Chris Strahl:

Hi, and welcome to the Design Systems podcast. This podcast is about the place where design and development overlap. We talk with experts to get their point of view about trends in design, code, and how it relates to the world around us. As always, this podcast is brought to you by Knapsack. Check us out at Knapsack.cloud. If you want to get in touch with the show, ask some questions, or generally tell us what you think, go ahead and tweet us at the @TheDSPod, we'd love to hear from you. Hey everybody. Welcome to The Design Systems podcast.

I'm your host, Chris Strahl. Today I'm here with Dana Lawson, the head of engineering at Netlify. Dana, welcome.

Dana Lawson:

Thank you so much for having me. Excited to be on the show.

Chris Strahl:

This is round two from a series of technical faults. We'll see if we get through it today. Being in the industry that basically builds the internet, you'd think that this would be easier somehow.

Dana Lawson:

No, this is how it goes, right. If only we could just bang the top of it like the old TV consoles. I miss the days when it was so analog that a little shake would fix shit. But now it's a lot more complex. But it is funny how many technical people it takes to do any kind of technical activity outside of their job. It's always a comedy of errors.

Chris Strahl:

Yeah, it's like you can't just pull the cartridge out of the game console and blow into it anymore.

Dana Lawson:

No, no, that always works. That always worked. They're like, "What are you doing?" "I'm fixing Simon's Quest?"

Chris Strahl:

Totally. Something about nine-year-old me saliva into a electronic thing worked really well.

Dana Lawson:

It really does. Now it's just not the same. It's just like, "Oh, can't fix it." I don't know if we actually fixed it back then, but stuff was simpler.

Chris Strahl:

It all seemed to work better regardless of whether it actually did or not. Well, any who, we got to go spend some time together in June out in the desert in Santa Fe at Patterns, and it was a super great time. And I felt like I got to get to know a lot of amazing design system leaders folks, and we're wanting to talk to a bunch of the key speakers and stuff that we had there to talk about your experience. So what did you take away from Patterns? What was the highlights for you?

Dana Lawson:

I mean, I don't know. And I'm not just saying this because I'm on the show, but in all honesty, it was one of the best get togethers I've had in a long time. It's been a long run since Covid and you get kind of isolated, especially when you are in a leadership position, to go and talk to other people experiencing the same problems you're trying to solve. And for me, just getting to hear from every spectrum of the design and product process that we're all in it together. And we have this amazing tool set that didn't exist five, 10 years ago, but how we utilize it. Kind of how we were joking about technology breaking, even to the point of being able to do this is just such a different ball game with all the integrations and in a lot of places, the way we're trying to remove friction from the process. And it was just a great time to reconnect, I don't know, not feel alone in the journey of making a better world for web developers and designers alike.

And I don't know, I had a great time. I still resonate with the people that I've met and the connections that I made. In fact, I might have hired somebody I met there. That's how strong of a bond... Honestly, that I formed that I was able to connect on a level where we wanted to partner together to go change the world.

Chris Strahl:

That's awesome. One of the things that I thought was particularly cool about your contribution to Patterns was that engineering leader mindset. I think that there's a lot of problems with the word design and design systems, many of which we've talked about several times on this podcast. But in brief, the idea that somehow this belongs to designers. And I think that the wonderful perspective that you brought was that engineers really care about the systems that are here. And these systems have been in place in how we manage code for a long time, but it's not ever really been present in that place that design and code meets. And taking your kind of love of systems and love of the way that engineering flows work elsewhere in the product engineering, product development process and bringing that into design systems. I think that's a cool idea and a cool concept. So tell me a little bit about how you think about the future of design systems as it relates to the relationship you as an engineering leader have with a design counterpart.

Dana Lawson:

I don't know. I'm painting that future right now. I'm not waiting for it to come. I'm trying to force my way down the path and pave that shit as I do it. And actually right now I have such a high belief in the ability to make a great experience by that tightly coupled bond is just really what we all strive for when we're trying to reach our user base. You want it to feel and look and just, ah. You want those moments. You want those moments where you're just like, "This is awesome. I'm doing the shit I need to do on it." And for me, I actually have the design systems team in engineering and it's so tightly bound with our design team. Half the time I'm like, who's doing what? But I don't really, at the end of the day, I don't care who's doing what as long as they're getting it done, I'll tell you that.

But that cohesiveness has just really helped us solve the problems we set out to solve with these systems. It's to bridge the gap between engineering and design and speak the same language. And what I love about it is now we're meeting each other in the middle and us engineers love the fact that we're using some of the principles as you've said, that we have in the past and thinking about, well how does this also just enable us? Right, because our goal is innovation at least for me. I want to utilize the design system so I can innovate quickly. I want to be iterative. We've been talking about agile for how many

damn years? I'm glad extreme programming didn't stick, but I'm happy agile programming, whatever flavor of it you do, because iteration helps you really understand the humans that use your stuff.

When you can iterate and have cohesion and a great experience, it just really helps you reach that end goal and it inspires you. And by tightly coupling these groups together because they're solving the same problem, they're not solving necessarily, I mean, we can be real narrow and say like, "Oh, I just want to make a ship product faster. I want to hit my OKRs." Cool, great. But what problem are you solving? You're trying to solve the magic that happens with the interaction of what you're producing. And as long as we can align about that, then the tools and the systems are just complimentary to the outcomes.

Chris Strahl:

Yeah, the way that you think about it is fascinating to me because you think about the focus is the goal. I can definitely see that, right. This idea that it's not about how you set up the system, it's not about how you set up the tools, it's not about, even though micro steps about how you actually build that collaboration, what it's about is about being able to say, "This is the thing that we ultimately need to ship at the end of the day. And that thing that we need to ship needs to be as free of distraction as possible so that we can all focus on that thing, instead of focusing on all of the ways that we have to think about building that thing." My co-founder, Evan, he says frequently, he's like, "An engineer will always find the path of least resistance."

And I think that that's a wonderful way to think about the structure of a design system is it sort of frees you from a lot of the bureaucracy, maybe that's the right word, the crufty bits that always seem to be in the way of being able to focus on the thing that you actually need to be building.

Dana Lawson:

It's that, right. And there's so many great studies about engineering focus and developer velocity and developer happiness and really just the state of flow, right. I'm some techie nerd talking about flow, but this has been in existence since humanity. And so it's been well studied that when people and humans are in a state of flow, they produce and it just comes to them. And we know from research that every interruption of flow takes 23 minutes on average to get you back into that state. So think about this, right? You're sitting down, you're like, "I got to build this feature and I don't need to stop and talk to a million people." Maybe it's in the PRD, maybe it's in Figma, but you want to go do the next step, hands to keyboards. And now you're sitting there going like, "Okay, do I have to sit and go read through all the minutia?"

Or in this case, having something with a little bit of bureaucracy just really takes all that away. You're going to be able to sit and say, "You know what? I know where to go. I know which well to dip out of every time because it's the same one." And as you say, engineers want the path of least resistance because resistance breaks you from flow. And flow is what makes you be amazing because you are just hitting all the marks. And now we're not always in that state of innovation and creation. Sometimes we just are in a state where we just have created that tempo and that rhythm. And if you think of it through that musical lens, we don't want freestyle jazz, we don't. We don't.

Chris Strahl:

But you also don't want classical, right?

Dana Lawson:

You don't want classical, you just want that nice spa music that you forget is on the background. You want Enya, you want Enya just to do whatever she does, right? And who doesn't love Enya?

Chris Strahl:

Well I think that the...

Dana Lawson:

Hey remember who you asked me to come on this show.

Chris Strahl:

I Did, I did.

Dana Lawson:

It's a Friday Chris.

Chris Strahl:

It's a Friday. It's before long weekend. We're all a little punchy.

Dana Lawson:

Yeah.

Chris Strahl:

Yeah. No it's great. So I think that the path to Enya as we'll call it, is paved in your lens with this idea that we have to create this flow state. We have to shed all of the things that have inhibited us from that. And when you talk about that, I'm reminded of times where you're searching through Dropbox trying to find the right file. Or you're trying to understand if the thing that you're looking at is the underscore final or underscore final, final or underscore, underscore final two of a-

Dana Lawson:

V3.1.2.

Chris Strahl:

Exactly.

Dana Lawson:

You're like, "I don't know."

Chris Strahl:

Exactly. And then you even look at something like some bit of code, right, that is built on a site. Is that actually how it still looks? Or do I need to actually trigger another full build, wait for 15 minutes and go have a cup of coffee to see if the latest thing is actually present? I think that this works on both design and engineering. I mean design gets a bad rap because things like semantic versioning haven't existed in

design traditionally, but at the same time there's tons of stuff in engineering where you're like, "Oh yeah, apparently I have to kick off a build process and wait 15 minutes to actually see the real thing and what's there isn't actually real right now." I think that that's ultimately a big part of what we talk about when we talk about flow is like how do we shorten our cycle time, have more iterations and have those iterations have less distraction and less time spent sipping coffee?

I'm always reminded of the xkcd cartoon where there's the two guys that are dueling on the office chairs and the manager comes in and is like, "What are you doing?" And they're like, "Oh, things are compiling." That's the engineering state of underscore final, underscore final two, like 17 folders deep inside of Dropbox.

Dana Lawson:

A hundred percent, a hundred percent we've all been there. And good or bad, you love that path of least resistance when you don't have something where you have defined constraints, guess where the path of least resistance for the short term becomes the paid path. But long term it's never, never the right option because people will run into the repo or they'll inspect the assets and then they'll go trace it back to the code repository and just be like, "Dude, grab that button." And that is going to get that out and shipped, but now you're going to have even more work if it's wrong, you're going to even have more work if it's wrong. Because we're going to look at final product after we waited for it to compile, and put the website up and go like what? That's not inversion. That's the challenge when you think about this and when you introduce new systems and new tool sets, because a lot of us have been in the world where it's like, "Oh my gosh, how many checks and balances do I have in the supply chain to deliver what I'm trying to do?"

And the reality is you want to have the right ones and things like this, if you do it correctly, people will use it. The challenge that I've done in the past is I've created bloatware design systems that was so hard to understand. People were like, "Forget it, I'm just going to go the least path of resistance, I'm going to go to the repo." There's two ways to always look at it, but I think it comes back to is over time, are you being kind to your future self as well? Right. Are you? Because I want to be because I know what's going to come back to bite me and it's a daily battle. I think we all encourage, should I do it? [inaudible 00:12:22] I shouldn't.

Chris Strahl:

I think there was a lot of design system voodoo for a long time and if you're a voodoo practitioner listening to this podcast this is not intended to be offensive, but the idea that you're sitting there rolling some bones and seeing what kind of comes out of the design system without that clear intention of what you're trying to serve. And I think that that's been one of the things that over the past three years we've watched the industry get much more mature around is this idea about ultimately what we're trying to serve is that ability to have tighter iterations. That ability to understand where truth lives and whether that truth is in design or engineering, and then very, very quickly get to that so that we can build things faster, better, et cetera. And I think that one of the things that struck me when you were talking a minute ago is the whole back trace a button into a repo and be like, Yeah, that little guy. Copy that little guy.

Dana Lawson:

Yeah, that's it.

Chris Strahl:

I think about that in kind of another sense of one of the things that has been wonderful that just before design systems really came out when you had Envision and early Figma and stuff like that is we got rid of the eyedropper tool that would sample colors off of websites.

Dana Lawson:

Oh man, I loved that tool.

Chris Strahl:

Oh yeah. I mean how many shades of gray or primary color did you end up with because somebody used an eyedropper to grab something?

Dana Lawson:

Oh it's so terrible. It's so terrible. Yeah.

Chris Strahl:

Just the simple elimination of something like that by putting a hex code in the same place where the design is, very few engineers probably ever actually copy that hex code out of Figma and then use that in a production thing because they had a CSS class that applied that for them. But at the same time they at least knew what that was and didn't have to go in eyedropper tools, some random block on a homepage somewhere.

Dana Lawson:

Oh a hundred percent. And I joke about it because front end instant my specialty. So I'd be like, "What do y'all want me to do?" But no engineers for reals, engineers, it's like speaking the same common language, right. It's coming and utilizing the common language, once again, thinking about the shared outcome that keeps us united. And I had the opportunity to work at Envision in the times when we were really starting to think about what is this new world beyond how we just create and share design, but how we store manage and then honestly collaborate on it with our engineering counterparts? Because that's a big part of it as well, is having that opportunity to collaborate outside of your wire frames outside of the first iterations. Because we want, at least in my opinion, we want to empower product minded engineers, right. We want people to be able to iterate quickly, but do it with the right cohesion, the right kind of foundation late in place and self-serve in the right way, especially when we're working early on and have that once again, like you said, that phase be just accelerated.

Just that phase between wire frames to actual production and then like you said, removing some of the things that were barriers and utilizing where we find those common language, whether it be hex codes or what, right?

Chris Strahl:

Yeah. You think about the idea of how did people go fast before? Right. That Mythical Man-Month, single mind concept, et cetera, et cetera. Right. Where it was like... In some ways it's straight up encouraged cowboying, the idea of, oh well you want to get something done, well tell two people to go get it done and remove all the rules from how they should do it.

Dana Lawson:

A hundred percent.

Chris Strahl:

But that doesn't actually scale. And so I think that that's what we've started to look at in terms of design and engineering collaboration, is we try to start to look at this as like, yes, we know that systems help us build better products. We know that systems help us build more efficiently, but why they do that is they enable that ability to go fast and that ability to go fast in a way that's cohesive instead of people being like, "[inaudible 00:16:06] motherfucker." It's a lot more controlled and understood and sanctioned. And honestly because everybody's in the same system on the same page, it's kind of innately blessed by everybody as a part of that system is you should be able to go fast, and you should be able to remove your constraints as long as you work within that system that everybody's agreed on is the way that we're going to manage things.

Dana Lawson:

I mean a hundred percent, right. And we're all on the same team. And the way that my team right now is currently structured is the three-legged stool. I know that's kind of overplayed, but I have product engineering and design, we win together, we lose together. I think we all say that when you're creating product, but I think by really having this understood way how we share and collaborate and having tools like this in place that enable those voices, because really we should all, once again, this is why I just harp on it, have the same outcomes on experience because if you have that type of tight coupling, it also gives you some back pressure on the things you're delivering. It's like, "Oh I hate hearing this." Oh my gosh. It's like, "Well I was delayed because I had to wait on design." I don't know how many engineers have told me that before.

Well I'm sorry. And you go to design and design's like, "Wait a minute." I need the PRD, I'm waiting on product, I don't even know where to get started. What are you talking about here? And let's just say this isn't something that you're building net new. Maybe you're just doing an update, right. And I'm like, "I can't believe we're having this conversation for the thousandth time." Because I want people to make informed decisions. I want people to understand the constraints. I don't want to totally say, "Oh work without what you need to create what you're creating." But on the other hand you have to separate what we like to say as boulders and pebbles and we want the pebbles to be almost self-managed so people can execute fast. But that's where you have the system in place so that you can, and then you can also make changes because as you make the change in the design system, those changes are going to cascade. So from an engineering leader, you always say "Don't strive for perfection, strive to get it right. Because we can always iterate on perfection."

But let's be real, you don't want to be hundreds of miles away from what that end product is. You know what I'm saying? It's got to be pretty [inaudible 00:18:19] a little shorter gap.

Chris Strahl:

Well it's the reason why everybody's changed from minimum viable to minimum lovable, right, is-

Dana Lawson:

Oh my gosh, yes.

Chris Strahl:

Minimum viable. Oftentimes they need to be a long way from actually usable.

Dana Lawson:

Yeah, it's true. It's like, "Oh man, MVP. MVP is usually not very good." So I love when that came out. I remember a product manager at New Relic when I was there told me that and I was like, "What?" And then I was like, "Yeah, I like that a lot better." It makes it just more human-centric, it just makes it more human-centric because not that developers fit the stereotype of just being very binary in the way of thinking where it's like we're not robots where it's like, "Give me the list of duties and I shall fulfill them." Believe it or not, engineers are creative humans because the art of product development or even systems, it is an art. We can all be given the same direction and produce a different outcome. But what we want when we build these experiences is to understand the correct outcomes. And so I think by enabling people then you're going to be more successful.

And I mean it's just another reason why I am super bought in as an engineering leader and I hope more engineering leaders invest that intention to the three-legged stool because that's where you start finding the magic happen.

Chris Strahl:

I completely agree that there's so much craftsmanship in engineering that is beyond just the whole, let me write some code. Engineers make design choices every single day. Oftentimes very difficult design choices and design choices that are often very hard to represent in design tools. And I think that what I heard you advocating for earlier is I heard echoes of my favorite sort of metaphor allegory for the creative review process where you have a designer that builds something and they hand it to engineers, it looks nothing like what they built and they go through this horrible tennis match of blame and regret between the different disciplines of, that isn't what I designed, well what you designed doesn't have enough fidelity for me to actually build the thing that you designed. So I built the thing that most represented your design. But what if I show you that thing? And then like, "Oh, but we need to do another iteration." And now months go by and then at the end of it all product's like, "But you didn't actually build the thing that was supposed to be built."

Dana Lawson:

That's right.

Chris Strahl:

That horrible swing set cartoon of how products get so screwed up in that horrible tennis match. And I think what I heard you advocate for was this idea that there is this system's point of view that doesn't look at things as handoff, right. It looks as accountability across the spectrum of the product creation process. And this is actually really interesting because this happens in all other sorts of businesses, right. Anytime... Marketing and sales is another traditional way where this happens a lot where like, "Hey, we need to sell more stuff." "Well marketing needs to generate more leads." "Well, sales needs to be better at closing." Well the reality is it's all of those things. And without somebody that has the ability to oversee that end to end within the same system, you would inevitably end up with these disconnects.

And I think what we're talking about is design systems being this operational infrastructure for how we think about how products get created that helps us normalize the way that we manage that process from

the very, very beginnings in product and requirements to design to engineering and all those little iteration cycles that exist in between. And that is really revolutionary. That's actually a new way of building digital products.

Dana Lawson:

I think it's the best way to bring people together because you rallied around a goal of the success of the work that you're doing. We're dedicating half of our lives to this thing that we call a job. Believe in it, believe in what you're doing. I know it doesn't have to be that grandiose or whatever, but I don't know. I want that nirvana. And I think it's exactly what I think the next phase of moving beyond the tool set. The tool set is an enabler. It's the culture that has to shift and I want to see that happening. And like I said, I'm not waiting for the future. I'm trying it out right now. Is it always roses? No, I mean absolutely not because we're all very highly opinionated people yet flexible. But that's where you come back and you try to understand from a bigger picture view and sometimes you're like, "Well, oh, it's going to be a trap." What is this decision by committee?

When you include the right counterparts, and I say, no, actually it's decision by information because you're so tightly aligned and you have these paved paths that connect you in your systems and you don't have to have a committee, you already know what you need to do because you're all aligned. And that's what we work for. That's what I work for every day.

Chris Strahl:

That's awesome. So how do you get people to trust and believe in that system? Because that's ultimately what powers this, right, is you all get in there and you all get in the same system and all of a sudden you're building stuff in a very different way than it's ever been built before. Instead of it being like it goes from this one part of the organization to this other part of the organization to this other part of the organization. And it's like [inaudible 00:23:17] waterfally kind of and maybe some agile pieces here and there. But instead what you're doing is you're saying, "All right, let's go to find the requirements of the system and then let's look at what we've already built before and see what we can use inside of our system that already exists. Instead of thinking about everything as a new thing, let's think about it as how we reuse our system. How do you get people to trust in that? Because it's a very, very different way than we've ever worked before.

And look, there are people that are doing this now, right but it's I think still very sporadic. You're not going to find a lot of organizations that are doing this at scale very well. And I'm curious when you think about this at Netlify, how do you get people to invest and trust in that system and how do you ultimately realize this nirvana? Even though I know you're very much on the pathway still and you're not totally there yet, what are the things you're doing now that help build this trust?

Dana Lawson:

I mean I think it's just really the investment on the engineering side. Ic They're the end users, they work so hand in hand and I think a lot of people don't want to invest engineering time on it. I'm like, "Why not?" It's hard for me when I know that I'm trying to once again keep people in flow and remove friction, not see this as a worthwhile investment. And I think by engineering actually stepping up and not just saying, "This is important and let me tell you why." You have to show the actual investment in it. And so I've done that and I think other people need to do that. But you still have the hard challenge of, oh okay, we're doing something radical, believe me, I live and breathe that.And it's like we got the team together, how do we do it? And I think like you said, it's looking at what's existed as we pull this together. For

instance, we're like, we should probably just start with a component library and glossary because we have all the fun stuff. So where do we start by finding the... Start with the low hanging fruit, start with the low hanging fruit. And I think finding that common ground and just like how you approach software development and when you're going and trying to create a new rhythm of work is you iterate on it, you outline it. And this is a part where you do have a little committee by decision because in this place you're going to have this team talking about the challenges, the risks, the skillset, the ownership, the accountability, the problems to be solved and then hopefully dreaming about the outcomes of those problems.

Because you know what, if you do this right, then we're going to ship more code and people are going to be delighted. And so you really have to be clear with the vision of what you're doing and how this will enable that vision. And then try shit to learn. I say try it out and that's why you reuse what you have and build upon or you know what I'm going to say it, you throw baby out with the bath water, you say, "I declare default." I am not against that. I have declared default occasionally. And sometimes when you have iterated and not produced the outcomes, do it. Clean the slate, simplify it, and then execute.

Chris Strahl:

Is that like Michael Scott declaring bankruptcy?

Dana Lawson:

Oh yes.

Chris Strahl:

Shouts it into the [inaudible 00:26:16].

Dana Lawson:

"I'm broke." I don't know, it's just call it like it is. Because once again, I think that builds trust.

Chris Strahl:

One of the things I love most about our conversations is the trust through vulnerability is ever present with you. And I think that it's this wonderful side of your personality that just engenders this earnestness in people. And I think that when you're out there painting that vision to the people that you manage and then the people that manage you, how does that message look managed up and managed down in the organization? And is there a difference there? Because I mean I know you are like... When you talk to Dana you get Dana and that's awesome, but what do you say to those executive leaders at Netlify and then what do you say to all those engineers and designers at Netlify when you talk about this vision?

Dana Lawson:

Oh, I mean they're just like, "Oh she's preaching again." I mean I live enthusiastically, I have bad moments. I really try not to have bad days. And it does change. Some people are like, "Oh." I'm on the spectrum of you love me or I annoy the shit out of you. It's a hard middle ground with Dana Lawson. So knowing that I just have faith that people can embrace that I am not full of bullshit. I actually do believe in this, but I have a great team that also has shared alignment on the vision. So the way that they need to curate the message so their audience may slightly be a little more tactical versus woo woo dreamy. Or

if you really have somebody that's like, "That is so ambiguous what she's saying. We're painting dreamland here." Then you have to swivel that ambiguity down into the details.

And I think when you get into details, you have a different form of communication, right. When I'm in project management mode, it's like whoa, here's going on. But when I'm trying to rally and really get creative juices flowing, like, "What are we doing? Why are we here and how do I feel?" Right, how do I feel about this? Then we can dive into the details, but I think it's about a repeatability too. And then measurement, do you have a form of measurement? Because the engineer in me, I'm still an engineer at heart, is we make informed decisions by the outputs that we define. And measurement's not always perfect. I always joke, OKRs sometimes can be super bloated if you're immature at them because it's like you can have any measurement, does it really matter? It's like, "Okay, I'm going to call the median here and just make it 10% better."

But if your median is the worst of the worst, I mean still you're going to see what you need to see, but you're going to be winning. So long story longer, I think that in these cases I have to back up this talk with actual outcomes and that's where it's a balancing act. And once again I try to just over communicate, I work with a distributed team and so you just don't want to assume stuff. We have 17 countries represented at Netlify and a lot of people are English second language. And I use very silly language sometimes. I've been working on it forever. It's just part of who I am. I'm like, "Oh yeah, people don't know what britches are." I'm like, Britches are pants. People used to call them that." You take these things for granted. So like I said, long story longer is you have to ensure that you work on that clarity through the ranks and that the people also understand once again what needs to happen, but they're talking to their people. So I think there's a big amount of leadership that you just have to continuously instill upon people.

Because good ideas come from anywhere and anybody can lead and it's your job to align them.

Chris Strahl:

I think that that's one of the fun things, right, is explaining to executives that britches are pants and explaining to engineers that britches are pants. You have to explain it to both audiences but you explain it in a slightly different way where one of them has a graph that basically says the number of times that Dana has used the word britches in the past six months. And then the other one is just literally a key value design token pair that says, "Britches, pants, same thing."

Dana Lawson:

Exactly. Same.

Chris Strahl:

What is that flowery, very data driven but still kind of dreamy sort of idea? And then the other is a very practical place and maybe it's also paired with data of this is the number of times we use britches in our design system.

Dana Lawson:

Exactly. And so it just kind of normalizes, right. At the end of the day, you're getting to normalize your workflow and when you have a unique set of people all invested in the idea, that's what you're doing. You're constantly trying to normalize it so that you don't squash those different ideas. But at the end of the day, that's what we're here for. And it comes back to what are we doing? We're trying to create

software that changes the world and we want to do it in the right way. And if we can iterate and ship faster and be on the same page, I don't know anything's possible.

Chris Strahl:

So at the risk of doing another wind up, because I think that this could be a fun one, is when we think about that leadership thing you mentioned, right, how leadership can come from anywhere. When we think about that relative to engineers in the context of a design system, you and I have talked a lot about the problematic aspects of ownership of a design system and how the single discipline, single owner model, how the amount of balderdash or hogwash to use a difficult term. The fact that that doesn't really fly. If you're an engineer at a company right now, like Netlify or somewhere else and you're trying to find that leadership voice or that leadership avenue, what do you do? What advice do you have for those people?

Dana Lawson:

In the context of design systems, understand the rings of influence, right. There's this true sociological study, and I don't have it written on a napkin right now to tell you who, but you have one, two, three rings of influence. And the third one is you have no influence, stay out of it. And that's where we have the rub, right. Where it's like, "You're not a designer, you don't work on the design ops team, get out of here." The same could be said in other parts. And I think what you do is you think of creating that space, right. Maybe you don't have this influence because you're outside of it, but you have an opinion and you want to participate. Or maybe you actually are on that team but you don't know how to surface it. Find who's the number one. Because the number one is the decision maker. That's more difficult in larger organizations.

It is. And I'm not saying go slide into somebody's Dms, or maybe I am, I don't know, be bold, go do it. I have definitely slapped people I probably had no business doing it that were much more important than I understood to be. But you know what? Life's too short. I think if you come armed with information and clarity on why you're trying to do it, but it comes down to does your culture enable that? I've been so blessed to work at places like GitHub and Envision and Netlify that really you do want to have good ideas coming from anywhere, but these are creative fields and they respect that engineering is a creative practice. And I think there has to be two parts of that, but understand who is the person that owns the problem.

But like I said, it's really hard when you're just doing something new in the whole company is trying to rally behind it. And that's where it makes it a little more difficult because you don't know who is responsible yet sometimes. And so-

Chris Strahl:

Right. And there's also sometimes a microscope.

Dana Lawson:

Yep. Sometimes there's a microscope. And so I think finding out really the right ways to bring the feedback up, but being bold and just doing it, but don't just come in and say, "I think it should be this way." I'm like, "Yeah." What do they say? Opinions are like assholes, everybody's got one. Tell me why your idea is better and then why this is going to change. And once again, if they don't have a good answer, they'll burn their own capital on that. But I think as leaders, we want to encourage people to

lead, be bold, share their ideas. So you have to build that culture and that safety to be able to have that happen. But there's definitely ways to make it possible. Understand and ask the question, how do I bubble up my questions and do I have a culture that enables me to be heard? And if you have that, then just go for it. But be articulate.

Chris Strahl:

Dana, always awesome to have a chance to talk with you. Love the energy. I can't wait until we get to go wander around in the desert again sometime soon.

Dana Lawson:

Oh my gosh, me either. It's been so fun and I know I'm so verbose, but I love this stuff man. And so I appreciate you having me on the show. And yeah, look forward to seeing you again in person.

Chris Strahl:

Sounds good. Talk soon. That's all for today. This has been another episode or the Design Systems podcast. Thanks for listening. If you have any questions or a topic you'd like to know more about, find us on Twitter #TheDSPod. We'd love to hear from you. Show ideas, recommendations, questions or comments. As always, this pod is brought to you by Knapsack. You can check us out at knapsack.cloud. Have a great day.