



IxDA Sydney Podcast

S02 E9 - Amy Howard Howard

Audio Transcript

Amy Howard: [00:00:00] I can't go past a button. And I say that , probably such a dork, You can do so much with them and they need to just, ah, they're foundational

Sam Hancock: Hello, and welcome to the I X D A Sydney Podcast, a show where we can't guarantee answers. Just better questions. I'm Sam Hancock. And in this episode, Vinita and Jess are chatting with Amy Howard. Amy is the Lead Designer for IAG's Design System, leading the creation and implementation of tools and resources that support designers using the design system. She works closely with designers and digital teams across the company to support its adoption and ensure it meets the needs of the organization and its users.

Let's get started diving into the world of design systems.

Vinita Israni: Hey Amy. Thanks so much for coming on to the podcast today. We're so excited to have you.

Amy Howard: Yeah, thanks so much for having me. I'm really excited to be here.

Vinita Israni: So one of the first things we love to start with is just asking you about your background and what has brought you to this moment today.

Amy Howard: Yeah, great. Well I actually started out in graphic design, but I wasn't that great at it. Like my drawing skills were really terrible. I did not get that gene in the family, although I tried really, really hard. So I kind of moved into designing for the web which back in the day was called multimedia.

You kind of had to, design and code everything. There weren't different roles I had to learn , hey, Gmail and c s s and all the basics. But I was really glad cuz I do feel that helped me be a, the designer I am today. It helped me understand more of the restrictions and impacts for code because that that's what, we do we design for code.

So once I kind of got a bit more into to that design side, it was a very long career going, oh, do I go code? Do I go design? Do I go code? But I [00:01:00] started to specialize more in UX and ui then, so went down that path and slowly moved into more product design. I wanted to start iterating and growing products more, just not looking at smaller projects.

I wanted something bigger I could sink my teeth into. Look at the whole life cycle of it. Look at the whole service delivery of it. And , I was lucky enough to find my way into design systems. I think I got given a go when I'd really knew what design system was. So I was really grateful for that and ended up just absolutely loving it and which is where I am today. .



Vinita Israni: Awesome. And I know that you are currently very interested in design systems. Curious kind of what that looks like for your role today and what specifically got you there?

Amy Howard: So today I work really, really closely with the team, that works on design system. So we've got design system specialists product owners, developers, and content designers.

So my day to day is really focusing on the product with them. And I, look very closely at the designing and documenting of the design system from the design kind of [00:02:00] perspective. But what I love that it's not just ever a design perspective, it is a product perspective. So not only do we work very closely with the designers for the companies that, I work on the design systems for but the others in the team.

So being able to bring that all together, from a content and a development perspective and being able to , iterate and improve that product based on the feedback. It's a constant ever growing challenge to keep something relevant and effective. I love that challenge. I think that's what draws me most to it, that it's something that never, I don't think it ever has an full end .

Always being able to go we got to a great point, but , where to next? It's always got a where to next. And that means I'm daily talking to designers and digital teams. I spend heaps of time in Figma. And , every day I'm probably staring at a new component or a style contemplating the meaning of its existence.

Which sounds really sad, but for me it's really exciting. , staring every new day at a form field or just a color. And , going deep into its [00:03:00] world.

Vinita Israni: And so maybe taking a step back there for those listeners that are maybe not as familiar with design system, what do you qualify as a design system?

Amy Howard: Yeah, so a design system is, it's so many things and they come in so many different shapes and sizes really, but it's giving that consistent and accessible base for everyone to work with and to have that same kind of foundations.

It is the building blocks for design and development, and also gives you a . Design language even to be able to build more from it. So it might be a smaller base, it might be just your core stuff. It might go into larger patterns. It might go into flows. It really needs to adapt to what its users need, but it is essentially all those foundational building blocks to effectively and efficiently design digital products.

Vinita Israni: And so what kind of forms do design systems take? I know [00:04:00] some companies use websites that are public that can be used for commercial consumption versus some things are quite external. Some design systems may only live in a Figma file ever. Any thoughts on that?

Amy Howard: Yeah, design system really is all kind of those pieces you just mentioned put together. So you have a design side which can live in a design file, whether a software whether that's Sketch or Figma.



Figma is the, larger one today. But you also have then a code side. So then your code and your design match.

So those foundations are there, those components are there, all those building blocks. Match from that design side to that code side. But also you can't have a design system without documentation. And that's where it comes together is it documents, what those pieces are, how someone uses those pieces, and then how to go further as well.

,design system will never be able to probably cover everything a designer or developer does every day. But what it does is with those building [00:05:00] blocks, say here are all the styles you need. Here are all the, that base visual language you need here is all the small pieces where we've taken those foundations, put it with that base visual language and started to create pieces for you.

So you can take that equation basically and go, yep, this is how I put all those together to then build larger components and patterns . So that documentation plays such a large role on going. This is everything. This is what they mean. This is the meaning behind whether it's hierarchy or some importance .

This is how we communicate this type of interaction or this type of meaning through something. Now, if you need to build more, Take that, and this is how we then put that stuff together and you can then go and scale your designs. You can build anything from a small little login page to a whole massive onboarding flow .

So really, for me it's those three pieces coming together. So the design, the development, and that documentation of how to implement.

Jessica Pang: It sounds like being able to put together that[00:06:00] base and, and shared documentation is building a shared language within the team, within the organization

I'm curious to learn a little bit more about how, the different ways that designers and developers might use this documentation, but I think it'll be helpful to kind of think. About as we make that transition from design to development, how might a designer consume and contribute versus a developer?

Consume the design system? I think understanding our user base, right? Like how do you feel about the different user types?

Amy Howard: So they definitely are different. Because what they're doing with the pieces of that time is it can be quite different. So where you've got a design system for a designer what it's giving is, All those smaller pieces, all those foundations, and then the application theory, all the usage behind them.

So especially when you come to your foundations when you've got say, color, it's about that color application. What does it mean to apply these different colors? Where should they be used? Even the

components themselves, it's what should this be used for [00:07:00] over another, say a date input set or a date format field.

And the design system's there to help create a consistent use of all of those pieces and help educate designers as to how the design system has built these to be used. So when it comes to, contributing back for designers look, I've worked with the kind of many different type of contribution kind of models and it's what works best for you.

And really, we were looking at What is the need you guys have? Is there a, suggestion? Have you actually designed something and then we look at that piece as well and go, okay, well for a design system, things have to be quite context agnostic and they have to be really scalable.

Which means that they have to be able to grow and they have to be able to use our foundations to be built upon. As well as someone can take that and just place it in any instance. It's not tied to one flow, it's not tied to one meaning. So quite reusable in that aspect.

And that's great cuz that's where you do learn a lot more about what designers are building even and, and what their needs of the design system is. And how your design system base is gonna start build [00:08:00] up to that. Because really, as soon as a few teams are building something you don't want them building it again from scratch.

You're like, no, this is gonna be reused. It should be in, yeah, the design system. So, We have one consistent application of it. And, working with designers to then see how they actually work in their teams. So their ways of working, how then they work with the developers as well.

Cuz the design system has a lot of power in that, handover as well. We see teams that work really closely with developers. We see teams that actually work really separately as well. And what can the designs do to help with that communication? Because really, Design software is our communication tool.

Designers are communicating what needs to be built or what needs to happen, what needs to, to look and flow so where we take the opportunity to. Kind of pack as much information into that design software that we can and make sure that what pieces we're giving designers match what we're giving depths.

They can have the same properties being changed. It does the same as much as it [00:09:00] can, in design software. It acts the same as that code piece too. And then what developers kind of need out of that is they need to be communicated to what part of the design systems they're pulling together and, and building with.

So This is where we kind of love to see that common language being built too. So designers and developers are referring to tokens the same. They're referring to components the same. They're referring to properties the same, because we want them to be able to say to a developer like It does all this.

And a developer doesn't have to translate that into what they have available. It's Nope. That is a one for one kind of thing. And, that does, it does get a bit hard because there are already design and development languages. So kind of bridging that gap, it's really interesting and , can be very different in the different ways that you approach it.

So then it's communicating to the developers what they need. And in a way, sometimes they will also be building their own components too. It might not be a, a design component, but it might be a dev component. So what does that look like in itself? It [00:10:00] can be , something that they're putting together that they've reused a few times as well.

It can be kind of even just a way to code. It doesn't necessarily need to translate to a UI piece to be a contribution back into the design system because it can be there for accessibility as well. Just a way to better code for, for these instances that we can apply constantly. It's really when you need to look at the company, your design system.

Is in and what those individuals need from it. You can take on many shapes and sizes to what you kind of provide, how you build, and then how you look at what can be contributed back and how you kind of define what does and doesn't live in a design system.

Jessica Pang: I remember actually when I first started in product design as well I think understanding how to work with developers contributing and also using the different design components that's fit for purpose. I think it requires a little bit of technical understanding as well.

I'd love to hear a little bit more, and, and you mentioned this There's that kind of one-to-one mapping between a [00:11:00] design components, properties versus translating that into code and, making sure that the tokens match. I think you'll be helpful to understand a little bit more the technicalities of how that implementation of going from pixel to code and being in the design system.

Amy Howard: We in the design team have to. Really understand ourselves, what design system, the level , of how close it can be to code, how close can we get it? Because it will never, I identically match. You'll never be able to prototype in, in design what you can encode.

So it is also kind of helping to. To educate on what things you can test wet as well. if you wanna test more accessibility, sometimes it does have to be in, the tech, because that does rely heavily on how things are built and how screen readers . But also then what needs to be kind of done in the design side as well.

So when it comes to , that translation, that's where we kind of work really closely together. And we have to look at the logic behind how both are applied. So a lot of things in a design [00:12:00] system, kind of will go through like, this is what designers might need to put together.

And then how do we go on and build that? But also, it can go the other way is , what pieces do developers need? Like we have pieces that designers probably wouldn't translate back into, designs. It's done a



different way in Figma as well. Especially saying auto layouts and stuff, whereas, Developers don't have that.

So how can we create those two kind of ways of working across both that doesn't have a direct connection as well. So that's where we kind of look at how both of those kits need to then live with each other and how they can talk to each other as well. And that's why documentation comes in massively too because we can't always do that communication.

So it's then What does this kind of application mean versus what this kind of application means?

Vinita Israni: Yeah, I think that's great. And you started touching on it just a bit, but I wanted to push on it was around accessibility in design systems. You mentioned the screen readers, and we've talked a little bit about accessibility on the podcast [00:13:00] before, but how do you start to build accessibility within design systems?

Amy Howard: Yeah. Accessibility is a great thing, to touch on because the design system itself is about creating pieces that help people build experiences. So the more you can build in at the design system level, the more you're actually helping teams to Focus on larger challenges and larger parts of accessibility.

So what we aim to do is really bake in accessibility from the beginning. And the obvious places are in the components and your patterns. And from a design side, you're very much looking at that color contrast. And the way things are kind of working there and that interaction as well.

An accessibility point of view. But then we also wanna look at then when you take those pieces across, how these are also functioning in codes. So looking at accessibility of it in code as well. Not many times can we go too big because we are only providing the pieces, but we're in a great position in design systems to help educate too.

So that documentation really plays a large part in assisting teams to be able to test further what they can be doing, what [00:14:00] resources you can be using. If you're using parts of the design system, are there extra additional things that we've got to help you be able to implement that as well?

But also one of the biggest things that I've learned since working in design systems is actually just even injecting accessibility into how pieces are used. So how your color, logic or color framework is used. You can provide a whole bunch of colors but if you're not providing how to apply them accessibly, then , when it comes to things being switched out, and you'll probably rarely find a design system.

That just has one application of color, whether that is reaming or re-skinning or whether it's looking at a different kind of theme for different platforms. color will switch out, color will be needed in different ways. And providing a way that means that if one lot of color might not be accessible with a second way that it's applied, you are giving designers the tools or that kind of framework to say.

This is how to design accessibly for that switch [00:15:00] or for that movement into a new kind of theme. If going into even Where people have limited color usage as well building from the ground up, things like that. So this is where that part becomes really important, not just building a year into those small components and patterns, but, how you design and then how you think about your color application throughout that design is really important.

Documenting that can be quite a, a mouthful sometimes. I think I've had to document it a few times and , once you can see it come into life, , that's what makes sense.

Vinita Israni: Can I ask a question through that? With, especially with documentation, some design systems start with. Mapping their edge cases for the design systems as case studies, and some of them say the opposite. Right. So that's design for the majority. Any thoughts on that? Especially as you mentioned, designing design systems for different kinds of organizations?

Amy Howard: Yeah, look,, I probably don't think there's one exact way or the other [00:16:00] because it really is an assessment on what , your company. Needs, and you have to consider those edge cases. And another thing you have to consider is what is the remit of the design system? So what are the limits that it can do or it can provide for that company and considering, the edge cases is great because you can Includes so many other things when you consider those edge cases, things you haven't even considered will be captured.

If you make something as inclusive as possible but then you might not always be able to, to capture everything and you shouldn't ever not do something because it might not benefit everyone. That's what I kind of think, like you might get 90% of the way there. And all this this fix covers 80 or 90% of people, or this implementation does it shouldn't stop you from getting it in there, starting, iterating, building, and growing on.

Does it need something additional? Does it need a bit of a tweak to what we've currently got? So [00:17:00] I think you might have to start with everything in mind, but you might not be able to solve all that at once. So design systems are constantly growing, so you might , have to take those small steps to get there. But having them in your mind is, , I think it's quite critical

Jessica Pang: now with a lot of the documentation and, , and being it an iterative process, I am curious to know how do you go about validating , and testing some of the design patterns and design components as you build it out.

Amy Howard: With design systems, you're actually really lucky that you have direct access to your users most of the time.

They're the designers and developers that work with you which is great so when we're getting those needs and getting those requirements You do have to look at how you facilitate getting those and how you collaborate with your teams to get them. And potentially that can be one of the hardest.



So what is the easiest way for you to enable your users to give you that feedback or to give you their needs? And because it's within the same company there's so many different channels or ways that you can do that. Making [00:18:00] collaboration spaces, making community spaces , just even communication channels.

And we've been really lucky enough that , you don't always have to hold too much formal research because you're talking day to day with your users and you are hearing how they're working and you are kind of putting those, little dots together. And then when it comes to going, okay what more do we need, you can then dive into a lot more of that validation and testing and iterating with them.

But getting more people in the room actually doing some prototyping in that. They can tell you directly what they need. And if you're starting to hear that kind of all across the place, you can start to work that into a really kind of reusable thing for a lot of teams.

Because , that's the goal. You're not kind of trying to fill one gap with the design system. It's not like this is my one issue and we need to plug that. You wanna see what the trends are. You wanna see what the base is that your designers are needing, what more can you give them? And it might not necessarily be even in the shape and form [00:19:00] that people think.

They kind of need, it might be, what? You actually just need more granular pieces to be able to fill your need and a way to put those granular pieces together. Or it could be straight no, you guys just need a tool tip. We just need to create you a component, , straight up, we know that's gonna work for everybody.

So it really is, figuring out exactly what that need is from the users. And then seeing how the design system can help to fill that need or that gap. But I think it's great that the, users of a design system are so close to you, there's no reason you should not be talking and collaborating and reaching out and testing with them.

, they're all right there. They're so close. And I'm sure they'll have no problems telling you exactly what they need. You are there to help support them. That's what a design system does. It's supporting its users. It's helping them to work better and helping them to work more efficiently.

Jessica Pang: I think we've touched a lot on process and how design system is, is baked into the product development process and improving , our product , going from pixel to [00:20:00] code. And I'm curious to know, maybe this is a good segue I saw and how does design ops, for example, play a role in maintaining the design systems?

Amy Howard: Design ops plays a massive role. I think this is what gets me probably the most passionate about design systems, cuz it really does kind of standardize and optimize that design process. And what I wanna do in design systems is really make the lives of designers so much easier and help them to focus on much larger.



Design problems or UX problems . Giving them all the pieces they need means that they can stop the repetition that they would usually have to do by building so many things from scratch. They can focus on those larger bits because we've given them all the smaller bits. It also helps to bridge the gap between designers and development.

Probably touched a bit on this before, but using whatever design software or system that you have to help designers communicate to developers what they're building. So this is a gap that it's constantly just trying to be bridged , or small [00:21:00] and I don't really know if that's a word.

But we're trying to see what we can a design system has so much power in that space. We can pack so much more information into design systems, , to help designers not have to write all these things from scratch, not have to annotate everything, and put notes everywhere where we can include that in our actual pieces.

So something small in, Figma you can actually be. Turning on and off variance in properties like code does. So making sure that terminology is correct, making sure that works, how it should. So really it's us optimizing design software to be a communication tool that I think really helps in that design op space.

It means that , those bigger things can be focused on, the teams can work. More smoothly together as well. Cuz that power to help designers communicate I, whether it's tokens, whether it's , the components or patterns. It just also gets people feeling that common language, like we said before.

So that even helped design to just work more [00:22:00] smoothly with development and make things faster and more efficient and be able to build kind of more solid digital products.

Vinita Israni: It almost sounds like there's a sense of solidarity when you have a design system in place, both from a ways of working perspective, but also from a org structure and, more formal processes. So building on that how do you start advocating for a design system for a organization that might be quite small and on the cusp of unsure if they should have a design system?

Amy Howard: Yeah. So you're right. It does create such a little community. I must know, one thing I do love about working in design systems is that I get to work so close with everyone in my team. So not only does it fuel that community collaboration in our team, but we hope that it makes teams outside of us helps them as well , to work more collaborative together,

so if you are fatigue starting out, To try and, advocate to [00:23:00] get a design system. I would say focus on the parts that people really do understand. And that's that it saves time and money and stops repetition. So it actually is helping people in the team to speed up their work, but also Saves the company as well.



So you are really killing two birds with one stone in that respect. And also you don't have to start big you can start quite small you can use other open source design systems as well. So again, even in building the design system, you don't have to start from scratch.

And I think that's the main thing design systems tell us or are trying to educate us on, is you never have to just go and rebuild that button again. Someone else has done it, even for a design system. So you have a base you can start with there. It doesn't require a big team. You can start small, you can iterate, you can gradually expand and you can build that momentum and demonstrate its effectiveness over time.

You don't have to go in and be at a hundred percent meeting every need from that split [00:24:00] second. And, the great thing is that there are so many design systems out there to learn from too. So you're not going in blind to go, okay, these are the needs of my designers and my developers.

you can learn from how other design systems have implemented it, and there's been a whole lot that have had really unique or edge case kind of moments where they've jumped in and explored how to really focus on their needs. But that doesn't mean that need can't also be adapted to go, oh wow, that's actually what we need to, , that's the same circumstances we're in.

everyone is kind of, not everyone, sorry. A lot of companies are going this is how we fixed our problem, or This is why we had a problem with our design system. This is why our design system wasn't working. This is what we thought we were trying to fix. And. They're out there now or talking in whether it's presentations in articles, in seminars about this is how we went about approaching that problem. And so it might not be [00:25:00] the same solution that you need, but that systems thinking and that problems thinking to get there and to understand how your design system might need to fit in. To your organization and how it might need to support all those stories are out there.

They are there to help kind of support you and take you in the right direction to start iterating and to start just from those small basics as well. I think that overall value of just what it will do for a company for your designers, for your developers, Bring everyone together, like you said, bring that whole way of working so much more together is something great to focus on when you're looking to advocate for a design system.

Jessica Pang: Yeah, definitely. Just as you mentioned advocating for a design system means building that shared language and having good strong foundational blocks and making it accessible and inclusive.

I am curious to know though in the world of design components, which one do you think is the sexiest component?

Amy Howard: The sexiest component? Oh, [00:26:00] interesting. Ooh.

Jessica Pang: Or your favorite component.

Amy Howard: Yeah. I'm trying to think. , cuz sadly, I probably would have one. I had to stare at them all day and I'm probably like, you are my fave.

I can't go past a button. And I say that , probably such a dork, but buttons are , You can do so much with them and they need to just, ah, they're foundational and they need to not only look good. Like look good is probably the bottom thing. But they need to be so recognizable, so easy to identify, have the right interaction so they carry so much with them.

And the amount of times I've heard people go, oh, but that's not the trend on how you do buttons. And I'm , don't, we're not following trends. We're making the best button. You'll ever need. In my head, that's what I'm always thinking. This is the one component that everybody will need at one point in their lives.

Every designer, and you can do the smallest [00:27:00] things to give it personality even it's Border Radius gives it just a little bit of personality. The text inside it, the padding, it's, yeah. I'm starting to sound a bit weird now talking this way about buttons, but it's probably that I'd go buttons.

Jessica Pang: No, I love this. I am also very particular about border radiuses. I usually just max out to 200 picks because I like the bullet style. Yeah. And I find myself commenting on real life things in Border Radius. I was like, oh, this bus is border radius must be like 20 picks, like in London. And I'm like, oh my God, I'm so geeky right now. So I totally understand. This is great.

Amy Howard: Someone gets me. I love that.

Jessica Pang: No, , this is what this whole conversation's for. We'd love to deep dive into topics and geek out on design. So this is great. I think we kind of talked a little bit about the styles of Designing different components, and I'd love to hear a little bit more around your experience of branding design systems as well

Some design systems and component libraries , like to name their [00:28:00] components related to food or have some sort of branding around it. I'm curious to hear your experience with that.

Amy Howard: Design systems can take, different approaches when it comes to how they brand.

So you've got a lot that just kind of are very standard with their branding, but then a lot that get a little bit funkier, like you said, give them fun names. And one thing about branding is also understanding what level of branding your design system needs to take on and what actually does change in that design system.

For example, the the Westpac design system has multiple brands underneath it. But they are rebranding components. So those components themselves just have a new theme placed on for those , the brand colors and kind of how that is Portrayed within their designs and how it gives that personality as well.

When I worked on a New South Wales design system they have re-enable things as well. But theirs is, different. It's not just one application. You can have a mixture. You can really dive deep into how you are. Branding something. So how you actually are themeing is [00:29:00] completely different.

So you might still be doing a theme, but how you apply it, how it's chosen, what level you can go to. Companies also have different ways or different hierarchies of theming, what can be done to what can't be done, keeping close to the brand, going really kind of to its own personality as well, but, Does it just sit on that color layer?

Does it go deeper? Does it go into things like border radius, typography? Does it go into spacing as well? So this is where you really have to look at your design system and go, what makes that personality for that brand? And is it even all going to just work if you flip the switch? So if you flip a switch and just change your typography and your color and your border radius, is that enough?

Does it actually need to have completely different pieces in it? Maybe brands within your design system might need to look completely different because one's a shampoo brand and one's a food brand. [00:30:00] They might need to look completely different in that respect.

You might be able to ream, but that theme is got a lot more complexities to it. And it might not be , a direct one-to-one mapping because they need to exist slightly differently. But the, the logic and all that is still there. The framework is still there, how it's applied and what pieces you would use is all still there, but it's visual presence and , that story that those pieces tell about the brand can look very, very different.

You might not be able to tell that they're the same design system. Or you might be really obvious to , a more trained di that goes, oh yeah I can see all that. But to a normal person, everything has to look completely different. But how it's built, probably there has the same framework how it's laid out, how, you're solving UX problems, that's all there as well.

Cuz design system isn't just ui. Design system is that whole way of building and whole way of designing. So whether a user can see it or not, there might be a whole strong design framework underneath that, that teams are [00:31:00] using that. Yeah, the user might never even be able to guess. So yeah, themes are hard.

Cause again start from like a small as just flipping a few colors to. , flipping a whole system kind of thing, or a whole kind of UI and stuff. They're fun to work with. And actually I find that really fun because every company has a different way they define that.

Like I said before, it can even be via platform. So you might just even have a different look and feel for your apps versus your desktop for just a little product within something might start or tweaking up and changing. But what does that need to do and what does that need to achieve versus your others?

So that's where I , find it really exciting to meet those needs. And then when you bring it back, how designers and developers can work with your theming frameworks. That's one of the biggest challenges, like you need to help them to implement that. You need to actually help them to take on all that, apply it, flip it, frame it.

Jessica Pang: I can see how a design system is something that [00:32:00] continues to evolve and , it has to go hand in hand with things like branding that you've talked about , the identity of a company, perhaps even how they write copy.

And I can see how it all needs to stitch up together. But at the same time, there's also, branding for internal users. So I've, I've seen some fun things actually. Sometimes I like to look around some of the open source design systems just to see what's up. And I've seen some really fun things, for example, , Trello's was called Nachos.

I think that was their design system name. , and so that branding internal users is also something that's quite fun as , just to kind of bring it a little bit more to life as well.

Amy Howard: Yeah, it's great to give it a personality. You're right, the design system itself. Because when you refer to it, you don't wanna keep saying, yeah, the design system, you want it to be , a living, growing organism.

So giving it a name, it has to have its own tone of voice. So it's personality, what do your users need from you? How do we talk? To our users. , what do we provide? How is that done? , that's all in the personality of your design system, which is, a really fun part of creating it is trying to name a [00:33:00] design system. I unfortunately have not been involved on any namings. But one day I hope that I can point to one and go, I name that. Yeah, that was me. That's my great idea.

Jessica Pang: I have to say, a random thought came to my head the other day and I realized there's a few components just named after food, like meatball and, the kebab venue and things like that.

I just thought it was a random fun fact. It makes sense. naming after food is probably a good and safe option.

Amy Howard: , I agree. Cuz when it comes to naming components, you've got navigation but. The amount of different styles and types of navigation you can have.

How do you communicate to someone exactly what this does differently to another? And , using something like the kebab one, you're like, oh, cool. In my head, I know what that kebab looks like, so this should function like that. But compared to a different piece, it's a meatball like that, you're like, oh I get that.

That is. One of the hardest things I've found is trying to differentiate different types of components. And I love that food analogy. I haven't seen it, but I'm totally [00:34:00] gonna , explore that now. See if that helps.

Vinita Israni: So Amy Howard, I know we've talked a lot about design systems themselves and how we create design systems, and we have a lot of listeners on the podcast that our designers, so curious about how design systems play a role in a designer's career. This is so meta.

Amy Howard: no, I love it. It's great. What I think a design system does in a designers career is, really helps them to cement a core visual language and something they can be guided by and designed by. So in a way what I actually realized was so many designers are probably creating design systems.

Without knowing it before you actually experience , a massive one yourself, you're probably gathering so many different pieces that , the developers have built that are pre-built or that you design with regularly and you're collecting them on the side. But that that's, one part of the design system.

So what a design system's actually doing is going to you. Okay? You've got those [00:35:00] brilliant. Here's how to apply them. Here's how we're gonna translate them over for you. So, I feel designers are already taking a step into that world of a design system. But a design system is taking it one step further for them and really supporting them a lot more with it, and then also giving them basically a formulaic way to design The way I kind of look at how a design system is, you are a designer, is building an equation a lot of the time for a developer.

If something isn't in the design system that's in that equation, then things need to be a bit reconsidered. , what is the design system missing that that designer needs? But putting together the styles whether that's color and water, radius and topography. Along with that visual language of how we communicate hierarchy, how we communicate navigation, how we communicate, all these little bits.

And going to a developer, this is what I built now because you can put all those pieces together. So it is doing a massive amount of the work that. Like that underneath work, that foundational work for them so that they can go Yep. And just start grabbing all those pieces [00:36:00] together and really focusing on those bigger things.

And also is there to help them to aid the transition from design to development. So A lot of designs might have to annotate stuff or write stuff all down on their designs. But what a design system is doing is taking even that away from the need.

you still need to have a really strong design to development kind of transition. Increasing that communication tool already being baked in is helping them do that. It's helping them be able to communicate and transition a lot easier. So really for a designer, what we are trying to do is just alleviate so much of that day-to-day stuff.

The mundane stuff, the taking away the hard bits so that a designer can just kind of run with what they need to run with work in the teams. They have the best way possible.

Vinita Israni: Yeah. One of the things I recommend for a lot of designers starting out is to actually work on design teams. Cuz I think you learn so much their completely different way [00:37:00] of designing your, you mentioned, users, audience is different and you're working on the.

How in the process rather than necessarily the end product, although you are in a completely different way. So thinking about that, what are some realizations that you've had as you've worked in this space over the last couple years?

Amy Howard: You're definitely right. One of the biggest things I've realized is that you have to leave your creativity at the door because you are building.

Not the end flow or pattern or anything. You are building all those smaller pieces. So that is where it needs to be. That language, it needs to have that language built in. , you're not building for trends. You're not building to make something look pretty anymore. You are building that tiny little piece that someone's going to put into something that's going to flow beautifully and have a wonderful experience in.

So you really have to go, what? I need my 80 year old granddad to realize this is a button. Or this is a form field or I can't get [00:38:00] too Creative and wild because I need this to be a consistent application that works for everybody. And I don't think it's a bad thing either because someone needs to do it, some team needs to do it they need to do that small, granular, little bit of pieces so that other people can not have to, and they can work with the same consistent base, which means that all your products, all your services, all your digital platforms are gonna be.

So easily usable. The experiences are gonna be smooth, so no one has to guess how to use your product. That's kind of what you are doing in this space. You are taking away an end user's cognitive load, but. Two users back kind of thing. So you might be building for your direct users, design and developers, but you want that end users experience to be the best possible way, and you want your users to be able to design then for it.

That's probably one of my biggest things is staring at a input field day after day. How do I make this more functional, more recognizable, and its [00:39:00] interactions more prominent? It's not, how can I make this look better? It's none of that. Sometimes the ugliest things work the best because they're just the most obvious and no user has to guess. So that's my biggest realization.

Vinita Israni: I love that and I think it's really important that. We define some of those principles that we have, right? So functional as something that you're aiming towards. Okay, so what does that look like? Right? So it doesn't have to be visual, visually stunning, but how do we make it as functional as possible?



So you've mentioned kind of scalable, reusable, flexible how do organizations or designers get to their definitions of each of those things?

Amy Howard: It's very unique, . What your design system needs to do in those spaces for each applications. What does your design and, and dev teams need?

What's perfect for one company might not be perfect for another, so you can model your design systems off others. , you can start with that, but it might not serve that [00:40:00] need. So when you're talking about like scalable, you want the design system to be able to grow with your designers.

So not only does the design system grow as a product, but can the designer scale their own design? Can they use the pieces to make something from small to big? But it's whatever their definition of that is as well. So you have to take into that consideration what are they doing with the design system that will take them there and , being reusable.

What are the context they're using it in so that you can make sure you are pairing something back enough that it is context agnostic and it is adaptable to those different user needs and user cases. But again, it's what are they using that for? Where are they applying it?

Why would they need to be? Is it being applied between different brands? Is it just being applied between different themes is just being applied between different platforms. What does context agnostic mean for you, for your teams and with flexible it's what is adaptable to the user and the [00:41:00] content need.

Do you actually need a fixed versus a flexible system , Do your designers need? This is exactly the way to do it. These are all the pieces you have run with it. that what, who is putting those designs together need, or do they need something that is, we could have bespoke stuff, so we might need to be able to adapt or have a flexible component that things can be added to, moved around and it's that core language and those styles that we maintain.

So it's very much understanding the needs of , the teams. To what that flexibility might even be. Whether it's also atomic or it lives at the atomic level, whether they just need those little pieces to be able to really create those custom or bespoke things.

Or you actually do go up to the level of UX patterns and flows, whether they need to be included as well. Whether it is actually just what the design is building. We make that into the design system cuz that is what helps their teams build more effective and efficiently.

It's understanding a lot [00:42:00] of what they need to, how you actually have to present pieces in your design system or to build to those pieces as well.

Vinita Israni: Amazing. Thank you so much for coming on and chatting to us about design systems. I wanna end with one final question, which is, what are some resources that you hang your hat on?

Sam is a hat wear. That's why, he has this question in here, but resources to further expand our knowledge, right? Even around things like misconceptions about design systems we can learn about or. How can we create more mature design systems and take our organization along for that ride?

Amy Howard: So I love listening to how other companies have overcome things or how they've built to kind of meet their needs. So what I do love listening to is Figma Configs cuz they get a lot of current design systems in to speak about how they've kind of especially looked at the design side of things. And how they've kind of adapted their design systems to be [00:43:00] able to actually help designers.

But there's also a lot of different kind of organizations. There's design ops, assembly and into design systems that actually have slack channels you can join that have events that kind of hold as well as. Much similar to that, like recordings or past talks with design systems about overcoming those challenges and the best way they've met the needs for their own companies.

And that's where I learn the most because the landscape is constantly changing. Which means that new kind of problems are arising or hurdles to get over or new things that need to be considered are always arising. Talking to teams and companies that are tackling those latest issues.

Is really kind of inspiring and it's really good to know that, one, you're not alone in your in things you're tackling, but two, that there's different ways you can think about things and someone's out there kind of tackling it with you one great source to look into is actually is [00:44:00] PayPal?

How they used different softwares to help with the small amount of designers that they had a massive amount of developers listening to stories that can show you how to really optimize what you are doing in your own workplace and learn from people who have tackled something like that before.

So that's kind of really inspiring. Listening to those type of things is really that design systems thinking and that design ops thinking of, yeah, how can I then, Learn from that, apply it to my own circumstances.

Vinita Israni: Amazing. Any final thoughts as we wrap up? Any design systems that you would recommend having a look at that are quite unique?

Amy Howard: I never go just to one design system. I think that's my thing. When I'm kind of looking at the way design systems work or how one is being built, or what's problems, it's solved I probably have my standard go-tos, which are UK gov.

Because it is just so functional. It is [00:45:00] there to meet the needs, inclusivity wise, accessibility wise, , and that's what we should be making the crux of all our design systems. So that is a, big go-to one for me and I think how they've done it and even how they've written the documentation, it is just so clear and so easy.



And. You don't wanna get lost in stuff when you're in the design system. Documentation. You want just those pieces and you want the reasons why all those pieces come together. So that's probably a big one for me. We all look at the big company ones , comparison on how they take on maybe different challenges in that.

But look, I probably don't have one go-to. I have my small pot of a few that , I'll tap into I mentioned Westpac before. That has been a great inspiration for house to document as well as looking at the theming side. So it's quite probably unique in that

opens source and you can look out and go, , I can see it changing before my eyes, which is really exciting.

Vinita Israni: Awesome. Well, thank you so much for sharing all [00:46:00] your wonderful thoughts with us today. It's been such a journey, I've learned so much. So thank you so much, Amy.

Amy Howard: Yeah, thank you so much for having me.

Sam Hancock: And that concludes the latest episode of the Sydney I X D A podcast for the transcript and resources, along with more information on I X D A.

Sydney, please visit. I X D A [00:37:00] sydney.org. See you next time.

Amy Howard: Hey, this is Amy Howard Howard and you've been listening to the I X D A Sydney Podcast.