Episode 49: Brendan Hopper
Episode Transcript

Mik Kersten:

Hello and welcome to the Mik + One Podcast where I sit down with industry leaders to discuss the project to product movement. I'm Mik Kersten, Chief Technology officer of Planview and best selling author of project product, the Survive and Thrive in the age of digital disruption with a flow framework. Joining me on today's episode is Brendan Hopper. Brendan is the Commonwealth Bank Group CIO of technology responsible for ensuring the bank has a world leading IT engineering capability and is making the right technology investments and decisions so that technology becomes an accelerator, not a disruptor for them and their customers. Prior to this role, Brendan was the general manager of CBAs Cybersecurity Center, working around the clock to actively defend customers and vital bank infrastructure from cyber threats. And as a co-founder of sce.edu, CBAs industry partnership with UNSW, another university cyber education in Australia since 2015. I've been collaborating with Brendan for two years now and I'm incredibly impressed at how effective he's at driving transformation results and flow at scale. So with that, let's get started.

Hello everyone and welcome to the Project to Product podcast. I'm thrilled to be joined by Brendan Hopper, CIO of technology at Commonwealth Bank of Australia, which is actually Australia's largest bank. So before we hear more about CBA from Brendan, I'm just thrilled to have him on the podcast because we've been working and collaborating very closely for the last two years and I've learned a ton from him in terms of what it's like to help drive innovation and flow at an organization of his scale. So with that, Brendan, welcome.

Brendan Hopper:

Thanks, Mik. It's great to be here.

Mik Kersten:

And if we could just get started listening to you talk about the early days of your career, it's just such an interesting story to me about how you came from InfoSec, how meticulously crafted and organized your entire career progression was, or not as we may find out, but really how you end up here. But tell us about the early days. Tell us about the start. Tell us about the kind of perspective that drove you to want to learn more about how to scale these kinds of things really from your early days, I think in InfoSec, unless you plan to go back earlier than that, so ...

Brendan Hopper:

Yeah, no I'll go all the way back but don't worry, I'll be quick, I'll be quite quick. I got into computing and programming, I was very lucky, I was gifted a computer when I was about five or six in the late '80s, and there was nothing to do on a computer

Mik Kersten:

Wait, which computer

Brendan Hopper:

Except

Mik Kersten:

... Was it?

Brendan Hopper:

What was it? It a ran CPMOS. It was an Australian specific computer called a Microbe.

Mik Kersten:

Oh.

Brendan Hopper:

And it ran CPM and so there wasn't a lot to do except for work out how it worked and try to make it work better and build things on it. So I got into programming very young. And then when the internet of became a big thing in Australia when I was about 11 or 12, not a big thing, but I was probably an early adopter for the region. I very, very quickly got into the online cybersecurity community and met some people who were involved in vulnerability research and exploit development. And they were a few years older than me so I learned a lot from them about very low level details of how operating systems work and how you analyze code for security vulnerabilities and how you write exploits. And I got very involved in that community.I started university but I wasn't actually able to finish. I couldn't afford to do university and not work and I couldn't cope with working and doing universities. So I actually never finished my bachelor's degree. And when I was about 18 I made a vow never to do information security at the time because cyber security wasn't a word they used yet. And then by the time I was 21 I was doing it as a full time career.

So very quickly walked that vow back. I spent about 10 years as a pen tester, red teamer as that industry was emerging. Also just doing vulnerability analysis and exploit development, researching, helping companies design and develop more secure products. And I worked for a very small firm for a long time over for five or six years. That was 10 people in Australia and 10 people in New Zealand. But we were very high end attack testers. So we in Australia rotated around our big four banks, and helped them make their security better.

And then 10 years ago, almost 10 years ago today I joined Com Bank as a security tester and then started eventually leading teams and growing through this cybersecurity team there. And I built our first ABSEC practice, helped the group for me discovery, did attack architecture, defense, rotated around all the roles in cyber until a couple of years ago where I decided to make a vow once again to stop doing cybersecurity professionally and flip over. And now I run engineering and technology strategy for CBA.

Mik Kersten:

And so what's behind that vow? Actually tell us a little bit more what it was like in those days as you were learning how to ... both the experience of moving from bank to bank but then really scaling this out and the practices out that ... at Commonwealth Bank, how did you start thinking about the engineering aspect of it? What actually brought you to that, to considering strategy and engineering in terms of what you were learning on the security side?

Brendan Hopper:

There's probably two things that went into me making a decision to put down the swords or stop doing.

I always think about that Kill Bill, like a Tori Hanzo style vow, never to do cyber again, never to make swords. One was a personal thing where in cybersecurity, very hard job. I respect everyone in that industry so much, particularly on the defense side because you are not really building ... no one ever thanks you when you're doing incident response, when you go and cut someone's access because they're being compromised. No one's like, "Thank you for stopping me doing the thing that I wanted to do today." So people in that part of technology that they work very hard and primarily though they're not creating delight. They're not building things that make people delighted. They're making sure that things that other people build don't go wrong and sometimes don't go catastrophically wrong. Super, super important. But for me personally, I didn't want to spend my whole life or my whole career just on that, stopping other people. I wanted to start building things myself or being more involved in the creative side.

And then the second element was also from a CBA perspective, I looked at it as ... if you think about when I started in vulnerability research when I was 11 or 12, you could get a piece of open source software, you could find a vulnerability in a day and you could write an exploit for it on that same day. You could basically have an attack tool in one day. Now that's like nine months of work that's actually professionals who are specialists. Sometimes one person finds the vulnerability, another specialist writes the exploit, a third person or a team of people write the framework that happens from.

And so I saw this curve of sophistication of attack rising and I realized that effectively in order to stay

safe, companies were actually going to have to start being able to deliver faster and faster and faster to outpace the kind of development of attack, but also to think very differently about how you engineer so that you reduce the blast radius of every element. So the worst thing that can go wrong isn't catastrophic. And that made me really realize maybe the best spot for me would be to flip across and start running engineering and start focusing on how we get that. You talk sometimes about this, the tech giants with their a hundred thousand time software velocity. How do we get that in a big bank while still being as secure as a bank requires and our regulators require? As soon as I started thinking about that, I fell in love with the problem and the rest is history.

Mik Kersten:

So that velocity would give the bank safety you mean?

Brendan Hopper:

Absolutely. And I remember at the time I was actually realizing that the way conventional security is done is through caution or the way that all safety things are done in sort of manual industries is through caution. And I sort of mentally pictured ... In Australia we have ... particularly where I grew up, there's lots of places where you can effectively drive your car or your motorbike around the mountains which are right near the ocean. And there's all these signs that are like, "Beware of falling rocks." And I sort of realize that the conventional approach to beware of falling rocks is to walk really, really slowly and look up all the time, but that actually a certain amount of velocity and agility if you are fast enough and nimble enough and able to move quick enough, you actually are too fast for the rocks. And so I had that mental picture and I realized that that's the from and to.

Mik Kersten:

Brendan, I'm glad when we were involved here in my area of the world in British Columbia, Canada where it says, "No stopping falling rocks." So ...

Brendan Hopper:

Right. Yeah, you're never going to get the agile of bush walking there then.

Mik Kersten:

No, that's right. So I guess this was a bit of an epiphany for you, is that caution and making sure everything is ... For me, I actually had a similar one, not with anyone near the depth of yours, but I'll just reflect on it briefly. But when one of my colleagues was actually responsible for some of the portions of Azure and the development supporting security initiatives on Azure, he was explaining to me how much of a pivot was for them. This is a few years ago obviously, or quite a few years ago, to move away from constantly thinking about meantime to failure and looking at how to be more safe, more cautious, and that everything was turning into meantime to repair and ... or meantime to restore service, how quickly they could actually evolve this complex service to patch things to respond to react. And the velocity would drive their safety. In the end, was it the scope of ... and the nature of the attacks that's a really interesting story. Is that the sophistication, was that the main thing that had you realize this? Is there anything else to that genesis?

Brendan Hopper:

It was probably a confluence of a few things as well. I read that Taylor book Anti-fragile and that whole concept anti fragility and started to think far more about security in that context as well. I got, when I joined the vulnerability research community, when I was 11 or 12, there was maybe a hundred people in a bunch of chat rooms on IRC. And so I've stayed connected with a lot of people that have spread across the tech ecosystem. So I've got ... just stayed in contact with what's happening with my friends and some of them are in the tech giants and some of them are basically spread out everywhere. And it roughly ... the Google and their whole zero trust movement over the last 10 years that clicked for me that that's part of the whole anti-fragile thing. Don't make each individual component really strong, make each in individual component, really ... make sure the system is strong, which means that sometimes you have to cater for components failing. And at the same time cyber as an industry was going through this shift where in the 90s, the 2000s, the 2010s, you thought about protecting a castle, keep people on the outside. And then it shifted into very much an immune system. Sometimes people will get in, track them down, develop antibodies and then don't just make yourself safe, share that as thread intel across the industry to keep everyone safe. And so it was all those factors of thinking that

came together that made me realize that velocity and safety and security are ... you'll hit a point where caution will no longer actually be acceptable, an acceptable solution.

Mik Kersten:

Yeah, and I think that's what we're seeing as systems get more complex that that's exactly where things are ending up. And I think security is just one example of it. So then tell us what happens next as you get into more of the strategy and the engineering side of things. Tell us more about your journey.

Brendan Hopper:

It was very, very interesting. As soon as we started doing this sort of engineering push velocity push to listen to our people to go and talk to our engineers who are building products, the first month of the role, I spent a quarter of my time, a third of my time reading books and connecting with people like you who are leaders in the industry and selling that ... driving the message and the transformative story and the rest of my time just right at the front listening to people who were trying to ship product to our customers and what problems they we're facing. And I actually realized that some things that I'd done, some rules I'd made, some solutions I'd put in place on cybersecurity were actually really slowing people down. They were adding the safety but they were almost making the developers jobs untenable because we've been very prescriptive about the how. "You'll do it this way," rather than giving them the outcome and letting them solve themselves. And then when you go deeper into that problem, it's very much about, I think successful organizations are where all the different teams recognize and respect and appreciate that each other is an expert at their own domain and it's about sharing knowledge and sharing problems rather than solving them in silos. So yeah, that was the first couple of months on the job

Mik Kersten:

And then that's the point at which you started detecting the amount of burden. And this is of course common, this is not specific to the organization or the part of the organization you're working with, but just the sheer amount of burden that's there in a typically large organizations with tens of thousands of IT staff and developers. So tell us a bit about that portion of the journey. So some of what you were seeing obviously is some of the security and compliance related items. What I've been really impressed

with obviously is how you've actually helped bring CBA forward in terms of recognizing this and in terms of optimizing for flow, looking at optimizing for autonomy and really starting to understand the economics of flow. Because otherwise what happens is, and what we've seen this across various industries and the amount of burden, compliance burden, the amount of security burden and safety burden that's put on teams and entire value streams slows the velocity to a halt and which is to your point, counterproductive to the thing that we're trying to address, which is safety in this case. So how did you start seeing this as you started working with the teams? The key thing is you got into the books, but what did you do next on this journey of discovery?

Brendan Hopper:

The first thing was that very clear talking to the teams, going and talking to a team of developers and they were frustrated that they was still shipping amazing products. And just so everyone on the call Commonwealth Bank might not be broadly known around the world, but in Australia we're systemically important. We have seven to eight million digitally active customers every day on mobile app. Our app's been rated number one in Australia by Forester for six years in a row now. So they were still shipping this amazing product and we still are and it is much, much better now than it was two years ago. So we're starting to work how to do safety and velocity together through automation. But the first real thing was that developers were struggling but they weren't really struggling to get the work done. They were struggling emotionally. What they were doing was working more to compensate for how much tension was in the process. Ops and security were very far away from development and so they were emotionally making up for that.

The second thing was when we brought the people into the room together, the first 10 minutes I remember very clearly, I got a bunch of people who worked on the development tools and some of the security people and a bunch of our developers put them in a room and just had a listening session. And it was the first half an hour everyone just had to get things off their chest. And then they actually realized they all worked for the same company and they started solving together. And so a little bit of what was just about connecting. And then immediately the next step I took was to go and engage at the highest levels of our organization and build awareness because the way we'd structured ourselves up

top with our executive forums was almost like they would discuss features and velocity in product and then on a separate day they'd discuss safety and risk. And by I really realized that unless we solved it with the CEO and the executive leadership team and to some extent the next layer down, unless they understood the problem and the need to bring it together, we weren't going to be able to succeed organizationally wide.

Mik Kersten:
So Bren, that's really interesting and I think that in my view at least a big part of the reason for your successes in the last months and years has been the way that the CEO and executive team got engaged. And of course that's how I got the opportunity to meet you as well. So how did you pull that off, right? Because all of a sudden you've got a level of sponsorship and engagement that I think is absolutely transformational. I had on a previous podcast with Christina Yadi, he and I both reflected on the fact that the fastest, most successful transformations we've seen at large scales and he's in the Silicon Valley product group, had CO involvement but closer involvement and some degree of oversight and of course the empowerment that's needed as well. But how did you pull this off at CBA? How did your CEO actually become so close to the transformation?

Brendan Hopper:
I mean first and I very lucky maybe the confluence of some accidental forces and then also just having a really good CEO and executive leadership team. But coming from cybersecurity and being in the group for 10 years or eight years at the time, I developed a lot of trust and I'd probably not said a lot to the key people around our executive leadership team, but every time I had said something it was pretty much really, really important because cyber becomes a conversation very quickly. When there's a cyber issue, it goes up through the organization really fast. And so I had this reputation for trust and then I effectively went straight to them with my diagnosis of the problem and said, "I need your buy-in." I remember talking to our CEO and saying, this isn't the CIO's problem, this is a group-wide problem that you have to sponsor.
And then he actually talked to a bunch of other CEOs of tech companies and things and bounced that

idea off them, like, "Is tech transformation the job of the CIO, the CTO or the CEO? And he unanimously got this story back that it was the CEO that had to sponsor it for it to be a success." And that's made it a lot easier. But in some ways it's also added its own difficulties too. And I think there's always a trade off there where you engage. If you engage with the tech organization first that has a set of pros and cons, but if you engage with the business units first, that has a different set of pros and cons.

Mik Kersten:

Yeah, but I think there's a lot to unpack there because I think again where the CO perceives it as their problem, the kind of speed to value in terms of the transformation I think is an order of magnitude different. Even of course it might surface a whole other series of problems. But again, I think this is such a key success pattern and in so many cases where it's limited to the CIO, the CTO, there is this, when it involves a change in the business, when it involves a change in operations, when it involves looking at the entire management system, when it involves bringing a CFO to the table to help the CFO understand some of the economics around software development or just the economics of flow are much different in the economies of scale that many have grown up with. And there's another whole host of problems, but I think it's led to a much faster path. So can you just take us a bit more through that path right now, how it's unfolded and where you're focused today?

Brendan Hopper:

It's definitely been faster this way, but it's probably all ... the price of speed or velocity of transformation is always disruption and friction. Maybe some lessons I've learned along the way, One is that in big organizations ... Maybe a side conversation in Australia we have these birds called lyre birds and they actually, they listen to all the other birds and then they make the same sound but they're really good at it. And so with the rise of modern society, you can walk around and hear lyre birds making the sound of chainsaws and car alarms and you could see lyre birds near raves that play dubstep, so [inaudible 00:22:28], and they're really cool. And what I've realized is that lots of people inside big organizations are repeating a message and they're spreading it and they're amplifying it but they don't necessarily understand it. And so it's really important that when you're talking to people that you gauge out, do you actually understand or are you repeating? But it's also really important to make sure that the message is spot on because of this phenomenon through large organizations that whatever you say as a senior leader is just going to get repeated a lot. So you need to be really clear about your soundbites and just

be making sure that your transformational messages as simple and crisp. And actually use that sort of liar bird network as a positive, not necessarily a detriment.

The second thing is, I don't know, is that experiment years ago that people have only just started to realize how smart octopuses are. And part of that is that octopuses are pretty good at hiding their intelligence. And I remember reading this study about this octopus that was squirting water onto a light switch so that it would turn the lock off on its tank so it could sneak out and do things at night. And I form this opinion that there's people who are very, very influential across all companies, all organizations, all industries. And the vast majority of what they do is watch and they don't say a lot, but what they do say is extremely impactful. And so I think there's a big part of transformation which is finding those people and just be because of the way they act, they have a huge amount of credibility and a huge amount of authority, but also because they spend their time observing and trying to understand rather than trying to get their own message across or to get the repeated message across, they're also the people that are the best people to be able to give you advice and course correct you and make sure you don't make mistakes before you could make them.

Mik Kersten:

Okay. So I think that's absolutely fascinating. And I have to say I did not have an appreciation for a lyre birds until I actually went to Australia and in July and people did play many lyre birds sounds on their phones to me and hopefully next time I actually get to see one. But I think it's really interesting that you see lyre birds as a valuable mechanism as well, as that they are a key part of the organization and the way that organizational message and culture and practices disseminate. So I think one of the things that you've done so effectively is to help CBA create this and to create this transformation network that actually that is learning, that is course correcting. So is that how you think of creating ... of that network as just a MCS or tessellation of octopuses and lyre birds or how do you think about that?

Brendan Hopper:

Yeah, so first of all, the lyre bird thing I think is ... it's not a bad thing and I don't think a person is necessarily a lyre bird. I mean you look at children, they start saying words long before they know what they mean. And all languages no one knows with a hundred percent accuracy exactly what a lot of tech jargon is, and the definition of things. Back to cybersecurity, definition of zero trust, that means 50

things to 49 people. It just means so many different things. And so it's a normal part of communication to repeat what you've heard, to try to sound out what it means. And that is definitely an asset but also a sign that people are trying to understand and break the message down and get into it. And then everyone is a lyre bird on some subjects. The octopuses thing, I think that's something that takes work. I think you have to work really hard to become a quiet watcher, but they're ultimately the most influential people. I think that transformation, where it can go very wrong is if you forget that primarily you're transforming people, you're trying to give them new skills, you're not trying to change their skills, you never learn something and forget something else. You're always trying to add a skill to people. And I think ultimately you have to realize that everyone is a real person with a real life and work is work and life is life. And if you come from that perspective of actually, how do you transform your workforce in terms of giving everyone the extra skills that are going to take the whole organization in the right direction, that's the right first step. And then the tech and the measurement and the metrics, they become mechanisms that reinforce your people transformation, not the other way around.

Mik Kersten:

Well so let's dig into that. So I'll give you an example of a bank that was making the shift from project to product for the name of the podcast. And in their transformation they issued an announcement as there were some [inaudible 00:27:44] involved that over 15000 project managers were now going to have the title of product manager. So they transformed their people overnight or not. You've obviously taken a different approach that's been a combination of people, technology, process, the way you've organized crews and so on. So I think we're seeing so many missteps out there right now with again and with a desire to speed up transformations rather than take this measured driven incremental but still fast approach. I think it's a pretty deep point that in the end what you are doing is transforming people and of course we're transforming the way people collaborate, communicate, plan, the burden that's in the system trying to remove it out of the way. So share with us some of your lessons on that.

Brendan Hopper:

You've got to be super careful of that renaming people and expecting them to be different is what I call magic thinking. It's like you're baking bread but you decide before you put it in the oven to call the salt sugar and you expect cookies to come out the other side. It's not going to make a difference. You

actually have to change the skill set, change the mindset. And not change the skillset, grow the skill set, grow the mindset, get people to understand the why behind it. And I think that everyone that I've spoken to across the industry in a large company who's been successful is like it's taken five years, it's taken seven years, it's taken five years and we're halfway there are what they say, but they're like ... I also get a lot of caution on, but you can't tell people it's going to take five years because then they don't drive and push the transformation it needs to be done in 18 months. In reality, it actually takes forever. You have to constantly transform and you have to constantly do it with this passion of a growth mindset. But also a lot of the success stories in other industries than banking that I've heard, they're driven by desperation. Unless we transform, we won't stay in business. It doesn't necessarily I think need to be driven by desperation, but it needs that level of energy and investment and then it needs the ability to sustain it for that five seven year roadmap. And understand along the way you're going to get things wrong and that's okay. And when I look at transformation, as long as you look after people and you're looking after the people in your organization, and as long as you are focusing a lot on measuring whether you are improving rather than measuring where you are, I think you're in a pretty good spot.

Mik Kersten:

Right. And I think that's obviously where our partnership is formed around is that my whole view has been we just need a feedback cycle on the transformation. Are we making things easier for people? The fact that across enterprise organizations we see a very strong correlation between flow velocity improving and employee engagement improving because the only way to really improve flow velocity is not to hire 50% more people. It's to remove burden and waste from the system to make it easier for people to do their work, to make it easier to make our large enterprises as easy for new hires from tech giants to thrive in which of course from any organization is organizations is not the case. So I think Brendan can you dig into that because I think you have had ... how you got to this notion of measuring flow because I think you've been able to leverage it in order to actually become more data driven on the transformation and have this feedback loop of where things are getting better, where to prioritize things. I think one of the things that I've really learned from you that I've personally been applying a lot is to actually understand was, well where are our product pathways that we really need to focus on flow? Which are the project pathways where we're going to leverage or continue ... leave more traditional ways of working for the time being. So yeah, tell us a bit about your approach and how you've been thinking about that kind of measurement.

Brendan Hopper:

I think about it in two ways and I think about it at the value stream level for each owner of a value stream. How do you give them the autonomy and the data and the information so they can do their best job possible? And I think the flow framework and task tops starting to help us a lot with that. Putting data at the fingertips of what we call it a crew, some companies call it a pod, but whatever unit of work is one value stream. Making sure that they have the information to make the right local decisions for their products and their value stream is the first thing I think about. It's not about making their decisions for them, it's about letting them make their decisions. And then it's two, across the entire organization, how do you treat it like a factory and work out where the constraint is and then go and solve for it. I remember maybe seven, eight years ago at CBA all work used to pass through the penetration test. Every product we shipped, whether it was inside, outside, whether it was our most important product or a test product used to require a security test and we couldn't hire security tests as fast enough and they became the central constraint. I remember at the time people had this belief that that would lower the cost of the organization because that central constraint would then mean that everything else automatically corrects itself and we just invest less and spend less and we do less projects. And what actually happens of course is that people wait and they just start producing extra work for each other to do. Ah, because they're saying idle hands of the devil's play thing. Super true in an enterprise. If people aren't actually working on pulling value through for the customer, they're probably working on creating work that doesn't need to be done. So identifying that is one thing that we're really focusing on as well.

Mik Kersten:

Okay, and so that's been your approach. I've been seeing so much more of this and trying to understand better as well is that at that agile team, that feature team, the scrum team level, this we understand things pretty well, but it is those next two levels like the crew pod, that value stream level is making sure that we actually understand the right metrics, we give them empowerment and autonomy to make those decisions. And then of course that organizational, that systemic level, what systemic conditions do we need to change? And where the bottlenecks is the only way we can get insights across the teams is the only way. And so organizations and digital natives tend to be good at this. The reason they see the need to invest in let's say security automation, vulnerability automation, simulations, attack simulations, tech giants increasingly invest in that because they constantly know what their bottleneck is and behold,

oftentimes it actually is related to security. So I remember on the podcast as well, I asked Andrew Cockcroft [inaudible 00:35:01] Amazon was where's the bottleneck today? And then actually it was answers around needing to automate more security and that becoming a key bottleneck. So I guess that sounds like a guidance, Brendan to understand that data that flow the bottlenecks both at that value stream level and then at the organization level.

Brendan Hopper:

Yeah, absolutely. And then the other thing would be security is shifting and risk management and resilience. They're all shifting away from this. And I don't mean for our company, I mean across the globe from what I observe, shifting away from this management of the likelihood of an occurrence of event and towards minimizing the impact, like blast radius reduction. And then I think one of my key things I've learned is that blast radius reduction where you are comfortable setting the blast radius, an outage at any organization is bad at a bank, lots of things have to go wrong, like an airplane crash, it's never one mistake, it's a sequence of things that have to go wrong because we've engineered a lot of safety into the processes but it still does happen. And so if you look at the actual, assume bad things will happen, what's your acceptable blast radius? And then make sure that you provide your autonomy and velocity to that blast radius size team. You won't have a blast radius that's a business unit, it'll always be a value stream or a sub-part of the value stream, which means that it becomes more and more important to make them autonomous. And those two concepts are directly connected. They're actually the same thing. Your autonomy is your blast radius. If you are running a system that can destroy an entire ... if it's down, the entire organization is down, you basically need to get the entire organization to agree every time you say something. So by minimizing blast radius you add autonomy.

Mik Kersten:

Okay, so that's interesting. You're basically describing that Swiss cheese model of failure where the that slices in terms of the vulnerabilities are, we imagine them as the slice of Swiss cheese and for a plane to go down those holes in the Swiss cheese to align. So how do you actually apply that to your organizational design? Because you're saying each one of those failure planes is responsible for its own blast radius. So you give them autonomy to move fast and restore or repair or? Can you dig in, this is super interesting. Can you dig into that a bit more?

Brendan Hopper:

It is really hard and it takes years and this is the deep tech side from my perspective of one of the reasons why these transformations are forever but five years is the minimum to really see a huge step change, is because that enterprise IT, how IT processes or how enterprises ran of before the emergence of the tech giants or before the emergence of DevOps, it was all about careful coordination And the economics pre-cloud was all about it's better to rationalize and have one of something. And so lots of organizations have ended up with big platforms that lots of things are dependent on. And that's okay if the platform is a team and you architect the system so that every consuming value stream has its own instance. And so then it starts looking like the reverse Conway maneuver. Architect and design your systems to give your squads and your crews and your domains or your various levels of hierarchy autonomy from each other at their own level through separating the blast radius out, asking the question, can you actually safely change without impacting outside of your crew? If not, then your blast radius is at the domain level. And there are terms domain, crew squad, but across the layers of the organization, understanding the impact is the same as understanding who has to sign off on the change. And then it's like being really, really clear. Everything new you build, you want to encourage that autonomy and then making sure that all of your enabling platforms are built from the ground up or redevelop from the ground up to encourage that incremental autonomy and reduction of blast radius and risk.

Mik Kersten:

Yeah, that's I think such a key success pattern. I've absolutely seen it. I've not seen it characterize the way that you're talking about where it's around that blast radius, but it makes so much sense, right? Because the thing that we see is that the more autonomy there is between value streams to the point where there will actually be duplication so that they can optimize around their flow and around supporting or fixing issues around their blaster radius very quickly, the higher the overall velocity of the organization whole as a whole, which is very counterintuitive when you look at old school ways of looking at IT service management enterprise architecture where there should be just one of everything and in the end that actually causes ... Again, I think that's really unique because what we see, it causes ball necks actually you say it causes much larger, in this case failure planes or outages or security or attack surfaces. So fascinating.

So then Brendan, it is just a tricky question and it's one I've been asking every octopus I know and you certainly qualified, how do you think about the ownership model for this and the organizational design model, right? Because you need ... let's just go back to your ... so your value stream, autonomy, the data, the information and you think about ... and then your number one thing, number two, that whole organizational understanding, that systemic understanding, what's the ownership and the organizational structure that you find effective for that? Because in the end it's like you're saying that that tools can provide the data, they can provide the flow metrics, they can help with roadmaps and OKRs and really, really providing that data, but in the end, the way that squads and teams are structured and organized, and then in that value stream case and the way that the ownership for the systemic conditions, so for improving that bottleneck, whether it's security today more it's user experience design tomorrow, how do you think about the ownership structures for that? Especially with the complexity of moving in organization. Imagine now you're an octopus, you're advising the next Brendan Hopper or the next bank who's got an org chart that's matched up to very old ways of working with a lot of separation with IT, with security being someone else's problem, how do you actually think of the leadership and org structure principles around this?

Brendan Hopper:

Ooh, that is a hard question. First

Mik Kersten:

I don't have an

Brendan Hopper:

First of all

Mik Kersten:

That's why I keep asking the question because I don't ... I'm not

Brendan Hopper:

I-

Mik Kersten:

... Sure there's a single answer but I'd like to know your current take on it.

Brendan Hopper:

People love to say org structure doesn't matter. I'd love that to be true, I think it really, really matters. I think lesson number one, org structure has to continually change. You wouldn't design a product for a customer and then say the design that we have now is going to be the same as the design five years from now and we're not going to listen to our customers. I think you have to over time gradually design your org structure around your products. You should start with what does the customer want and what product meets their needs? And then what's the architecture that supports that, minimizes the blast radius, and lets the team who are building that the right autonomy. And from then when you've mapped that out, then you can start working out, "Okay," Because it's almost like a ... once you've got that level and that diagram, your org structure's pretty much done. You just put names in boxes. Very, very hard to start taking that approach to software, but I think it's essential because if you can tell an organization's org structure from looking at their APIs, that's a problem. One of the people we have. Victoria later, she's taught me a lot about that. If you can tell an org structure looking at someone's API or looking at their applications, there's an issue. The only solution I can think of that is work out what your customers need, what your applications should look like and structure around.

Mik Kersten:

But that is to your point, just reverse in Conway's law, right? Because your APIs will look like your org structure, but

Brendan Hopper:

Right, and then

Mik Kersten:

... If that's not matching the business problem, change your org structure.

Brendan Hopper:

And have the mechanisms so that you can change whatever portion of your org structure needs to move as fast as your software architecture in place. If you need to change someone's reporting line every time you want to change the flow of power around the organization so you can change your software architecture, it's not going to work. Your constraint's going to become the HR department. Which is why if you look at how tech companies are structured, they're very much more structured like there's pools of people who are then linked to initiatives. And I think one of the reasons why that seems to succeed is because you can be far more agile. The reason org structure matters and reflecting the software design is actually the flow of funding and power and decision making up and down the organization. And so if you can separate that from the human resources side of it, then you're in a much better position to be agile because you need to be as agile with your org structure as you do with your software.

Mik Kersten:

Yeah. And I think your point there, Brendan, is that the org structure has to continually change. I have not heard that ... actually have not heard those words said exactly that way before and I just could not agree more. And I think there's been so much concern around disrupting teams. But in the end, if you're actually … So here's something we've seen in the data of organizations in terms of their flow metrics, it's actually ... whereas it's really expensive to break and reform teams, of course that takes a lot of time, it's actually a much cheaper operation if we think of this as a refactor operations on the organization because we're factoring in the end, the org structure then that impacts the software architecture which then then impacts the product value streams. Moving teams between value streams is actually a lot cheaper because if you've got a bottleneck, you move one of your agile teams into that value stream and you bring, let's say the front and closer to the API or to a piece of the infrastructure or something into that four quarter or two. And of course doing that in a data driven way to see did that improve things or did the not improve things by measuring the flow is ... we certainly look at it in terms of my own teams every quarter as we look at the org structure and changing the org structure every single time we do our road mapping and OKR planning, the quarterly OKR planning. So I think that's such key

advice. So anything else more on that? Because I think that's a learning that's not percolated out there enough.

Brendan Hopper:

Yeah, you and I were talking before, one more point and you laughed a lot because I said that sometimes agile coaches are the least agile people in the organization.

Mik Kersten:

Please don't tell anyone I laughed at that.

Brendan Hopper:

That's actually also really related to the org structure conversation because this core principle that you should always be evolutionary and you should focus on small increments, not major redesigns needs to apply to everything. It needs to apply to how you approach your org structure, how you approach your metrics, how you ... revolutions don't work or revolutions that do work are actually just the sum of thousands of evolutionary steps. And agile coaches, I think when they become non agile, the opposite of agile is when they try to move too far in one step and they start focusing very much on, "We should look like A, not B, that's wrong and not, hey, we're very far away from A, let's move a fifth of the way there." And so I think that if you can apply this incremental approach to absolutely everything, that's probably like my one last message.

Mik Kersten:

That's awesome. That's awesome. And I could not agree more. And of course we've got the coaches out there who take that approach, who take a small batch approach to

Brendan Hopper:

Oh, yeah [inaudible 00:47:53]

Mik Kersten:

Transformation. Yeah. But this

Brendan Hopper:

I don't want [inaudible 00:47:56] agile approaches across earth because they're probably half of them listen to your podcast, most of the time they're not like that, they could just get stuck in that groove occasionally.

Mik Kersten:

Well no, I think there's a disconcerting number of agile transformations happening in a very waterfall way. And I think that's a really good way to characterize it, is that we need to go from A to B in this timeframe, in this number of months and implement these ceremonies. And there is a lot of this waterfall and flexible large batch thinking around it. And I think challenging that is key

Brendan Hopper:

Right, like, "Give me a project's plan for how we're going to roll out agile."

Mik Kersten:

Exactly.

Brendan Hopper:

Yeah. You're like, "I'm not sure, are we really going to get there if that's our approach?"

Mik Kersten:

Yeah, that's right. And the way that you bring in these, the fact that other things have to change and taking that fifth step of the way, but changing the architecture and the organizational structure and the team structure and the alignment of the value streams, that fifth every quarter of course allows you to

course correct so much more. So Brendan, this is fantastic. Any other final words of wisdom or guidance for everyone sitting in your shoes elsewhere?

Brendan Hopper:

No, thank you so much for having me on the show. I think it's all about patience, persistence, but also passion. And then just once again reiterate, it's a transformation of people and they're real people. And if you just keep that in mind and understand that it's going to take a long time, but it's worthwhile, then you can't go wrong.

Mik Kersten:

Amazing. Thank you so much.

Brendan Hopper:

Thanks mate.

Mik Kersten:

Thank you to Brendan Hopper for sharing some of his expertise with us today. For more follow me in my journey on LinkedIn, Twitter, or using the hashtag #Mik+one#ProjectToProduct, you can also find Brendan on LinkedIn. I have a new episode every few weeks, so hit subscribe to join us again. You can also search for project to product of the book and remember that all offered proceeds go to supporting women a minorities in technology. Also don't forget to join the flow framing community on Slack, which you can find on flowframework.org. Thanks, and until next time