 and I can make it 15." So it's, I don't know, like human nature to almost focus on them. But, at the same time, you have to be careful that you don't over-fixate on them and spend an inappropriate amount of time.

[00:26:12:12] - **Lizzi**
Yeah, focusing on that and then still having, like a terrible article, like the words on the page are not good or the design is not good, and you made it really fast. Okay. Is that really going to make an improvement for your users or for Search?

[00:26:28:00] - **John**
I sometimes position it as the extreme case, like you have an empty page, it'll load super quickly and it's like, "Congratulations." But that's not going to be good for Search.

[00:26:37:23] - **Lizzi**
Yeah, I think that's a good way to think about it because the numbers thing, it does align with the pitfall that people fall into with the how many words do I need in a meta description or on the page or whatever? I will get to this particular thing. And now it's good. I've reached it. I have the Core Web Vitals check. I will now rank well because I have them, but in reality, we all have Core Web Vitals.

[00:27:01:18] - **Rick**
The way that I look at it is less about getting people onto your website, but once they're there, how is their experience? There's a whole user journey from, "Okay, I've loaded the site," and the user has some intent that they need to complete once they're there. Core Web Vitals are a way of ensuring that they're going to be able to complete that journey in as frictionless a way as possible. When the site is unusable, it's slow. It's unclear, unhelpful. People bounce. They say this site is just so frustrating. I can't use it anymore. And they leave. They go to their competitor instead. And that's really what it's about to me, is the frictionless journey of accomplishing some task. A site owner might look at that and say Conversion Rate, but to a user, they're just there to do something. Let's get out of their way, and let's not make the site performance something that impedes them. And, in that respect, it's really an accessibility metric. If somebody is unable to use a site, that's an accessibility problem, the same way that I can't read the text, it's too small or if the contrast is poor. People will leave because of these things. It's less about being discoverable and marketing your site, which is like more of an SEO thing. It is very nice that Core Web Vitals does feed into Search. I've even heard from the web performance community like, "Thank you, Google, for making Core Web Vitals a part of Search because it has made so many more people aware of it, of the performance problem that's on the web." And honestly, I really do think that's what's contributed to the improvement we've seen to Core Web Vitals over the years. The program itself is just a few years old, but if you plot the metrics over time, you'll see a significant improvement due to both the site owners making their sites faster, and also browsers like Chrome improving it at the Chrome level under the hood to make sure that the ways that people access sites can be as fast as possible. So that's my soapbox about the importance of Core Web Vitals. I just want to say thank you, Search, for

making it part of your ranking algorithm, and also for making it clearer in the documentation that it matters to Search, because I think a lot of people will continue to care about it.
[00:29:09:23] - **John**
Cool. I didn't realize that there were kind of like these bigger trends that we're looking up because it feels like, from our side, we keep talking about speed and Core Web Vitals. I could imagine externally they're like, "Oh, you just say this all the time." It's like, "Am I never good enough?" And it's nice to hear that it does have an effect on aggregate across the web.

[00:29:35:05] - **Rick**
Absolutely. I really feel like the way that people are building websites is evolving over time. So it's important to remind people about, for example, like the best practices of using too much JavaScript and taking advantage of newer APIs. These are things that I think people see a shiny new JavaScript framework, and they try to implement it on their website because it's really fun and has a great developer experience. But they're not always thinking about the user experience, or they knowingly trade away the user experience for the DX. I feel like that's a mistake. And it takes constant vigilance to remind developers of the costs of those types of decisions.

[00:30:14:10] - **Lizzi**
So all we have to do is remove the JavaScript from our site?

[00:30:16:29] - **Rick**
No no. No no no.

[00:30:18:06] - **Lizzi**
JavaScript is evil.

[00:30:19:26] - **Rick**
It is not evil. JavaScript unlocks much of the rich web capabilities that we enjoy today. Otherwise, we're back in the stone age of really plain boring websites that are just text. We wouldn't have, like, Netflix or things like that where we can watch videos online, on YouTube or whatever. I really think JavaScript is a necessary part of the web, but we need to use it responsibly. It's about auditing your usage of it, not doing too much of it. Not just having it on your website because it's cool and fun. This is the problem I have with client-side rendering. This technique of doing what the servers used to do and now streaming in just the data so that the website can piece it all together itself on the browsers themselves. It's like putting the cart before the horse in a way. It makes your subsequent page navigations faster because then you can reassemble things on the fly that way, which is great, but that initial render should really be done on the server the traditional way. So, in that way, I would say less JavaScript is better, but in general, it's a good thing and I want developers to use it but use it responsibly.

[00:31:23:02] - **John**
Okay, fine. So you're saying it depends.

[00:31:30:12] - **Rick**
I am a staff engineer, so yes.

[00:31:31:26] - **John**
Okay. Cool, I appreciate that. Cool. If there's one thing that you'd like the average site owner or SEO to kind of take away from this discussion, what would that be from your point of view?

[00:31:43:16] - **Rick**
Eat your vegetables and wear sunscreen. Eating your vegetables is a euphemism I use for web performance. It's like doing the right thing and not forgetting, like brushing your teeth. It's just like a part of your web hygiene to go through and look at your user experience data and be mindful about how your users are experiencing your site. It helps to build empathy to see how slow these things can load. You might be complacent by saying, "Oh, it works fast on my

machine and be totally happy not care about this web performance thing," but the data tells a different story. I would just highly encourage anybody with a website to see how fast it is or slow to your real users, and use that as motivation to make it faster and better. Because, like I said, it's an accessibility problem. In the YouTube case, when you make a website faster, it lowers the barrier to entry for so many more people to use it. It makes it more enjoyable. Honestly, it makes the entire web ecosystem healthier. When more people can do their business on the web, it's good for everybody. The users like it because the experiences are faster. The site owners like it because they have more people using their site. The advertisers love it because people can see more ads and everybody makes more money. And a company like Google likes it because we're a huge player in the web ecosystem. We don't want to be the fish in a polluted fish tank where things are just becoming more and more unhealthy. It's really good for everybody, and the rising tide lifts all boats. Check your website. Make it faster. Eat your vegetables.

[00:33:15:20] - **John**
Cool. Any comments from your side, Lizzi?

[00:33:18:21] - **Lizzi**
Oh, I would just be the other side of that is don't panic if you can't get a perfect score. So, on the other side, don't freak out if you don't have control over it. Because I'm coming from the scenario where my content management system has declined my requests to fix the score, and I don't have the controls to make it faster, so to not lose sleep over it if you've tried your best to make your case and it's still not happening or being prioritized from the development team, that it's also not going to mean that you're at the bottom of the rankings just because of this bad score.

[00:33:55:08] - **John**
Well, we'll see.

[00:33:56:18] - **Lizzi**
Well, maybe. We'll see.

[00:33:58:08] - **John**
We'll see. Maybe if we had enough—

[00:33:59:20] - **Lizzi**
What are you saying?

[00:34:00:27] - **John**
Maybe if we have enough people that submit feedback, then you have something to push with.

[00:34:08:15] - **Lizzi**
Oh, so write in with more snapshots about the bad scores of developers.google.com/search. So I can say, "This is a real big problem for this many people."

[00:34:21:00] - **John**
I'm just thinking it. I didn't actually say that aloud. Yeah.

[00:34:26:27] - **Lizzi**
Okay. I'll redact it.

[00:34:29:18] - **John**
Cool. Well, that's it for this episode. Thank you for joining us, Rick. If folks want to reach out to you online, where could they go?

[00:34:37:18] - **Rick**
Oh, well, you could find me on Twitter these days, and I refuse to call it the other thing. My handle is Rick_viscomi. Other than that, if you have any questions about Core Web Vitals, my team also monitors Stack Overflow with all of the usual tags like Web Vitals, Core Web Vitals, Interaction to Next Paint. So we're here to help. Let us know by posting a question, and we'll do our best to answer your questions.

[00:35:01:08] - **Lizzi**
Cool. We'll put all those links in the description as well, so that people can find you without typing in all of the URLs and everything, but it was great chatting with you and catching up.

[00:35:12:11] - **Rick**
Yeah. Thank you so much for having me.

[00:35:13:22] - **John**
Cool. Well, in that case, thank you all, folks, for listening in as well. And goodbye.

[00:35:19:05] - **Lizzi**
Bye.

[00:35:20:09] - **Rick**
Bye, everyone.

[00:35:25:23] - **Lizzi**
We've been having fun with this podcast, and I hope you, the listener, have found it both entertaining and insightful too. Feel free to drop us a note on Twitter, @googlesearchc, or chat with us at one of the next events we go to if you have any thoughts. And, of course, don't forget to like and subscribe. Thank you, and goodbye.