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 @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 Todd Wilkens. He's the CPO of Remote. Welcome to the program.

Todd Wilkens:

Hey, thanks.

Chris Strahl:

So, tell me a little bit about your background. You have this kind of interesting progression, where it's not really about design or engineering or product individually. You kind of had this movement you made in your career. Just kind of give us a sense of where you come from.

Todd Wilkens:

Yeah, so I'd say the short version is, I was always really interested in people, and I always liked making stuff when I was a kid. And, so, I've sort of just been kind of exploring those throughout my career. Right? So, I got a Comp Sci degree, but I got really interested in design. So I was doing a lot of front-end prototyping. Moved over to being an interaction designer back in the day. Did a little stint as an academic social scientist, studied sociology for a while, which was really helpful for me because I learned about people a lot, but I also ... It made me really ready to be a leader in an org because I think about orgs.

And then I kind of came back out of academia, went through design for a while at a few different companies. Adaptive Path was one of my main ones there. And I worked at IBM, worked at Atlassian, and then I sort of switched to more of a general executive leadership product kind of role. I went to work at a company called Automattic. I ran a line of business for them, called WooCommerce. It's one of the biggest eCommerce platforms in the world. I was the GM, so I ran everything there. I stayed with the product, so I moved to a CPO role after that. And I'm in a CPO role now, but I usually, the last two jobs, I run product and design because I see them as so clearly related. I've experienced a lot of different pieces of the puzzle.

Chris Strahl:

So, one of the things that you said before the show that I thought was really interesting, is this sort of diversity of experiences and focus on people has led you to really value the cross-functional perspective inside of your organization. And that's, I think, not totally uncommon, coming from a product person, but I think you have a little bit different take on it, specifically as it relates to the practicality of systems in your organization.

Todd Wilkens:

Yeah, that's absolutely true. I used to say this thing to people when I was in the design at IBM, which was, we were trying to work with all these product managers. And the designers would be like, "My product manager's doing X or Y or Z, and it's frustrating to me." And I would be like, "Well, why don't you become a product manager? You could take that job. You're totally smart enough. You have experience. You should go do that job." And I was like, "But you have to be responsible for the things that they're responsible for if you do that job."

And it was funny, 50% of the people were like, "Yeah, I'm going to go do that." And then 50% were like, "I don't actually want that job." And I was like, "Well, then, don't complain. Be a good collaborator." Right?

What was good about that phrasing was, it points out that designers don't ship things by themselves. Product managers don't ship things by themselves. Engineers don't ship things by themselves. Right? It takes a cross-functional team. And that's that moment when I started realizing, I was like, "I'm in the design organization, but is that really the only thing I care about? Is that what it takes to get the impact?" And I was like, "No, it's not."

And, so, then that leads you to systems. It's like, "Well, how do I create ways for people with different backgrounds to work together?"

And then there's all sorts of ways that that plays out. Right? Design systems is one of the ways, but there's all sorts of other kinds of systems that you need to use, when you want to bring the expertise of bunch of different people together.

Chris Strahl:

I think the fun thing about that statement is, it really goes to show that, without that collaboration being present across multiple different disciplines, nothing ever gets shipped. You don't actually create something. Of course, you create the comp, or of course you create that line of code, or of course you create that team of people, and you make a bunch of Jira tickets or whatever. Right?

But the reality of it is, is what it actually takes to build something that is useful to a user is inherently multidisciplinary.

Todd Wilkens:

Yeah. We have this phrase we use at Adaptive Path. I wrote this book, and one of the phrases we used a lot was, "The experience is the product." Right? And, so, what we're doing is we're shipping experiences. Right? I always like to think, no one is ever going to use your comp. No one is ever going to see your line of code. No one is ever going to see your PRD or your roadmap.

What your shipping is an experience, and all those are just means to the end of delivering an experience to somebody. And that experience could be software, but it could be something else. Right? It could also be delivered by humans or it could be a physical product or there's a lot of ways that it ... But the end result's the experience.

And, so, what's important about that is that, no matter how good you are at the craft of the thing that you're doing, the chances are, there's nothing that you're actually making that's really literally experience, unless you think of a maître D at a restaurant. That's a person who's literally delivering the experience. Right? But it's very seldom that it's not mediated. Right?

And, so, therefore, that's the collaboration issue. It's like it's the fun part of bringing all those pieces together, but it's also the challenge because everybody makes a thing that's not actually the thing that you're trying to get to people in the long run.

Chris Strahl:

No, it's interesting. There's so many analogies or, I guess, comparisons to the construction industry or to the auto industry or to playing with Lego sets as kids. Right? And we think about the common thread of all those is, innately, an architect isn't a builder. A car designer sculpts with clay before anything ever hits an assembly line or production line floor.

And there is this sort of mediation, where there's this liminal space between that creative process and the end result of that thing. And I think that that's what we're all trying to solve for at a very philosophical level in this industry, is how do we take something that is innately creative and get it into the medium it's destined for in the most efficient, effective way possible?

Todd Wilkens:

Yeah, that's a really great way of saying it. The thing that makes me think of is why ... jumping ahead to something we talked about before, which was why the heck, when you talk about design systems, is it called design systems? It could have been called something else because it's usually cross-functional, right? Sometimes, I think the design superpower is taking intangible things and making them tangible. You take something intangible and make it tangible, and it's usually not the thing itself that you're making. You're making a representation of what the intangible idea was. Right?

But once it becomes tangible, you can manipulate it, you can evaluate it, you can do all sort of things with it. Right? And, so, the thing that's interesting is, when you talk about design systems, it's interesting because a design system is really this opportunity to take something very intangible. I can imagine the patterns that make a great product experience. And you have to make them tangible, in such a way that other people can manipulate them and evaluate them and evolve them, but that they're still flexible enough that they can be applied in many different situations. Right?

And that's a designerly skill, is that intangible to tangible. And, so, it's probably the reason why we refer to creating these tools, the design systems approach within product development, a lot of the major steps came from designers. Right? Because it's their superpower.

Chris Strahl:

No, and I think that there is an organizational leadership quality to designers around stepping outside of the boundaries of process. Designers, almost famously or infamously, hate to feel constrained in the way that they work, because that constraint often limits creativity.

And I think that one of the powers of design here is, yes, of course, that original idea, given form, is this kind of magical creative process. Right? There's a lot of really cool, almost tingly sensation you get when you think about, "I have this thing that's in my head. Now I'm going to give that form, in a way that other people can see what's in my head." And I think that's really cool.

And beyond that initial form, what I think that is, is really an expression of intent. And I think that one of the things that we get caught up with in modern tooling, like Figma and everything like that, is we oftentimes think that giving it form is giving it its final form. And I think that the reality is, is now what we're starting to view with design systems, is that form has to be flexible. It has to be usable in a lot of different contexts.

And instead of pushing this idea of pixel perfection or "We need to make the most brilliant, high-fidelity thing possible," it's "No. I need to create an expression of intent of what's in my mind and giving that thing a form that can now be used by others to go create."

Todd Wilkens:

Yeah, there's a lot of things that have been built before. And, so, there's a lot of best practices. There's a lot of things that don't need to be reinvented. Right? They do need to be given the right sort of flavor and instantiation for any given product or culture or area. But you're not really fundamentally recreating the basics of type-size relationships when you're trying to ... You're not fundamentally, at this point, recreating how contrast works in palettes so that it's accessible. Right? You have a lot of opportunities to choose different palettes, but the rules around what kind of contrast you need are very well-understood.

And, so, there's a lot of these things that don't need to be reinvented, but they do need to be reused really regularly. And, so, what I think is really interesting here is, it's the idea that you want to focus people's creative energy on the things that are the hardest and most ambiguous problems.

And, so, what I think is really powerful design systems, and we talk about creativity and the constraints, is that what you actually don't want is people spending a lot of time making pixel-perfect prototypes or sketches of ideas. What you really want them doing is saying, "How the heck should I make the most sense of this thing to a person? And how can I explore the widest set and the most varied set of possibilities for a solution that I possibly can, but only on the things that matter?"

And the things that matter are not usually the color of the buttons or the shape of them or the particular little micro-interactions. Because once you've figured the micro-interactions out, they're golden. Right? The thing that's hard is like, "Wow, is this five steps or is it three?" Or "Should this be a mobile experience, or should it be a web experience?"

Those are the questions you want people working on. And one of the things I really love about design systems is that it gets almost like the micro-craft creation out, so that you can focus on the really meaty problems, as a cross-functional team.

Chris Strahl:

How I sometimes joke about it is, they're fundamental physics to design at some level. Right? Like gravity or inertia. And that's things like contrast ratios and structures of typography and the basic fundamentals of systems and grids and fills and all that sort of stuff that represents how you actually build UI elements. And those fundamental physics are something that we shouldn't ever try to reinvent.

And then there's a lot about those expressions of those fundamental physics that also probably shouldn't be reinvented. I don't think that anybody's really successfully re-imagined a button in a meaningful way that has changed the world in quite some time. And maybe we should not try to do that. Maybe we should try to focus on, "Do I need a three-step process or a five-step process, or what does this experience look like in load order on a page? Or how does animation affect my engagement with this particular moment for a customer?"

I think all those are those things that there is still a lot of untrodden ground to go and explore in. But the places that are well-trod, I think that's a lot of the power of design systems is, let's take all the things that have been predominantly mundane decisions for some time now, or are really fundamental physics and laws of design, and let's codify them, and let's make it so we don't ever have to think about those decisions anymore.

Todd Wilkens:

Exactly. Attention and time, there's the least of. Right? And, so, it's like how do you put your attention and time on the things that matter the most? And they're usually the most ambiguous things, rather than the least, that will matter most for putting your time and attention on.

Chris Strahl:

So, I think what we're really talking about here is this balance between efficiency, time spent, cost of things, quality of things, these sort of balancing scales that exist within pretty much every single organization. And what we're trying to say is, we have this solve for a lot of these baseline problems that exist.

And I mean, we're not necessarily just talking about design, Right? We're even talking about, "Where the heck is my design file?" We can solve that now with design systems. Or "Where's my linting script, or where's my build process for something?" Right?

All those can be elements of content inside of a system, but ultimately, I want to know what you care about most in that balancing of the scales in your role, because you have this interesting position where, for the past, I don't know, decade nearly, it's all been about, "How do I build more product faster?"

And I think that that mentality is starting to shift, and you're starting to say it's not necessarily just about the speed. There's this element of quality and efficiency to it too, that also matters. So, give me your point of view on that.

Todd Wilkens:

Yeah, so I got two answers to that. The first thing you said was about my perspective. And I would say, I have to admit to anybody listening here, which is that, while I come from a design background and was an actual designer, doing design and design research for a long time, I definitely have crossed the threshold, and I am a heartless executive now, who mostly just cares about getting things done.

I have a perspective that I bring, and I believe in the value of design and the value of product management, but ultimately, I'm just trying to get great outcomes, very efficiently, from a set of people. Right? So, just wanted to say that from the beginning, so people can keep that in mind.

Chris Strahl:

See, you wear a suit now, but ... Well, you don't, because you're not wearing a suit right now but you, you're a "suit," but you still have that design hoodie somewhere in the closet.

Todd Wilkens:

Maybe that's a way of framing it. Yeah. Yeah. So, the reason I said that is because the other part of your question, which is, everybody was focusing on sort of quantity of outputs. Right? It's one strategy for creating great, especially software. It's the idea that, "Well, if I don't really know what the right thing is," which is often the case, "I'm going to move quickly and iterate as fast as possible and learn by shipping things."

And, so, what that creates is a kind of general, quantity-focused measurement sort of approach within organizations. "How much are you shipping? How many ink commits did we have, and how much code went out, and how many features went out?"

But what's interesting is, what I have determined over time is that, clearly, I don't actually care how much you ship. That's my ruthless executive part. I don't care, really, if you shipped one thing a month,

but it was incredibly impactful. It grew our users, and it creates our CSAT, and it increased efficiency for various things. That'd be awesome. I wouldn't care. The business doesn't actually care how much you're shipping, how quickly you're shipping. But it cares about the impact. Right?

And, so, that's what I was going to say is that's interesting is, you really want return on investment. How much time and energy and people are putting into something, and what's the impact I'm getting for my business and for my customers?

And, so, then you have to start thinking about efficiency. You have to start thinking about efficiency, not pure efficiency, in the sense of just "Can I make something go faster?" But efficiency almost in the physical sense, energy in to energy out, how efficient is something.

And what you really want is you want to say, "How little energy can I put into something to get the most value out of the outside of it?" Right?

Chris Strahl:

Show me what you mean by that. When you think about that energy in. What makes that energy, as a concrete thing?

Todd Wilkens:

So, it's like I have a team. There's a product manager and a designer and four engineers. And they're going to spend four weeks on something. That's energy in. What's going to come out at the end of those four weeks of their time and attention? Right? They're going to ship something. And the question is, is it going to increase conversion by 5% or 50%? Or is it going to double it? So, from my perspective, I want four weeks, double conversion. Right?

Four weeks, 50% improvement is great, but I really want four weeks, double. And, so, the question is, how do I increase the chances that that four weeks is going to double conversion, as opposed to just 50% it?

And that's the thing I was saying before, which is, the best thing I can do is to make sure that those people, who I've hopefully hired, very smart and capable people, spend the vast majority of that time solving a really hard, ambiguous problem and very little of that time solving something that's actually already kind of solved, like "What color are the buttons, and where do they go?" Or "What's our way of doing linting?" Or whatever. Right?

That's the thing. Because they're smart people, I want their brains on the hardest thing possible because it's the highest, gives me the highest likelihood, that I'm going to double my outcomes, instead of just 50% improve them.

Chris Strahl:

I love the way you phrased that. I also think that it's interesting that you didn't say, "I'm going to add 50% more people, to increase the likelihood of that doubling of conversion happening." Right? I mean, I think that, obviously, you need to have people to be able to solve these problems, but you think about solving the problem of this in an organization at scale quote, as one of leverage of a system, not of leveraging more people to solve the problem.

Todd Wilkens:

Yeah. Well, Marcelo is one of the founders of Remote, and we were chatting the other day about this. We have almost 1000 people at Remote right now. And we are having this conversation, and we were like, "The dream is to be able to do as much of this work with 400 people as possible." Not because we want to lay people off, but because the dream is that you can make them just so effective and so efficient that they can just get lots of outcomes. Right? Or that we have 1000 people and, all of a sudden, we can produce twice as much great customer value, without doubling our size. That's another way of looking at it. Right?

And the only way to do that is to think about efficiency, but it's a kind of efficiency. That's the reason why I use this ROI energy in, energy out metaphor. Because sometimes when you say efficiency, people start to pay attention to ... They do these weird, stopwatch-type time things. I don't care about that. I care about the ROI, but very explicitly, I want to take a four-person team or a six-person team, and I want to double the impact that they have.

Because the bigger the team is, the harder it is to coordinate. Right? So, I would much rather have a system that is reliable and consistent and less people, rather than more people and of somewhat haphazard, non-systematic approach.

Chris Strahl:

And I think that this has been a big part of, frankly, the problem inside of product organizations, and especially in enterprise, for a while now, is this idea that we want to show the value of design in our product. We want to improve that experience that you were talking about. Right? And, so, our experience metric needs to go up. Well, what do we do? We quadruple the size of our design team.

And in doing that, the problem then shifts from, "We don't have enough resources to meet that demand," to "We don't have the systems in place to get those people to work well together."

And that inability to bridge that gap between how does my efficiency relate to the way people work, to the efficiency relating to improving experience? So many product organizations and product leaders I talk to are focused on, like you said, the sort of stopwatch metrics of, "Hey, in a 40-hour work week, how many lines of code do I ship? Or how many comps do I create?"

And it's that very production-focused idea that, "If I just build more, my experience will improve."

And I think that the wonderful part about this conversation is hearing a very product-centric way of understanding how the ROI is really not about those immediate efficiency gains in the activity, but ultimately, are related to the experience gains in userland.

Todd Wilkens:

Yeah, I use this as a concrete example. This is something else we talk about, in my company in particular. So, my company, we do payroll. Global payroll is one of the main things we do. So, we have people who run payroll in 60 countries, and so they have these specialty understandings about regulations and all these places. Right?

And they're really specialists. They're experts. And we talk about this all the time, which is that some of them spend 50% of their time sort of manually checking things and moving things around, which is the worst use of their expertise. Right? But someone has to do that thing, until we automate it.

So, we're automating those things all the time. And, so, we're constantly getting them to spend their time on what's expert for them. That's the thing. Right? The two things that are in short supply at any organization, generally speaking, are attention. It's not time, exactly. It's time and attention. You want

someone's brain cycles. There's a shortage of enough brain cycles, and there's a shortage of expertise. You don't have enough people that are skilled. Right? Everybody's always trying to hire more people that are really skilled. Right?

So, the most important thing you can do is hyper-optimize for the experts putting brain cycles on the thing that has highest impact. That literally, that's the math. And it's true for engineers. It's true for designers. It's true for product. It's true for sales. It's true for anybody in your organization. Right?

And, so, the thing I love about design systems is if you do them really well and depends on how widely you ... is that it helps every single person, not just designers. It helps every single, expert person in your organization, for the most part, focus their expertise on the more important things.

Chris Strahl:

I think it's a really powerful statement, and it shows the value of design systems, well beyond design. And I think it's interesting, also, because we've had these within single disciplines for a while now. Right? I hearken back to the DevOps days of yore, when every, single company was trying to hire DevOps engineers everywhere and every way they could, because the way to scale infrastructure was with highly-skilled, highly competent people, tackling big infrastructure problems.

And now, we have orchestration software, and we have things like Spinnaker and Kubernetes and dozens of Cloud products that all go and help us with this, from a systems perspective. And it's not that the DevOps discipline has necessarily gone away. It's that it's very much changed, to be one that isn't focused on how do we solve that same problem for every, single company on the planet, but how do we think about the systems that we use to manage infrastructure for our applications?

And I look at that as very much a similar kind of space as what design systems is becoming, where you have this idea of, there was this problem where we tried to solve it with people at scale. We realized that that created a whole new set of problems, so how to even get those people to work together at scale? And now we started to focus less on what the job of those people is, and more about the systems that empower them to do things the right way.

Todd Wilkens:

Yeah. Or the other way I think of it is, on the one hand, there's not a one-size-fits all. Right? So, you can't have one design system or one DevOps approach, set of DevOps tools and practices, that works for every organization because the things that they have going on, or they have different needs.

So, you can't just make one and say, "Oh, that'll work for everybody." That's one end of the spectrum, which is everything is totally standardized.

And then on the other end, well, everything is highly bespoke, which means we just need a lot of people. And each and every day, those people are doing manual, specific decisions and doing things in a really bespoke way. You sort of assume that there's zero consistency across contexts.

We know that that's not really true either. Right? So, it's this middle ground. That's what a design system is, but it's why there are multiple design systems. It's not one design system in the world. You put together a design system for an organization because there are specific things there, but they're also generalizable, at least within that context. Right?

And it's the same thing with DevOps. There are common practices with DevOps about how you do things, and there's common tools, but the way you set up a DevOps infrastructure for something like Atlassian. I worked on HipChat. So, it's a real-time messaging, like Slack. Right? The way you set up

DevOps there and what you have to handle, like the throughput of data and what an uptime means, and the way you handle that kind of stuff is very different than when you set up DevOps for something, I don't know, like-

Chris Strahl:

Like payroll software.

Todd Wilkens:

Yeah, like payroll software. Or you just have to solve it differently. The nature of the problems are different, but you're still using patterns. And once you've set it up, it actually works across your whole company.

I think that's the other thing that's really interesting here is this, I feel like design systems is kind of that sweet spot in the middle. Right? It's the, "How can we look for the generalizable patterns, but acknowledge that it's not one-size-fits-all across all organizations?"

Chris Strahl:

Yeah. I mean, within my context, as a company that provides the design system application, I think about this a lot. Right? There's oftentimes people that ask us, "Well, why don't we just have a starter library of components that could work for everybody?"

And the answer is, because creating that is actually a really hard problem. What does everybody's card look like? That's basically an impossible question right now. And there's also a lot of this kind of headwind to this idea of, "Why don't we just have design files generate code?" It's like, "Well because not everybody creates the same code from the same design file."

And, so, this idea that we want people to have the freedom to build things in the way that most professionally suits them, have really talented engineers craft the things that they need to craft, have really talented designers design the things that they need to design, but then have a place that houses that as content, that is portable, from company, to company, to company.

And I think that that's kind of where we've landed in this space, is this idea that the wrapper, the infrastructure, the general function of a design system, should be relatively similar, but the content you put in it should be able to be widely varied, in terms of its applicability and its use.

Todd Wilkens:

The problems the design system solves are fairly universal, but the solutions are particular. Right? Everyone needs a type ramp. Everyone needs a card. Everyone needs a set of guidelines around animations and micro-interactions. Everybody needs buttons and forms, but they also need, how do those forms have error messages?

Those are the problems. Everybody has those same problems to solve, but the solution that you find in your given organization is going to differ. Right?

Chris Strahl:

Absolutely. And it's amazing the amount of variation that we've uncovered, just even within this podcast, around how people are thinking about these things inside of their organizations. You have folks that they're at a point where design systems are just how they think and how they work, and they can't

imagine life without them. And then there's other people that are just trying to get people to use the same freaking color palette across their entire organization. Right?

And it has a lot to do with maturity. It has a lot to do with how people think about the servicing of those individual needs. And then it also has a lot to do with environment, to harken back to talk about design thinking, is one of the parts of design thinking is assessing the environment that you're trying to build that system within.

And lots of different organizations have lots of different ideas of what service design they're going to have, what types of interactions they're going to have between people, how those different departments are going to talk with each other and work together.

And I think that's kind of one of the brilliant things about this is that there's not a right way to solve all these problems right now, but the outcomes are this idea of efficiency that is grounded in experience, not grounded in this perpetual metric of, "How productive am I or how much activity am I doing?"

Todd Wilkens:

Remote is an interesting one, in the sense that, like I said, there are many things that are not mature about our organization because we're pretty young, but we have a surprisingly mature design system. And I was so pleasantly surprised. And, so, what's interesting is, that it's just a fundamental part of how we work.

And the other thing that's interesting is, that I don't think there was a time before that. Right? So it's not like there was some period of time, where the organization existed without having a design system work from. It kind of always did.

And, so, it's been a really interesting, almost social experiment for me, coming from other organizations that are not like that, to find one where, literally, they can't think of working in another way.

Now, is it implemented always the best way? No, and it was definitely led by the design team, so it has a tendency to be much more alive and Figma. We're working on how to create more of the Storybook Dev tool-type stuff, where we can have consistency in the code. But we still use shared components. It's just that you find that people fork them a little bit, or they overwrite or override things in them somewhat more regularly than you would like.

But it's still, no one would ever think of starting from scratch on something, and no one would ever think of not referring back to our design system when they started on something.

Chris Strahl:

Yeah, as the company matures, because the company's changed a lot in the years that it's been around since the design system was created. Has the design system kept up with scale, or are you feeling like there are cracks beginning to show?

Todd Wilkens:

Yeah, so we are actually in the process of just revamping the design system right now. So, I think it lasted our first year-and-a-half, two years, pretty well. But now we can sort of see where things are breaking. And most of it's because things, like our global navigation, we each have more things in our product. So, the global navigation, the IA, it doesn't quite hold up to the things we're trying to share with our customers. Right?

So, we're rethinking our global nav. We're rolling out a new brand, and so there's a whole bunch of visual styling that needs to be redone. And, so, we're taking opportunity to think through some of these at the same time, which affects type and color palette and that sort of thing, where we're making sure we're much better with accessibility, at least visual contrast and that sort of thing, and trying to address screen readers.

But then we also see things like our product is a lot of tables and forms. I mean a lot of products are, but ours is especially a lot of tables and forms. And, so, we're spending a lot of time rethinking how tables work and how you search for and access and filter through data. We've hit limits on our original designs and plans for that, and so we're redoing it. But we're doing it in a very systematic way, so that there's a very clear pattern, and we've got flavors for the different kinds of areas. Right?

There's a lot of things like that. And then there's some things we never had, that we're now building, but we're starting by building them with a systematic approach because we know they're going to need to be part of the design system. They're going to be an extensible thing.

Chris Strahl:

When you say that, it makes me think about that being more about the content of the system than the system itself. And it sounds almost like the system has really empowered you to be able to make a lot of these decisions and not have to think about these within the context of a broader, "We need to completely rethink our app." Right?

You're thinking about very specific, pointed bits of your application, these patterns that you're recognizing aren't meeting your needs anymore. And your system is allowing you to evolve those patterns, one, in isolation, two, in a way that you can still think about it systematically because you've sort of created that culture of that approach inside of the organization.

Todd Wilkens:

Yeah, that's correct. And also, I mean, you never get this perfectly, but we actually are seeing some of the beauty of having a design system that's highly coupled with our engineering, our front engineering, in that some of the changes we're doing, literally, just someone goes, has to go in and make a couple of quick changes, and it rolls out across the entire product. That's the dream. Right?

And we actually are experiencing that dream coming true, in a lot of situations. It's actually relatively straightforward for us to roll out these major changes to these patterns. And it's not just visual changes. I mean, even functional changes, like the way that little micro-interactions in our form elements help you want to dig in and get some more information or help about a particular form of question or part of our app. Those things get automatically changed and fixed as well when we've changed the patterns because we actually have taken a systems approach.

Chris Strahl:

So, what advice would you give to somebody that is sitting in a product organization, in a leadership role, that is sitting there looking at an environment where the pressure on them is still to scale, scale, scale. And that's not going away. But the economic environment is such that they're looking at not being able to expand their team and not looking at headcount, and they don't have this baked-in systems approach inside of their culture.

What would you tell that person? Or if you were sitting in that chair, what would be the things that you would try to push on, that would help make a better future for those folks?

Todd Wilkens:

So, it'll never work unless product design and engineering are all bought into the idea. And if you have a different organization that other parts to it, marketing or QA or-

Chris Strahl:

Cross-functional's essential.

Todd Wilkens:

... whatever. Yeah. Yeah. You have to start off with that, because, while you can gain efficiencies just from purely UI pattern systems, you don't really see the improvements unless it's kind of codified in code, and unless the product managers, unless they're bought into the fact that this will improve the quality of what they're trying to roll out. So, there's a convincing aspect to this, is the first thing that you have to do.

And then my recommendation is, and this is with everything that's a transformation, because I've been involved in all these, I hate to use the word transformation. I'm making air quotes here. Right? But when you have an organization that you're trying to get to do something really differently because the problems you're trying to solve now are different than the ones you were trying to solve before, is it is always better to start small and have a win. Right?

You do want to start a conversation at the top, where you're like, "We need to convince everybody," and it's going to kind of top-down, and everybody's bought in. Right? But the fact of the matter is, the best way to convince somebody is to succeed. Right? Because if you do something in a new way, and it works, you're much more likely for have people to go like, "Oh that's working. I want to do that thing."

Rather than, "Let me tell you about what will work." Right? "Well, I'm not so sure."

And, so, I guess my point is, pick something. You got to pick something that you can get a small team, engineer, designer, product manager, together. And you pick something that you can do, in a systematic way, to show the value of it. And it's hard for me to suggest what that is because different organizations are different. Right?

What I've found, oftentimes, is, if you want to just start with components and their visual styling, is actually not a bad way to start, because what you'll often find is that that's the thing that front engineers hate the most, is having to spend all the time getting all the X codes right and getting all the whatever right, whenever they're trying to implement someone's design.

And, so, if you can say, "Hey, for every drop-down menu, here's this thing, and it's got this nice visual styling on it."

And if you can get that rolled out to most of the drop-down menus in the software, it's very powerful. People, all of a sudden, are like, "Oh, that saved me three hours." I was like, "Do you know we just saved three hours for 20 engineers? Do the math."

Chris Strahl:

Yeah, exactly. That's one of my favorite things when we're doing implementations for Knapsack. Right? We oftentimes talk about, what's the most painful thing that's easiest to solve and trying to suss that out. And it sounds simple when you say it out loud. Right? But it's actually kind of a fairly deep conversation, to get to the point of understanding of what actually is a really painful thing, in terms of a

mundane task that's easy to automate away, that provides a huge amount of value across that organization?

And inevitably, we always find one. Right? And it could be the button, but most often, it's something like that, where it's like, "Oh, you know what? It's just a real pain to normalize line height across all of our applications," or something like that.

Todd Wilkens:

Right. Right. Or the other thing that's helpful is, the holy grail is something that you can see saves time internally, but also, you can point to a customer improvement, customer experience improvement. That's the holy grail.

And, so, what I often like to do, that's why I'm saying things like, you can pick visual styling, but pick visual styling on something that actually has a readability issue regularly throughout your product, or choose something where the interaction is a little janky that causes some issues sometimes.

Oftentimes, I'm like, that's the thing you're going to solve. You're going to solve it, so that everybody can just, all of a sudden, every drop-menu doesn't do that janky thing anymore. Right?

I find that errors and inline help are also one of the most painful things because they're almost always, people start off doing them with these just hard-coded whatevers. And then people need to update them, and it gets crazy.

Whereas if you can create a pattern and even a piece of code that can easily be attached to almost any kind of element in your system, that's attached to a database, so it has a systematic approach to pulling into actual text or whatever, that can be a relatively easy thing to do.

And you roll it out. And it's like, all of a sudden, all the engineers are like, "I don't have to hard code the crap anymore."

But all of a sudden, you have inline help for everything, and so your customers stop sending things. Your tickets drop by 50% sometimes because the thing that they always have to write an email about, is now just available right there, next to that form element, where someone doesn't understand what's going on all the time. Right?

I find that that's the right kind of thing to solve because it demonstrates the full value. Right? Some could argue, I always argue about efficiency, but they can't argue its customer value.

Chris Strahl:

So, now the other half. So, you have somebody in that same product management, product leadership chair, that is unconvinced that there's ROI to be found inside of a design system implementation.

I'm curious what you'd say to convince them, because my approach is, I look at this, and I'm like, you can't afford not to invest in systems. But I'm a little bit biased. I'm kind of curious, from your standpoint, how do you approach that kind of conversation?

Todd Wilkens:

It depends a little bit on what's the stage they're at and the nature of their company. Right? Because I'm going to just tell you, I don't believe that you have to systematize everything, and you don't have to do it at every stage of a company. And, so, I would argue that a design system can be really helpful for anybody at any stage.

But what I define as a design system for an early startup, is much smaller and less complex than the design system you have for 1000-person company that's six or seven years old, but they're both design systems.

So, it depends a little bit, but let me ... I'll put it this way, which is the biggest pain I run into, as a product leader, is that we know what we want to build. We know what the customer wants. And it seems really straightforward. But it turns out that we didn't know about this use case, and we were rushing to get something done, and so we hard coded some stuff. The data model is not, that we have, is not robust enough to handle this new use case, or we built it in such a way that if we add this extra thing to it, it tanks the performance and load time on something, or whatever. There's all these things that happen.

What you realize is, you walk in, you're like, "I can't do what I want, and it's going to take a long time to do it because we didn't ... We weren't thinking a little bit about scalability from the beginning." Right?

And then what you realize is, somebody's always like, "I got to refactor that," or "I got to ... " whatever, whatever phrase you want. But what that really is, it's that moment when you're like, "I really wish we'd had a systematic approach to this. I really wish we'd had systematic ... because now we're going to waste a lot of time on something," which, at the time, we thought it was faster to just plow through in all the individual cases, but it actually turns out this is something we do all the time, and it sucks because it slows us down there all the time.

That's the case I make for a design system, which is, that's very common understanding, you run into that in platforms and engineering all the time, surely engineering things. "We didn't make the data model, or we chose the wrong table structure in our database, or we ... " or whatever. "Chose the wrong technology to build this thing on."

What I think design systems really does is, it's this moment where, early on, you're like, "Well, if we make a product, and you ask us what's important, experience is the product. We really need to think about a scalable experience and a scalable approach to delivering a great experience, as soon as possible. And if we haven't been, we should start tomorrow."

Chris Strahl:

I love that you even brought in this idea of it's this structured, systematic way of avoiding technical debt and avoiding having to pay later for today's decisions. I think that that's a really brilliant way of thinking about it.

Todd Wilkens:

It's not actually that brilliant. It's a hard knocks. This is the most frustrating thing, as a product leader anywhere, is usually that something that someone did before, wasn't very strong.

Now, I will say, in all fairness, I gave you the example of, "We screwed this up and didn't think about it," but what's actually a more common thing is, "We did exactly the thing we should have done at that time. And now we're a bigger company. We want to do something more than what we were doing before, and so, we need a different approach. And now's the time to take a moment to standardize the things we've learned over the last year, so that we don't have to redo them or relearn them again." Right?

So, it's not, "I'm fixing the mistakes of the past." You often hit these inflection points when you realize where you're going, and you realize that there's a level of quality and flexibility that you need, and that certain problems, you should solve them once really well, and then be able to build on that solution, rather than assuming it's a peripheral problem. It actually turns out you realize it's a central problem.

And I think design systems are really good for that, because you start to realize, you're like, "These are the central user experience problems of our product, and we're going to solve them really well once, and then capitalize on that for the next year or two, at least." That's the design system fundamental. That's convincing to me, as a product person.

Chris Strahl:

So, when you're thinking about, again, that product manager, that product leader, and they ask you back, "What convinced you, what made it so that you believe in this, and do you believe in this? Is this really your idea of what the future holds?" What do you say to that?

Todd Wilkens:

Yeah. Well, I mean, part of it comes across because I love design systems. I'm a huge fanboy of design systems. And, so, sometimes that enthusiasm, sometimes it's infectious, and sometimes people look at you, like "Are you some sort of zealot or something?" I'm like-

Chris Strahl:

Are you okay?

Todd Wilkens:

... "I am. I am. I am." But I am because I have just seen them work. The biggest problems I have had to solve, as a product leader, and even as a design leader, design systems are always at the center of those solutions. They always make it better.

That's the thing I would tell if somebody is like, "I could try to convince you with logic. I will do that. But I also just want you to understand how foundational this is for me, just as a person, based on my experience. Hopefully, that helps you understand something too."

Chris Strahl:

Todd Wilkens and the Cult of Design Systems. I love it. Thank you so much for being on the program. It's been really great to get your insights, understand where you came from, and what's led you to being a true believer.

Just so appreciate the time and looking forward to hearing back from you again sometime soon.

Todd Wilkens:

Yeah, thanks, Chris. It was really fun.

Chris Strahl:

Take care everybody. Have a great day.

That's all for today. This has been another episode of 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 with 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.