EPISODE 1269

[INTRODUCTION]

[0:00:00.3] JM: Flutter is a UI toolkit developed by Google, that helps developers build natively compiled applications for mobile, web, desktop and embedded devices from a single codebase. Development is fast, because the screen hot reloads as you develop. The architecture is layered for fast and expressive designs, and its widgets incorporate all critical platform differences, such as scrolling, navigation, icons and fonts.

In this episode, we talk about developing Flutter apps with Allen Wyma. Allen is a Founder of Plangora, a web and mobile development company that specializes in PHP, iOS with Swift, Android with Java, WordPress and Flutter. He's also an Elixir mix panelist at devchat.tv, and he's the host of a Flutter podcast called Flying High with Flutter.

A few announcements before we get started. One, if you like Clubhouse, subscribe to the club for Software Daily on Clubhouse. It's just Software Daily. We'll be doing some interesting Clubhouse sessions within the next few weeks. Two, if you are looking for a job, we are hiring a variety of roles. We're looking for a social media manager. We're looking for a graphic designer, and we're looking for writers.

If you are interested in contributing content to Software Engineering Daily, or even if you're a podcaster, and you're curious about how to get involved, we are looking for people with interesting backgrounds who can contribute to Software Engineering Daily. Again, mostly we're looking for social media help and design help. If you're a writer or a podcaster, we'd also love to hear from you.

You can send me an email with your resume, jeff@softwareengineeringdaily.com. That's jeff@softwareengineeringdaily.com.

[INTERVIEW]

[00:01:51] JM: Allen, welcome to the show.

[00:01:52] AW: Thank you for having me.

[00:01:53] JM: You are a Flutter expert. You have a podcast, Flying High with Flutter. I'd like to explore Flutter in this episode. It's been a while since we did a show about Flutter. Tell me about how Flutter has advanced in the last couple of years?

[00:02:08] AW: Well, so I haven't been using Flutter for a long time, maybe just a little bit over a year. I would say that it's really advanced in terms of the amount of platforms it supports. It's funny, when it originally came out, it was actually for web, then they backed away from web and they went straight into Android and iOS. Then they actually became full circle. Went back to go into adding now a web support.

One of the things that actually just came out recently is that web is now production quality, or as they say. That's pretty big. Now you can build your app for iOS, Android, and web, all in production quality. Now, they're just added in the support for desktop. That's Windows, Mac, and Linux. That's what version 2.0. Actually, version 2.2 just got released a couple of days ago, as of this recording date.

Looks like, they actually added in another couple of more platforms, which is really surprising. I think, it's quite big to add in so many platforms at once. That's UWP, which I believe is Windows 10 and also, Xbox. I believe, they're going to be trying to add support for Tizen and Wear OS, something like that. It seems like, they just keep branching out to more platforms. That's quite amazing, I think.

Another thing that came out is sound no safety. That means that you can pretty much guarantee that something will be either not, no ever, or it will be nullable and the code will actually force you to check for nulls before you try to use it. That actually catches a lot of errors that usually we run into. Nulls, I mean, if you ever caught it before, you'll know that nulls will give you – Yeah, they're like a billion-dollar mistake. I think, the guy came up with nulls said that. That's pretty huge.

The other thing, too, is all this null checking is quite interesting. I didn't know this, but they claim that removing the null checking that you have to do removing this idea of nulls, actually reduces the code that you compile to, which is quite crazy. You can actually reduced your app size just by turning on this null safety, so you get the benefits of having safer code, less crashes on your app, and actually a smaller app size, which is really, I mean, I can't find much negative for that. Those are some of the biggest things. Other than that, there's a lot more things coming into Dart. Flutter is ran in – mostly run with Dart. Dart is just exploding with new features. I mean, I can go on and on about this stuff. I think we're here to talk specifically about Flutter, right? We can keep going now about Flutter.

[00:04:51] JM: I am curious about a lot of the things that you just mentioned. Let's talk a little bit about how Flutter actually works. It's used for cross-platform, mobile apps, it works obviously, for both iOS and Android. Give me a bit of a description for how Flutter code gets transpiled, or compiled down to the native runtimes.

[00:05:16] AW: I don't have a very deep understanding of exactly how that works. I believe Eric was on the show quite a few years ago. From my understanding, I'll do the best I can, so please don't ding me if I make a mistake on this one. What happens is that you build this tree of nodes. Maybe you have a column and within a column, you'll have a bunch of text widgets, these text notes, and you end up having a tree. This tree will get created. Then from there, that will actually create some longer-standing tree of, I think they're called elements or something like that. That's how it works.

The nice part is that as long as you keep changing up the tree, then the changes will keep coming in updating the UI. Now, yeah, I feel like, I don't have enough technical background to really explain this. I'm more of on the app side and making apps with this stuff. Yeah, what I can say is that based on a discussion I had with one of the creators of Dart, a lot of this technology actually came from Chrome itself. It's kind of interesting. Because of Chrome, we got V8. V8, we have no JS.

Also, Flutter basically came from Chrome, too. They ripped out some of the pieces of the rendering part of the browser that I understand. Because of that, they were able to create the web platform, which was recently called Sky. Then, they're actually able to render all this stuff using the graphics from Chrome, on mobile, to make the UI for Flutter. I think that's actually a little bit more interesting that because of Chrome, we've gotten a lot of technology just from a browser.

[00:07:00] JM: What's your background in developing mobile apps? Did you write mobile apps before using Flutter?

[00:07:06] AW: Yeah. My background in nearly any tech was just been pushed to actually just do something. Let me just talk about mobile apps since you brought that up. It's a great example. When I was working in a consulting gig in Shanghai, China, my boss came into the office one day and said, “Looks like we have to create a iOS app.” He just came in and said, “We have to create this.” I said, “Great. Who's going to do it?”

He had a Ubuntu machine. I had the only Mac in the office. He said, “Yeah, it's you.” At the time, I never created any iOS app before. I just had to hunker down and just do it. I create a very, very basic app. This was before even Swift. This is Objective C, using – I think the storyboards just came out. It was quite an interesting time. It was pretty fun to use that. In the end, the app looked like a general iOS app. That was my background was just using just the general iOS app.

It functioned. It did what it had to do for the client. Of course, whenever you create a iOS app, you also have to create accompanying Android app at some point. I also had to create that one. I really had an idea about how to create apps from doing this. They probably weren't the best, because again, I made them myself. I just watched a bunch of videos online and then just made the apps. They work. People were happy with them.

Again, they were not interesting. They were just very general-looking, like you open up a typical Android app or iOS app, and just basic components. Actually, because of this, my history with this, I always wanted to actually take a look at React Native and things like that to be able to create cross-platform apps, because I wanted to be able to have an app that not only function, but also looks good.

In the end, I actually thought that the only thing I'd ever end up doing in the end was actually creating a native version of the app. Because with React Native, things never really seem to be very, I guess you would say, performant. I mean, the Facebook app is not very performant in – it's gotten better, but it used to be worse, I believe. Also, trying to work with React Native in the past, too. I always felt very brittle whenever I try to update something or do something. It just ended up breaking on me. I just didn't want to touch anymore.
I've always went native, but the native apps never really looked that nice, because I wasn't very good. I mean, it's hard to focus on one platform, but then two, is very, very difficult. I took a look at Flutter. I just kept hearing about it. Actually, that's what really brought me into Flutter was because I could make a very beautiful looking app. Because the app looked good, it gave all the users of the app a good feeling. I mean, nobody wants to use an app that's not very nice to look at. Function, of course, you have to have, but it also has to look good. Or else, you don't have a good feeling about actually using the app.

[00:09:58] JM: Tell me more about how the development process of building a cross-platform Flutter app compares to building on respective native platforms.

[00:10:08] AW: I have to say, the nicest part about this is that, from my experience, nearly as close, that could be a 100%, but nearly a 100%, I would say minimum, absolute minimum would be 90% of the app is feels about the same across. You do get some differences. If you're on Android and you have a ListView as they call it, so a long scrollable list, and you pull up on Android, you're going to get a little bit of an inkwell, this weird color popping up from the bottom to let you know that you've reached the end of the list. iOS, I can't remember what is the way it looks like, but it is different, so you do get some differences.

In general, the apps are exactly the same. You can make them change per platform, but in the end, they're really identical. That really helps to cut down on a couple of things. One is general testing and making sure things look and feel nice. Because you can bet that when you run this app on Android, or iOS, you're going to get basically, the about the same feeling. There's never any worries about how does this look, or how does that look. That to me is really the powerful part of Flutter is that you get a consistent feeling across, of course, with some differences between the two that people will notice.

It's not something you notice without actually taking a look at how they all work. In the end, you're going to get very, very consistent app across both platforms. That's going to cut down on a lot of things. If you have a bug in one version of your app, like an iOS version, or the Android version, you're going to have the same bugs there of course, but at the same time, if you program it correctly, you're going to not have any bug, because you're going to have a well-performing system that runs on both sides. That is always huge, too.

That also draws me in, because you can really take your time on these apps and play with them more, rather than rush to get them out the door. For smaller teams, it really, really makes a big difference.

[00:12:05] JM: What are the kinds of applications that still cannot be built with Flutter?

[00:12:09] AW: This is a great question. Flutter really excels at apps that have to look nice, and they have to feel nice, and that you have a consistent feeling across, especially when you're doing multiple platforms. It's great for prototyping. What it's not good for is if you need to be extremely native, like if you need to use some really, really native peripherals. If you need to maybe talk with a lot of Bluetooth, using Bluetooth, Bluetooth connection to different devices, using a lot of native components of the phone, and like a particular type.

If you're only working on an app that's only going to be using iOS features, say only Apple Pay, not even Google Pay, then I think that Flutter probably doesn't really fit in that case, especially if you have Swift UI, so you can probably not use it. If you're going to be using a lot of the native SDK features, I think Flutter may be a bit of a problem. If you have an app that you already built a lot of it already out with native, sure, you can still bring in Flutter, but you'd have to really consider, is it really worth it to bring in something extra? Because it is going to cause your app to bloat a little bit, because when you're running a Flutter app, the reason that they look the same across is because it has its own rendering engine. You're not going to be relying upon the native rendering engine of the platform. They have their own engine.

Yeah, your app is going to be slightly bigger, but it's not that much bigger. For instance, I have one of my client apps is about 48 megabytes, I believe. I compare that to Facebook. It's a 150 megabytes. It's way smaller than that. In the end, having a slightly bigger app that looks and feels a lot better, I think is definitely worth the bigger, slightly bigger size.

[00:13:57] JM: Is the tooling for developing a Flutter app, the support and tooling, is it as good as what you get developing on Native SDKs?

[00:14:09] AW: I haven't developed a native app in quite some time. I do remember that the experience wasn't very nice on iOS. I can't remember too much on Android, but I remember that when I wanted to debug things and set through the debugger, it didn't feel very nice. For Flutter, it's really, really nice.

You could just set a breakpoint within Android Studio or Visual Studio code, and you can step through everything. I mean, it works pretty great. They've been really focusing on their tools a lot lately. From what I understand in the latest version, 2.2, they added some memory management functions to their dev tools, so now you can actually take a look at how much memory is your app leaking, or taking up. You can really get a lot more details about what's going on your app than before.

I know there's some stuff within iOS. I haven't really played with it. I can definitely say that I haven't had much issues with debugging my Flutter apps. I just set a breakpoint. I get to it, or I can do simple print statements, whatever. I can pretty much debug whatever I want to. The only time I ever have an issue is when I need to debug a native code from my Flutter app. That is the only time which is slightly more complicated, but there definitely is ways around that. You can actually run the app still from Xcode, or whatever, then you can set your breakpoints within there.

Because when you do have these apps, you can still reach down into the native SDK whenever you want, using these things called platform channels. That means that you never have to wait for anything, any catch up from Flutter, or whoever else to build this stuff out. You could just build it yourself. That's really also another powerful feature of Flutter.

[00:15:46] JM: What are the most complex apps that you know of that have been built in Flutter? Like mainstream apps.

[00:15:52] AW: It's hard to say what's complex, because when you look at a Flutter app, they all look very nice and very simple, and very clean. I think, some of the more interesting things that are being done with Flutter may not even be on typical platforms that you know about. People actually porting the Flutter SDK to other platforms than just the ones that I've mentioned today. Even the new ones that are just been released. I think, I can't remember correctly, but I think BMW or Toyota is actually working on putting it into their media centers. When you get into your car, you'll actually be using Flutter application.
I know, within the Elixir community, there was a guy who, I think his name's Connor Rigby. He was actually porting Flutter to these nerve devices. These ones could be arm-based, or even X86-based, little, mini-PC boards. Yeah, you could be using Flutter for that one also. That's also extremely interesting and I think, quite complex.

There's all kinds of people who are using Flutter nowadays. There's WeChat, Tencent. There is a new bank, which is I believe, they're some type of bank within Brazil. To me, it's quite complicated, I think. It's hard for me to say without actually taking a look at all these different apps and how they all work. Yeah. I mean, I need to know more about what complications you're looking forward to really say, but I think that in the end, I mean, the complicated parts are probably multiple animations that you're doing.

[00:17:19] JM: One of the more obvious questions to ask is, how this compares to React Native? We've explored that in previous episodes. Do you have a perspective on how Flutter compares to React Native?

[00:17:31] AW: Yeah. My perspective, again, I haven't really delved into the tech of the way that React Native works. My understanding of the way React Native works is that you use JavaScript code to somehow call back into the platform to render native code. The code that you're running is actually still running interpreted, so you have to have some type of JavaScript engine running that code for you. The positives of that is that you could do hot code pushes to your device and actually fix bugs. That's pretty cool. A lot of people in Flutter also want to do the same.

The difference is with Flutter is that it actually compiles directly to the hardware. There is no virtual machine. When you're developing with Flutter, it is slightly slower. You do get this thing called hot code reload. That means, as soon as you save your code, literally, as soon as you do command or control S, and before you even lift your finger off the keyboard, it's already reloaded, and you have your changes ready to even see.

That's because of the one of the properties of Dart is that they have this virtual machine that they can ship to your device. You get this hot code reload, which I think is really what makes Flutter stand out. When you develop your apps, you just make changes, you save it, you can see what happens and you can make sure everything's working great. When you do it into release mode, that's when things become more interesting. It does actually compile everything down to machine code. They rip out all of the hot code reload features. There's no VM anymore. You're just compiling directly to the hardware using AOT, ahead of time compilation.

Basically, just a couple pieces are left. One of them, of course, is memory management. Another one's probably the scheduler. The way Dart works is very similar to Elixir and Erlang, where you have this idea of lightweight processes called isolates. The only way you can connect to them is using ports, or sending messages back and forth similar to Erlang and Elixir. That's how you can do multiple things. Yeah, you get this stuff, right? You get this native compiled code running directly on the device. It's extremely fast, extremely smooth, very efficient, and it works great.

[00:19:52] JM: Who have you spoken to on your podcast that have changed your mind about something in Flutter?

[00:20:00] AW: Well, I had a big discussion with Kasper Lund. He's one of the co-creators of Dart. He definitely brought into perspective about why Dart is a good language for the platform and how the situation came up and where they came from what they're trying to do. I also just had a discussion today with a guy named Will Larche, and he's one of the engineering leads in material design for Flutter. He also showed me a lot of things that I didn't even know about before that Flutter does already have a lot of cool properties, where you can just have these adaptive controls.

Depending what platform you're in, if you're in iOS, you get a certain type of switch, versus if you're in Android. The same for some other pieces, too. That's really cool, because there is other ways you can do this. You can also check what platform you're running. At the time, you can say, is this platform iOS? Is it Android? Is it Windows? Is it Mac? Is it Linux, etc.

Now, there's actually a lot of things already in Flutter that do this for you, so you could just say, make this thing adaptive. That just means it's going to try to choose the more native-looking component for you, so then your users feel at home with everything.

One more thing that has also changed my mind is that I did talk with another guy, who actually gave me a tip when I'm developing my Flutter app, is that if I'm developing on Mac, I should develop the app on Mac. Actually, targeting the Mac platform. Versus, even if it's going to be only iOS or Android, because when you're compiling your code for the same architecture that your computer is on, it’s going to be much, much quicker, much faster to hot code reload and to run and to start the app, etc.

When you compile your code for another platform, it has to do a lot more work. That's also like something, that also changed my mind about when I actually compile my app, I should probably be running it natively to my platform, then I just export it. Actually, today, I actually spent some time showing Flutter to somebody. We actually created a very simple iOS app, then I turned it off.

They said they're interested in learning how to make web app. The same app idea, which just started up in the web, show them that it's already done. Then I turn it off, and I turned on the Mac and said, “Look, we also have a Mac App, too, because that's developing on my Mac machine.” It really also blew them away, too. This is the fact that I could show somebody how to make a mobile app. Then without thinking too much, I just, okay, let's show you what it looks like on web, let’s show you what it looks like on Native Mac OS. That also chases my mind to make sure that, okay, what I'm using for my mobile apps is actually something I could use for nearly everything, because it just supports everything on the box nearly.

[00:22:44] JM: Tell me about the debugging experience of writing a Flutter app, because I've talked to some people who work on Flutter apps. Sometimes the debugging experience sounds like it can be – the error messages can be quite cryptic.

[00:22:58] AW: Do you have any more specific examples? Because for me, the error messages are usually quite straightforward. I haven't had anything too cryptic. I think, the only time that things have been cryptic is if I was calling some native code, then that can be a little bit cryptic.

[00:23:15] JM: Okay. The debugging experience in general?

[00:23:18] AW: For me, I usually just – if it's only Dart code, I just send a debug point and I'll just hit the button. As soon as it gets to that point, then I can step through everything, see what's going on. I mean, that's been my experience, or of course, typical print statements. The only thing I can complain about for the debug experience is that sometimes you need to switch apps or something, and you're running in debug mode. Yet, it just stops, or disconnects. That happens sometimes. That's not fun, right?

Because maybe your app is calling into another app for a walk. You could send permissions or something to read data. I've had that happen before. Like I said, that's not it's not fun. Otherwise, I think most things is quite clear. There is some gotchas sometimes that you – the way Flutter works again, is using this idea of the widget tree. Usually, you'll have something like a provider or some type of widget, which will be holding some state. You can always traverse up the tree, but you can never traverse down, or to sides, to the siblings of your node.

Sometimes, you will try to reach up to grab some of that state and it's not going to be there. That is not also a pleasant experience. Then, it takes some time to understand what you need to do in order to make that happen, how to actually fixed all that. Once you wrap your head around a couple of things, it becomes basically second nature, and you start to understand why certain things are breaking at certain points. That's been my experience. I mean, I did quite a few apps in Flutter already, so I went through a lot of this troubles at the beginning. Once I wrap my head around everything, it's not too bad.

[00:24:55] JM: What about the challenges of maintenance of an app? One thing I wonder is, can you hire Flutter developers easily enough? Or is it, if I work on a Flutter app and then it gets to production, then I don't want to work on it anymore, can I find somebody who is talented enough in Flutter? Or is the market sparse?

[00:25:16] AW: I mean, it depends. This is a very good question. I mean, I'm pretty deep into the Flutter community. I always feel there's a lot of Flutter developers. I can understand people say, it's not so popular. I am an employer, so I have some employees. One of my previous employees, he actually came from a physics background. He's never really coded before. He spent about a weekend just taking a look at Flutter by himself, not even coding anything. He just looked at it. Then he came in on a Monday and he just started working on it. He just started going through everything and wasn't really a problem.

Sure, he wasn't a 100% up to my speed, but he was able to actually do things. He's actually built something. I've been hanging around a lot on the Facebook groups for Flutter that I could find. I join them and I've been reading what's going on. There's an explosion of people in Asia, in particular, like Pakistan, Iraq, Indonesia, Philippines. These areas of people who are just – they're just downloading the Flutter SDK, they're just installing it and they're just going, they're just running and they're building stuff.

My personal opinion is that Flutter is extremely easy to get up to speed, no matter if you've had a background coding or not. Because a lot of these guys have no background at all, and they're building apps. Now, they may not be bug-free, but they're building apps and some of them look really, really nice. I've also put out a couple of workshops here in Hong Kong. I'm based out of Hong Kong. I've actually found that a lot of designers who've actually never even coded before, able to pick up Flutter quite quickly and make some really interesting-looking apps. Because that's really the power of Flutter is that you can make some very beautiful-looking apps just by using a couple of widgets and components.

I have to say that, maybe you won't be able to find somebody who's been a native – someone who’s been a photo developer for a couple of years. You can definitely find somebody who can do it, or you can pick it up yourself, or you can find somebody who can pick it up. It's not that difficult to do. I really believe that and I've seen it myself.

[00:27:25] JM: When you're building Flutter applications, are there any other challenges that you encounter that you would not encounter if you were just writing native mobile applications?

[00:27:36] AW: The good and bad thing about Flutter is when the packages that are available for Flutter are good, then Flutter is great. When the packages available are not very good, then Flutter is really not a very good experience. I mean, Flutter is basically focusing mostly on the UI side. If you go to flutter.dev, you'll see that they're basically, talking about Flutter being a UI toolkit. That's basically what the main piece of it is.

Of course, you can just start your business logic and everything in there. In the end, I mean, most apps need to do things more than just the basics. You snap a picture, you need to actually store data on the phone, you need to use some native pieces of the phone or something, particular to Android or iOS. That's when you can run into some problems. If you're creating obviously, a native app, you have full control. You can write whatever you need to. You can do whatever you want.

If you don't have that experience or that ability, then that could be a problem. That's why I said that if the packages that are available for you to use are not very good quality, or they're too outdated, then it can be an issue. I did talk about before that there's this idea of sound, no safety. You can only have sound, no safety, if all the packages that your app depends on are also sound, no safety.

Actually, what I'm running into for one of my apps I wanted to update is that it has a ton of packages. Some of the packages that it does require are actually haven't been updated for some time. They may have been discontinued, I think, even some of them. Actually, for sure, one of them has already been discontinued. I had to change packages for that one. That's not a good thing, right? If I wrote this natively myself, I would never have this problem.

At the same time, I don't have to actually work on that. I got most of the things I wanted for free. Otherwise, it's not a big deal. In the end, when I switched packages, I also had to actually extend that package with some features that I needed from the previous package that weren't in this one. Yeah, I mean, there's positives and negatives. Like I said, one of the worst negatives is that there is some of these packages which have not been updated in some time. Because of that, they don’t all have the no safety on them, so I can't enjoy the sound no safety that some other apps, or photo apps have. That to me is definitely a negative.

Yeah. I mean, that is definitely something you're going to run into. I think that for cross platform apps, like Flutter, like React Native, it's always going to be a problem. At least, I think with React Native, it's even more bare bones from what I've been hearing is that you can't do anything interesting without a package. For Flutter, I mean, you can do things – you can just have a typical app where you just log in, log out, use some data. You don't really need any package for that, so that could be fine.

React Native for my understanding is that even for rendering certain components, there's nothing in the native, or when I say native, there's nothing in the actual React Native SDK that can do a lot of the things that you want to do. Yeah. I mean, there's definitely some negatives. I think, the positives in my experience so far has proven to me and given me the confidence that I don't ever see a reason to go back to native, especially since I could make whatever I want. I also have the experience and know how to actually extend, or create whatever package I may need to do. I really can't find a good reason to ever want to abandon it. Unless, there's something that comes out in the future that's even better.

[00:31:28] JM: Do you run a consultancy based around building Flutter applications?

[00:31:33] AW: Yes. Actually, we build apps. I mean, we use Flutter for mobile apps. We actually did build a desktop app twice now using Flutter. It's been a fantastic experience. We also use Elixir for our back-ends, because some things are actually, we have to have a lot of concurrency and scale. We also use Rust, too.

A new feature actually came out for Dart is they have FFI now. When we built one of the desktop apps, we actually needed to build this for Windows. What we did was we actually generate a bunch of DLL files, dynamic library, for Windows using rust. all the business logic was all in rust. we just connected directly to Dart FFI, so just directly to the Dart VM, or Dart language, whatever you wan to call it, the platform. We were able to just call into Rust to do whatever we want to do. Open up files, save data, even call out to the APIs. Then in case Flutter at that time was not actually very good on desktop, we already had the main business logic and build files. We could just switch platforms and we wouldn't lose out too much. Yeah, I mean, we definitely do a lot of Flutter over here. More and more each day, it seems like.

[00:32:50] JM: When you wrote the web application, Flutter, did you reuse – I guess, I was just trying to understand if you used Flutter on web, as well as on mobile, if there's a component reusability between the web and mobile.

[00:33:02] AW: Yeah. Actually, no, everything is reused. Because Flutter controls the rendering of the UI, totally on all platforms, you can run everything across. The only thing is of course, when you interact with it, if you have an iPad versus a desktop, you're probably not going to have touch abilities, like multi-gesture, but you can compensate with scroll wheel, or double-clicking, or right-clicking, etc., so you can get something similar.

Yeah. I mean, for instance, there was a guy, a client, he gave me a designed to do in HTML CSS. This was a time when Flutter 2 became stable and web was stable, that they officially been stable. I just mocked up whatever he wanted using Flutter. I showed it to him and he liked it. At the current time, Flutter web is not – they don't really have very good SEO. It only makes sense to use it for things where you have to have a web app, but you don't really need the SEO. After you log in, you probably don't need much SEO anymore. Some of that's actual web app, a PWA that would make sense.

Any case, yeah, I built it actually with web. I show it to them. They liked it, but they need SEO. I just said, “Okay, no problem. I just want to play with it, see how it looks.” Actually, one of their engineers actually came up to me and he said, “Do you mind to build an Android version of this? Because I know you did it with Flutter.” I said, “Sure, no problem.” I didn't do anything different. I just literally just did Flutter build APK and then kicked out the APK, gave it to him. He was super impressed. He actually also makes Android apps natively. He was super impressed with the quality of what I gave to him. I thought that was very interesting. I didn't go from mobile to web. I actually went from web to mobile in that case. There was no changes on my end. Just build it and give it to somebody.

[00:34:59] JM: Why do you think Flutter isn't more popular?

[00:35:02] AW: I think that Flutter isn't more popular because I think, people don't really understand it. If you look at the history, you had phone gap, you had Ionic, you got React Native, all these things. They just don't feel nice. They don't react nicely, especially a phone gap. You know when you're working with phone gap, you know when the app is using HTML, instead of using native components. You know it. You can feel it. You can touch it. It's just not a good experience.

When you say cross-platform, everybody just assumes that it's using similar as other technologies, where it's HTML CSS, or it could just be your JavaScript calling components, but it's a little bit clunky. React Native, I feel is a little bit clunky at times. It's getting better. At the same time, I think that's the perception is that people say, okay, when you have cross-platform, you're just not going to get very good performance and quality out of it. I don't think that's true at all when you use something that's Flutter. There's a lot of Flutter apps that people don't realize are actually Flutter.

I didn't know until recently that WeChat – I use WeChat quite a bit, because I have to deal with people in China. I didn't know it's actually using Flutter. To me, I was quite surprised at how well it all works. Yeah, there's quite a few apps that when you use it, you don't even think about it. Then once you know it's Flutter, you're like, “Wow, that's really interesting.” Now, it makes sense about why it looks so nice, and why the animations look so good. I think that's really as I said, people have a preconceived notion that cross-platform means that you're going to be – have worse quality, because you're sacrificing quality, native performance in terms of cross-platform, when that's just not the case for Flutter at all.

[00:36:53] JM: Is Dart a hard language to write? Or do you enjoy it more than other languages?

[00:37:00] AW: I enjoy Dart much more than writing JavaScript. I try not to write JavaScript. I can't remember the last time I actually wrote JavaScript, to be honest. It must have been at least six months, maybe even more. I wouldn't be surprised if it's been a year. Dart, I would say is a very fantastic mix of JavaScript and Java.

It's funny, because the reason for this is that the creators of Dart actually came – The first time that they work together was actually working on a Java hotspot, if you remember hotspot. They actually worked on that together at sun. Obviously, that's Java. Then their next thing they did was also Java, together some IoT company, I believe. Then they went to work on V8. That's

JavaScript, right? These guys have been in these two languages for quite some time. Then, they wrote Dart to try to actually make a better version of JavaScript.

These guys, they've been in Java and JavaScript for quite some time. I would definitely have to say that you don't even have to really even learn Dart to write Flutter. I didn't really look at Dart deeply until about six to eight months after working with Flutter, I think. I'm trying to think clearly, but it was definitely some time. That's because I actually started having some issues with trying to do more and more advanced things, and understanding why certain things or certain way and understanding how to do certain things, because I really wasn't into Dart so much.

Of course, to be really, really good at Flutter, you have to get into Dart. To be productive, you don't really have to learn Dart, because it's just – it’s just natural. I think it's quite natural language. That goes to show you, from – like I told you before, designers, which don't code at all are able to create very beautiful-looking apps. I just taught somebody recently, a couple months ago, how to write Flutter apps. We never dug into Dart at all. He's creating features for a client of mine now. That's the nice part about Dart, is it's really a great mix of Java. We have strongly typed language, and JavaScript where you have a language which you can still just write and things just work for the most part.

[00:39:17] JM: As we begin to wind down, what do you see in the future for the Flutter ecosystem? You see it growing? You see it changing? As it takes over more platforms, do you expect it to grow in popularity? What's in the near future for Flutter?

[00:39:32] AW: I expect it to grow. I think, I seen a stat recently, I don't remember what it is at the top of head. I think, I’ve seen something that Flutter is basically the biggest cross-platform framework in terms of people actually using it, if you compare it to others. Just the amount of apps that people are producing at such an alarming rate.

I can only see it getting bigger. They're just adding more and more platforms. They're not really adding platforms in terms of maybe making it, like bloating, but they're adding platforms, because it's just so flexible. I can only see things getting better. I can't think of ways they could actually get worse. All the people who are in the Flutter team have been great about making sure that things are actually getting better. Now they are adding features, but they're making things more and more smooth, and they're trying their best to really make things better.

They just announced that they're basically having their bug time. If typical time is 10 days, they're able to get it done in five days. I can't see anything negative going. I only see positive so far.

[00:40:35] JM: Awesome. Well, anything else you want to add, Allen?

[00:40:39] AW: I think that, don't take what I'm saying as – that it's great, whatever. Just pull it down and try it yourself. Go to dartpad.dev. Check it out yourself. Just play within the browser. If you feel up to it, install on your machine, just play around, try create a Mac App if you're on a Mac, create a Windows app if you’re on Windows, or just try Android or iOS or web or whatever. Just having a try and see how you feel. Really give it a try before you ever write it off. That's really what I think is important.

[00:41:10] JM: Allen, thank you for coming on the show.

[00:41:12] AW: Thank you.

[END]