Episode 33: Scott Prugh
Episode Transcription

Mik Kersten (00:06):
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, Founder and CEO of Tasktop, and bestselling author of Project to Product: How to Survive and Thrive in the Age of Digital Disruption with the Flow Framework. Joining me on today's episode is Scott Prugh, Chief Technology Officer at CSG. Scott is an industry-recognized leader and international speaker on DevOps and enterprise transformation, and sits on the programming committee for DevOps Enterprise Summit.

Mik Kersten (00:39):
He's a 20-year technology industry veteran who has broad experience building and leading high performing teams across development and operations functions for companies ranging from small startups to large enterprises. Scott and I chat regularly, and he recently introduced me to his revamped DevOps portfolio management approach. I find the clarity of his thinking and his approach to elevate the principle of DevOps to the business spot-on, and I'm always amazed and entertained by how his stories bring these principles to life. Listen on to hear about Scott's take on value stream and portfolio management, architecture, and culture.

Mik Kersten (01:14):
Welcome to the Project to Product podcast everyone. I am here with Scott Prugh, CTO at CSG. Scott and I have been talking flow and value streams and product transformations for many years now. He's been an inspiration to me in terms of his work, in terms of the achievements he's had at CSG and elsewhere. And I think he's actually one of the foremost thinkers in terms of how he transitioned from proxy metrics and activities to actually stable product value streams and measuring the flow of value. So I'm quite excited to have him here. Scott has put together something that's been just, I think, a fascinating model around DevOps portfolio management, in these three domains. So, Scott, I can't wait to dig into that. But first, I would just love you to recount some of that story that actually introduced you to me, which was year 2014 DevOps Enterprise Summit talk. And I think it had something to do with print shops. So refresh my memory, and for those who haven't heard it, if you'd tell us that story, that'd be excellent.

Scott Prugh (02:07):
Excellent. And, Mik, thank you so much. I'm really honored to be here. And I appreciate the praise and the partnership that we've had over the years. It has really been great. And I've learned way more from you than you from me. So I think how I originally got into this community is really a bit of the story that you talked about. I was really working on kind of work transformation and also agile transformation at CSG. We have a print shop. And we would always go visit the print shop because it’s just really fascinating to see. We sent about 70 million pieces of mail out every month.

Scott Prugh (02:49):
The managers of the print shop, they really run a fantastic system, very lean. The have this row, and they call it row one. The manager's always excited and was always showing me row one. He's like, "Hey look at this. There's all these carts." I said, "Well, what do those carts do?" He said, "Well, these carts represent jobs that we put in the system." Now, I kind of scratched my head and I'm like, "Well, you've got computers. And we actually fill those computers with all of the jobs you're supposed to print. Why do you need these carts?" He's like, "Because this represents our job and materials released. Each one of these carts represents a job of millions of pieces of paper. And on the cart we actually have tagged the materials that we need. And when someone wants to put a job in the system, they have to go over to a cart, and the physical activity of taking that cart, moving it, and putting it in the system is significant because they're releasing work."

Scott Prugh (03:40):
I was like, "Well, that's fascinating." And so what I would do is I would then take our IT project managers to the same spot on the floor. We'd drive over there and we'd look at it. And I would say, "Hey, tell them how this works." Then I would say, "When we release work into our IT system, i.e., we start epics, we call them, which are these large efforts, how do we do this? How do we understand that we have the jobs and materials, i.e., the people and the money and all that other stuff, to do the work?" And I would always really kind of get just kind of blank stares because they didn't know.

Scott Prugh (04:13):
And so I thought that was kind of a fascinating story. And then when I read the Phoenix Project the first time, I was convinced, and everyone has this experience when they read it, that Gene Kim was following them around, because the story of the Phoenix Project was, of course, part of it was about manufacturing, and really the relationship between manufacturing and IT. And I found myself kind of in that same world. So kind of got introduced to the DevOps community through Gene Kim, and through that story, because I told him that same story and he called me the next day.

Mik Kersten (04:46):
I think there's so much to unpack in that story because there's queues, there's the visibility, the fact that work is visible, the release of work, and that intake of work is visible. And then of course all the work in progress and everything else that we deal with day to day. So what happened then? What did you do with that inspiration from the print shop? What you learned from, I think, like you said, a lot of us realized Gene just characterized what we're seeing everywhere. You illustrated it. Or you saw it, actually, within your own organization. What did you do next?

Scott Prugh (05:18):
Yeah, so I mean that story's from a timeframe, let's call it about 2012, when we were struggling with this problem of just not being able to complete work. We would look at projects. We would actually look at epics. Actually, the software development side of our organization had epics. Our IT side of the organization had projects. And that's another kind of piece of the story. But at the end of the day, we were not completing what we wanted to complete very quickly.

Scott Prugh (05:46):
Things would just stall. Epics and projects would sit out there for years and not get completed. So we continued to look at this and try to understand why we weren't finishing things. And at the end of the day, what we had discovered is we would have close to about 200 large efforts in flight kind of at one time. And we were only completing a couple of them a year. And so we went through a rigorous analysis of collecting and really getting all that work together. Well, it's not even analysis, it's just collecting. Do we have that list of everything?

Scott Prugh (06:15):
And then really looking at that, prioritizing that list, but also really looking at the dependencies, and in essence, the kind of job and materials, like the different people and teams that were required to deliver it, and began to kind of limit the WIP that we put into the system. And we started looking at it. We did some kind of simple dependency. We would find certain teams, and of course the highly centralized teams, especially around system administration, network, storage, other shared groups, some of them were 300% overbooked. And we would then have teams waiting on them. So basically, what would happen is you would have a team that says, "Hey, I want to start some work." And they'd be like, "Well, I can do this work." But the dependencies on these other teams started to then become kind of bottlenecks in the system that they couldn't complete the work because they had dependency on another team.

Scott Prugh (07:09):
And so what we had to start doing is one, recognize that we couldn’t start that much work at once. But also, we had to look at the work and make decisions around the dependencies of what we could and couldn't start. So you can't start seven projects that all depend on the same seven teams. We had to basically make differentiated decisions about what we could get done based on those dependencies.

Mik Kersten (07:32):
So, Scott, I imagine to any print shop floor manager, that having 300% overbooking sounds completely crazy.

Scott Prugh (07:40):
Oh yeah.

Mik Kersten (07:41):
But the thing that I'm seeing, and I think as you just illustrated, is in almost every single large organization, the amount of flow or WIP that we see compared to the capacity is two, three, four, five X. We're now looking at queues of epics, of features and the like, I used to be shocked by it about a couple years ago. I'm no longer surprised to see 24 month queues. And meantime, of course, the crazy thing is the load and the backlogs are growing. So in your experience, because I think you're ahead of the curve given that you were having these thoughts and conversations in 2012 to 2014, how do we end up here? How do organizations actually end up with this much of a mismatch between capacity and understanding of capacity and the work that's actually been booked, the work that's been committed, the work that's been promised to customers?

Scott Prugh (08:39):
Well, I think of how end up here. That's the easy ... There's a couple reasons, and I think that's the easy side of it. The hard side is, okay, how do we fix it? And that's a continued struggle. So I think the way we end up here with this problem is ideas are easy, priorities change, and it's hard to stop work. And so you kind of look at those and you say, "Hey, you come up with an idea." And it may be very viable and a great business idea, but you want to start that, because it may be urgent, but you've got other work that you've already committed to. And then it's really hard to stop that work.

Scott Prugh (09:20):
And the further you get away from the teams, the understanding of what they're working on gets kind of less and less. So it's easy for sales, or executives, to kind of come up with these new ideas, and then be like, "Hey, well I want the teams to work on this." But they just don't have the mechanisms to visualize and communicate out, "Well, we're already working on these things. And we're already 200 or 300% overbooked. And if you add this one more thing, we're then 400% overbooked and everything's going to take longer." And I think there's a lot of that that also sits underneath this problem with virtualized work. And I don't know if it's a problem, per se. It's just the nature that virtualized work is something that you can't touch or see. And in a factory, you could walk up and you could see carts, or you could see printers backed up. And you would know at least to stop the work. But in a virtualized environment, you can't see those things.

Scott Prugh (10:22):
The entire organization, and we've been fighting this problem for years, we still struggle with it all the time. It's tools that you brought to market with Tasktop and Viz that really help illustrate that to everyone to say,

"Look, this is how long things are going to take right now. If you add something else, it's not going to get better."

Mik Kersten (10:41):
In my mind, actually, it sometimes seems like it should be that simple. There's two things you said, I think, that are so key right there. One is it's just the ability to see. And I think, Scott, like you, my experience has been the same. The further that we get away from the team, the team has a sense of its own capacity. The team has a sense of its overload. The team has a sense when scope or priorities keep changing on it, how much it's thrashing. And somehow in manufacturing, and my experience of this was not the print shop. Where it really hit me over the head is being on the catwalk, and every large manufacturing line has this catwalk where executives get to look down. They get to see how work is flowing. They get to see.

Mik Kersten (11:21):
And I think one of the most interesting stories that was told to me at that time was how important it was, in some cases, for executives to see the re-work area. The re-work area is where you see quality problems compounding. And basically, if too many cars are there, there's something wrong. Action needs to be taken that hasn't been taken. The end on the cords didn't do enough at that point. There's a systemic problem that we have with quality because too many cars are piling up, and we see them piling up. So-

Scott Prugh (11:46):
It's hard to see that in virtualized software.

Mik Kersten (11:49):
Yeah, that's right. And in the end, we need this catwalk, the raised platform where executives or the stakeholders can actually get that real view of capacity. Which, I guess, brings me to what you've done, which I really want to dig into is taken so much what you've done with teams, both on the operations and infrastructure and services side, and the development side. And I just really love this work on DevOps portfolio management, how we elevate the things that you've elevated, the things that we've learned that work at that team level, to the portfolio level, and basically in these three different domains. So can you, just at a high level, take us through that? I'd love to dig into a couple of these things, because I think some of the answers of what that catwalk looks like are in what you've got here.

Scott Prugh (12:38):
Yeah. So, I mean, just credit where credit's due, I mean really, when we talk about DevOps portfolio management in three domains, it really ties back to the Accelerate work that Nicole Forsgren and team worked on for years. It really is a way that we can talk about that in our organization, and to other organizations, of what's important at the portfolio level, and at the team levels. And I like to think of the ... I've got a kind of nine box of the three domains. I really like to think of that as kind of a CIO, CTO kind of level discussion slide where if I'm talking to a client, or I'm talking to leaders in other organizations, I can talk about kind of these nine areas, three domains, and then nine areas that are important.

Scott Prugh (13:25):
And then we can kind of dive down into the other areas. And really, there ends up being 38 high performance capabilities, 24 were in Accelerate, and the others were scattered through the state of DevOps reports. And then we can dig into those other areas and really kind of talk about really the performance levels and the maturity at those other areas and where we would actually want to focus improvements.

Mik Kersten (13:48):

So take us. Where should we start? We've got your three domains are technical practices, work management, and people and culture.

Scott Prugh (13:55):
Well, I mean, I think the other thing to kind of tell you a little bit more context on why I structured it like that and talk about DevOps like this is everyone asks you the question, "What is DevOps?" Or I get a lot of calls from our sales folks and our consulting folks, and they say, "Hey, we've got a client that wants DevOps."

Mik Kersten (14:15):
They want to buy DevOps.

Scott Prugh (14:16):
They want to buy DevOps. I mean, it's the joke right? And it's the joke because they definitely want DevOps, and the thing is it's really hard to buy. And we can help you, and it's not just a tool. So this is really, the portfolio management, the DevOps portfolio management, and the three domains is my way to talk about the different things that DevOps is and brings together.

Scott Prugh (14:40):
So the first is the technical practices, which is really what people go to when they think about DevOps. It's things like the CICD. It's really the stuff from Jez Humble's book on continuous delivery. It's architecture and security. Architecture is really important in DevOps, being able to refactor code, making code operable with architecture, things like feature switches. And then it's observability, and that's really kind of taking production telemetry and feeding that back to the teams.

Scott Prugh (15:09):
So those are really kind of the three boxes of the technical practices. Then the work management, and Mik, this is your space that you love. This is really the stuff that is managing really in value streams and a portfolio, so not doing kind of that project based work. Having lean work management practices. We talked about that in really the print shop case. And holistic visibility, getting that holistic visibility of the work.
Scott Prugh (15:36):
And the final box there is I call kata and measurement. You see measurement all through the state of DevOps report, and you see that in the high performance capabilities. I put kata in there because I think that really what Toyota did with creating lean work and lean standardized work, but creating an environment where they were coaching and teaching people to improve every day was really a key thing that I consider kind of part of those work management practices.

Scott Prugh (16:04):
And the final and just so important domain is just people and culture. You see this a lot in Accelerate. You see it a lot in the state of DevOps report how important this is. And you can be great at technical practices. You can be great at work management. And if you're not great at the people and culture stuff, you'll never reach those higher levels of performance. Honestly, I've seen portfolios do fantastic on technical practices, decent on work management, but horrible on people and culture. And they have horrible performance. And it all makes sense. If your people aren't happy and you don't have a great culture, you're not going to reach those higher levels. And for us, these are things like cross-functional, long-lived, and empowered teams.

Scott Prugh (16:48):

We call service ownership and leadership, which is really how we kind of think about owning really kind of the end-to-end development operations components, and then leaders and managers who can actually do those things. And the final is the culture learning and psychological safety, which is just so important to have. So that's the three domains of the nine boxes. And I can give you a slide of that and you can attach it to the podcast.

Mik Kersten (17:14):
Yeah, absolutely. That would be great. We'll put that in the additional materials. And before we dig into any of these boxes, what really appeals to me about this is how these things, you need to think about them independently, these three domains, but how they're all required and they reinforce each other. So you gave the example where people and culture were just not there, the other practices were. I actually know multiple examples where there as a lot of great focus on people and culture, but there's a lack of technical practices. So no matter how much you focus on that, if people, they have to wait 6 to 12 months to see any of the impact of their work, well guess what? They're not overly motivated.

Scott Prugh (17:49):
And I think that's important when it all come together. And one of the current themes out there now that Ron Westrum just talked about is the concept of technical maestros and leaders who are experts. And this was also discussed in David Silverman's work, and then also in Admiral Richardson's, is competence as a leader also includes those technical capabilities, and are the technical practices. And it's so important because yeah, I have seen the other one where you have a great culture, but not great work management, not great technical practices, and you don't get great results there either. So it really is vital to bring those three things together.

Scott Prugh (18:32):
And the data in Accelerate shows us that. It shows us that you need all of those things. You need transformational leadership, you need leaders that have a vision and can inspire, and they also have to be great at their craft. And that's what the technical practices are really saying.

Mik Kersten (18:46):
Exactly. If these things are not in place, with the best culture, you can feel like the system's working against you. But then, of course, if there's just focus on anyone of these boxes, it's insufficient. Again, I think we've both seen it. Lots of organizations only focus on technical practices. But everything happening upstream, all the work being passed to them, the way that the business is interacting with technology is fundamentally flawed because the work management layer isn't there. So again, I think this is just a great, holistic view. And why don't you just pick a box, Scott, and take us on a journey through this, because I think, again, having a balanced view on these nine panes and these three areas, I think, is so key.

Scott Prugh (19:24):
Oh, geez. Where do you start? They're all so important. That's the hard thing. I think we'll probably ... Well, I was going to say we'll skip the technical practices, but I do want to hit on one thing there that's kind of important. The only reason to skip the technical practices is because that's what most people associate with DevOps. I do think the concept of the architecture and really how important that is to operability is something I often see missed. And so you can be great at CICD, and actually I've seen this occur, where you're really great at building the code, and getting it in the container, and delivering that container somewhere into production. But it's still a kind of mess to run.

Scott Prugh (20:08):

And so that's where these components of architecture are really kind of so vital. And the architecture is not just a software architecture thing, it's an operability. The best canonical example, I would say, here is feature flags. That's an architectural construct, but it also is an architectural construct that has a tie to the operability in production. And it really gets the development team involved in the discussion and understanding of how is this feature that we are working on, how is it going to get work in production? And how is going to get turned on? And if we just give it to the ops folks to turn this thing on, and it corrupts a whole bunch of data, how do we roll it back?

Scott Prugh (20:50):
And oftentimes, if you just focused on getting your code in a container, or on a VM, you're not really having those discussions. So those components of architecture and feature flags, I find, is a really ... One, it's a decoupling concept, but also it's a concept that really kind of brings that operational domain into the team. And really, even if they don't have operations engineers on the team, which they should, but even if they don't, now they have to really kind of think through how is this going to be activated in production? And it's a very powerful thing. So that's the one I would hit on, because it's something that comes up in a ton of conversations for me a lot, and is also something that even if you're great at the CICD, that is kind of often missed.

Scott Prugh (21:32):
So the next box, I think I go to probably maybe one of your favorites, or at least one of the ones that you hit on a lot is the concept of lean work management and holistic visibility. This is one that is really hard to kind of get people sometimes to understand. It really can be kind of a game-changer. And if you don't really focus on this, you really kind of just create a lot of wait time and waste on the portfolio.

Scott Prugh (22:02):
When I talk about holistic visibility, it really means all of the work. I think this is something we see, especially in organizations that are siloed between different roles. Like if you have an engineering organization and an operations organization, that the work is stored kind of in different systems, and/or like different projects or somewhere else, and people are working in their silo of work, and they're doing a good job at that, but they're not aware of the other silos of work. Incidents are one we see a lot. I've got the ops team working on incidents, and I've got the feature team working on features. And I don't have holistic visibility of those same things.

Scott Prugh (22:44):
And we've got examples, and we've seen them where the customer's like, "I want this feature done, but I also want this incident fixed." But those two request go to different organizations, and you've got the customer that's unhappy because they have both an incident and feature, and neither of them are getting done because then you have the team that's fielding the incident eventually getting back to the team who wrote the code, but they're working on the next feature that the client's asking for. And you've got this almost double-blind problem where we don't have that whole list of work, and we're not going and saying, "Maybe we should fix the incidents first before we start working on the next feature."

Scott Prugh (23:28):
And it's important to take that holistic list of work, across all those different work types, and also get your stakeholders involved. And the key stakeholders, usually the product manager, and to really bring them and make that list visible and surface to them and say, "Look, 80% of our backlog is filled with incidents. We have to fix those. We have to stop these other things." And to get them to be a stakeholder in that discussion as opposed to just creating more and more features that we can't finish, that the client actually, escalations are really around the incidents.

Scott Prugh (24:05):
And we've seen that occur a lot, this kind of double-blind. Or we've seen double, triple, quadruple blind scenarios where you have four lists that all different people are working on with different priorities. So I think that one's really important. Figure that out, you can have great results. And both operationally, customer satisfaction, and then additionally, your flow through the system improves significantly if you can figure that out.

Mik Kersten (24:31):
Well yeah, exactly. And the double-blind thing I think is exactly the right image. And it's everyone's trying. A set of people, or a set of stakeholders need these features. They might be salespeople, delivery people, someone of that sort who need these features for the client to expand, make them more successful, get the next deal. While of course the incidents are what's going to prevent the client actually being successful. So I love how you paint that, Scott, where we need that to be visible holistically, and we need the other things that are invisible. I think as you have and I have discussed at length, invisible to the client, to the customer, tech debt improvements, and tech debt reduction, we know those are not visible to the customer at all. And that's the challenge, where that's not visible to the business or the customer, but if we're not investing in it, then of course we know the incidents will come.

Mik Kersten (25:18):
So I guess how do you approach it because I think that the challenge here that I think a lot of us now understand we need to make ... Well, Dominica Degrandis told us all quite a while ago that we need to make all this work visible. And I think we've got ways of doing that. But in terms of the actual conversations, because I think what's so important that you're doing here is that you've got a way of elevating this to the business conversation, and to leadership within your organization and others. How do you approach it when it can just be difficult to convince someone that invest in technical debt and delaying those features is going to give them a better economic outcome for their customer, when of course, they're trying to deliver for their customer yesterday.

Scott Prugh (25:58):
Yeah, well, I'll tell you it's hard, and that I think that's what I said when I said, well, how we got here, there's a series of reasons for that. But it's how do you get out is really, really the hard part. It's a couple things. And I mentioned the product management relationship. And I'm an engineering and operations leader. I take a vested interest in building that relationship with the product management team, and continually having the discussion around the importance of one, seeing all the work and having them understand what that does to the teams, and especially getting them on the page list for incidents. So the product managers, they get the major incident pages. It goes out. I'm like, "Hey, you're on the page list." They don't have to wake up and get on the call, but-

Mik Kersten (26:51):
I like that. That's a good tip.

Scott Prugh (26:53):
Yeah, that's what we do. And at first, there was a lot of pushback. And then, after awhile, I start getting questions like, "Was that a big one? What happened?" And so there's things like that, but also sharing in the goals. We have ... I mention in some of my presentations this concept of an impact minute. Think of an impact minute as the size of an outage and failure group size, like how many customers you impacted, times the number of minutes. That's the simplest way to think about it. And there's some subtleties, but it

really gives you kind of the size. And we have impact minute reductions that we continue to work to reduce those. We have those on our goals.

Scott Prugh (27:31):
Years ago, I went to my product management peers, they're my peers, and I said, "Hey, I'd really like you to share in these goals." Their commentary to me was like, "Why should we share in those goals? We can't affect those." And I'm like, "You're wrong. You control the investment for the product. I want you to basically share these goals with me and my engineering and operations teams. And then we're going to look at the whole backlog of all the things, the incidents, and we're going to make decisions on where we want to invest to basically accomplish shared goals of reducing the impact to our customers on the system."

Scott Prugh (28:07):
That was a little bit eye-opening. One, the discussion was tough and eye-opening. But then, getting them as stakeholders in those measurements on what this meant to our customers was a key kind of galvanizing point too. And then I really think the one other thing I work a lot with on that relationship, but also just really continually elevating and making visible the work, the tech debt work, and the reason why it's so important. And some of that's just a discussion. Like we're doing a lot of work now to improve developer productivity and simplify the technologies, and really getting examples of like, "Hey, when we change this, it takes us two weeks because this technology's really difficult."

Scott Prugh (28:52):
So if we change out that technology, which may take us two or three months, then these types of changes, and we do a lot of those types of changes, we can do those now really easily in a day or two, because the technology's a lot easier for us. And we have a lot more skill. It's something like we might have boutique technologies that only certain folks can change, and being on more standard technologies, we have more people that can do that work. That's another example.

Scott Prugh (29:16):
So we continually kind of take that list and make sure that the visibility to that is there with those product stakeholders. That's another technique. So to summarize, building those relationships, sharing operational goals, and that whole list of work, and then continuing to elevate the technical debt and the benefits that that brings. And they want stuff faster and cheaper most of the time, and really kind of talking to them in that language is very important.

Mik Kersten (29:43):
Yeah. That's exactly what I've noticed that you do so effectively is talking with that economic language. But I think it's that trick of actually having them see and feel incidents, and wonder how much of a problem that is, or what the impact minute of it would be. I think that's also really interesting, is that just to get them involved, to get them looking at the production line, and looking at what ... Seeing what some of the team sees. But then, Scott, for the bigger picture things, so when you've got ... I guess my question is you're bringing them into the stream, which is awesome.

Mik Kersten (30:16):
But is there some cadence at which you reinforce these things during planning? What I'm leading to is I've noticed that organizations do strike a good balance between features, defects, risks, debts, between balancing work in the end for customer outcome, but not ignoring the things, or not over-investing in tech debt, when actually, this is a value stream that you need to get off of or strangle out. Do you have a

regular cadence at which you do this kind of planning? Is it a release planning cadence? I guess how do you approach that in terms of making some of the more forward looking things, changing team structures, changing how you look at managing services and the rest, how do you play the longer game?

Scott Prugh (30:55):
Yeah, so I mean we have quarterly cadence for planning, what I would say kind of the big rocks. So we use epics. So we use mechanisms in SAFe for portions of the planning. So we have PI planning every quarter, and that's when we have the whole list of epics and we intake that big stuff and really decide what the next quarter is going to look like. And that's when we look at ... We have basically an investment portfolio or PI we kind of look at and we say on one portfolio, 60% of the work may be client driven, 20% may be tech debt, and the other 20% may be team driven improvements, for example. Those may be how that portfolio's broken down. Or it might be 10% to security.

Scott Prugh (31:51):
And so those are generally set. Those percentages are generally set at that time too, depending on the priorities that we see. So maybe there is a lot more tech debt work, or maybe we have a big audit or compliance coming up, and we have to swell the security investment that quarter, so everyone's working on that. And so those epics are done on the quarterly.

Scott Prugh (32:14):
And then features, which are really what roll up in making the epics, we used to do those quarterly. And that was too ... Like try to decide all the features that you're going to do also, and pre-plan all the features that are not getting done on the next quarter. And we started to understand the problems with that. So with features, we actually then kind of go to weekly intake for features. We call it weekly grooming. And it's pretty powerful because now, you're not bottlenecking all of that feature planning and intake into one quarterly event. We're doing weekly intake.

Scott Prugh (32:49):
Like, here's a new feature that a client is talking about. And they're asking for this, probably in the next two months. And we try to leave spare capacity so we can even kind of get those in. And so that's one key thing. So we don't do big batch planning of features. We're doing the batch planning of the big epics. And we do change those if they need, but those are the big rocks.

Scott Prugh (33:13):
But the other thing we started doing the last couple years, and you'll see this in a couple presentations, is something that we call ROD, or release on-demand. And we used to do that quarterly planning, plan all the features, and then let's release them all on one day. So what we started to do in using some of the architectural constructs, and I mentioned why that box is important to kind of tie it to that, is we then start saying, "Well why don't we just start releasing those features when we finished them?" So plan big rocks quarterly, intake features weekly, and then release features when they're done had just an amazing effect on kind of flow in the system because now, I do the big planning because I still need to do vision and decide on priorities and investments and stuff like that.

Scott Prugh (34:00):
I can adjust granularity on a weekly basis on what my finer-grain priorities are. And then, I can then get that work out of the system as soon as I'm done with it. And then people can say, "Hey, let me pick up the next set of work." So it had really kind of amazing change to the portfolio when you get through all those changes to increase flow and increase how work flows through the system in, what do I say, a much more cadence-based, but also much more flow-based way.

Mik Kersten (34:31):
That's awesome. And I think just so many key things in there. And I think I completely agree with approach. It's our approach in value streams I'm involved with as well. And I think that you said something so important there at the very start of that, is that you're not only balancing that work, work around incidents and features and understanding those capacities. But I think you said 10% allocation for actually improving flow. And this is the other thing. The two things that have been amazing and disturbing to me in terms of actually seeing the data out there across organizations is just, again, the amount of flow load, the amount of WIP, the size of the backlogs, and backlogs and the fact that they're growing.

Mik Kersten (35:10):
But then the other thing that has been even more surprising to me is just how little capacity is allocated to improvement, to that ideal of improvement of daily work. So you're saying, and I think this is critical. I know we do this. For us, improving our flow efficiency is a top-level OKR, top-level objective for the entire organization. Because oftentimes, you'll have dependencies on other parts of the organization. You'll have dependencies, be it on legal, on infrastructure, in the end, the business models and the sales functions and these other things. So you deliberately allocate. And by the way, we do also check it, do it annually and quarterly as well, and dovetail it with our planning. You're doing that as part of your release plan and your PI planning.

Scott Prugh (35:57):
Yeah, so there's a couple things that we do. I guess I can elaborate a little bit more. So there's one on tech debt that we ... Those are kind of the big tech debt stuff. And those are generally planned and kind of intake. So we might be porting off a commercial database to opensource, when we set to public cloud. Those are things that are generally in this kind of bucket or tech debt or technical kind of work. I mean, they aren't necessarily client or product facing features. And we allocate a percentage to that, somewhere between 20 to 30, sometimes a higher percent.

Scott Prugh (36:36):
And then, we also have this bucket, and years ago we kind of came up, when we were going through some of these transitions and we were struggling, and kind of really looking at this concept of empowerment, but also the improvement of daily work, came up with a concept, not that it's rocket science, but we basically said, "Hey, what we're really talking about is helping people improve their work life balance. And let's take time and give it to the people to do what they want to make things better." In other words, we said like, "Here is 20% of time, and you figure out what you want to do with that." And what I said is I said, "The only thing I require is that you document what you did so that we can celebrate it and publish it."

Scott Prugh (37:27):
And so at the end of a PI, or we get to the next PI planning, we can say, "Look, here are all the things that the teams did to make their environments better." And so I mean it did a couple things. One is it got the concept of daily improvement, and it gave space for people to do that, but it also built the empathy between leadership and folks that we truly cared, and we did, and we kind of put our money where our mouth was, which was like, "Look, we are giving you this time and money to make your lives better." And stood hand-in-hand with product management on that too to say like, "This is important."

Scott Prugh (38:07):
This isn't something you business case and say, "What return am I going to get?" This is something that is vital to improving employee morale, improving people's lives, all of that stuff. And we really, we truly ... I

truly believe that. And I need my product managers to be in the same place with that too so that we can make that investment.

Mik Kersten (38:27):
Yeah. I think that's just an amazing approach. One thing that's really ... We've had something similar with... We do our quarterly planning. We have this jog week where there's a week that's dedicated to experimentation. And in the end, very similar to what you're doing. And we've made it maybe just a little bit more specific where it's about experiments to improve flow. That can be finding a problem with the infrastructure that's causing all these incidents, and creating a little automation, fixing something in a test suite, or just looking at a bottleneck externally. It's been interesting. Some of these experiments, they do span outside of engineering.

Mik Kersten (39:05):
It's a little bit harder when it's limited to a week. And that's why I love your basically much more continuous approach to this where we've realized we actually, across the release cycles, we want teams to continually experiment with identifying their own bottlenecks and be running multiple experiments and parallels. Sometimes you're wrong. Sometimes you think this is the source of your bottleneck, but until you actually try resolve it, you don't quite have the right hypothesis for there. And you experiment, and then you move on. So the fact that you've actually baked this into daily work I think is critical. And I think it's just so important other organizations look to what you've done and apply the same thing. Because so often, these teams have no capacity for work. And then of course, the culture problems and the burnout and everything triggers because they're not actually able to get all those wait states out of their way. And those things, fundamentally, are just frustrating and a death knell to culture.

Scott Prugh (40:01):
Yeah, I mean, bottom line is when you cap people to 100%, they don't have time to improve. So they're just going to take the next feature off their backlog and start grinding that out. When we create space for them, and also set the expectation that they're empowered to make things better, amazing things happen. The stuff that you would never think of. They come up with, I'm like, "Really? That was a problem?" And it's fascinating to see that creativity for making stuff better just kind of get unleashed at scale.

Mik Kersten (40:37):
Yeah, and the customer gets more in the long run as well.

Scott Prugh (40:41):
Yeah, yeah, yeah. Exactly.

Mik Kersten (40:41):
Well, Scott, I've think we've made it through, what? Two boxes.

Scott Prugh (40:45):
Yeah, yeah, yeah.

Mik Kersten (40:45):
Of the nine. Maybe let's pick one more.

Scott Prugh (40:49):
Okay. One of the ones that's pretty important is this concept of service ownership and leadership. And there's two things in that box. And they're a bit related, but they do kind of cover two things that are vital.

And when we went through our transition to really create cross-functional teams that really kind of build and run to give the ... What is often said, we came to the conclusion that it's not necessarily only a team thing. We need leaders that can really own the service.

Scott Prugh (41:27):
And we call that role a service owner. And we needed really very strong technical leaders that could say, "Hey, I can lead a set of teams that both understand how to engineer products, and to operate those products." And so we really kind of talked about that in this term of service ownership. And we call these individuals service owners. It was our way to really kind of put a name to those characteristics that we wanted. And I think now, hearing some of Ron Westrum's stuff when he talks about a technical maestro. And we also talk about in Admiral Richardson's discussion he talks about leaders have to have competence. It was really our phrase to attach to those things, to really say we needed these service owners that could really embody both the engineering and operational capabilities.

Scott Prugh (42:21):
And of course, the leadership capabilities. Those are all the other things that you see in the DORA research around transformational leadership. Can you lead? Do you have inspirational communication, a vision? Do you know where you're going and can you talk about that for the next five years for your teams? Those are all things that are really embodied in that.

Mik Kersten (42:41):
Excellent. And so question though, because you've got this ... There's the value stream management box, which we've kind of edged in on. How do you actually ... I think this a interesting question. I know it's one that my thinking on it's evolved a lot. How do you actually combine that service ownership with, in the end, these product value streams? Of course, some of the product being these internal platforms and operational services and so on. So how has your thinking on that front evolved? Because I agree, they both are key.

Scott Prugh (43:09):
Yeah, so the value stream of portfolio management really kind of captures, I think, two key things. One, are you aligning the work for value delivery. So it's really right out of your book, Mik. And it's saying you've got a value stream. You don't have these siloed handoffs all the way across the org. You basically have organized the value around a product, and you're flowing the work through there. And then, you're investing in that value stream.

Scott Prugh (43:40):
Portfolio management piece captures kind of then the component of you're doing that across your portfolio, but also you're treating this portfolio like a set of products, and you are investing in this product and portfolio management capability. So in other words, you're investing in product management leadership, which is so vital. And this is where software engineering firms, I think, really have a leg up on traditional IT firms because they have a heritage of really strong product management. And that's so important because the product managers are really the ones that are taking the market-facing requirements. They're working with the internal teams on, of course, "Hey, we got technical debt. Need fixed."

Scott Prugh (44:22):
They're trading off those kind of priorities. So kind of how that works is really we look at our service owners as really the partners to those product managers on the other side. And so you've got the product managers who are managing the investments. And we got a portfolio of these product managers across

the portfolio. They're working with those service owners that own the engineering and operations of that service to basically store that investment and really kind of decide how to get the optimal value out of that product and set of services.

Mik Kersten (44:58):
Yeah. And I think, exactly. It's making sure that value streams, and tell me if I'm overstating it or you think about it slightly differently, but in the end, for each value stream that runs as a service, the service ownership, all of the operations, is part of that value stream?

Scott Prugh (45:11):
Correct.

Mik Kersten (45:12):
It's not this handoff. You're responsible.

Scott Prugh (45:14):
Yes. And of course there's always exceptions. Like you'll have team topologies. You've always got these things where you have dependencies on other teams that kind of might be their own value stream, but they're tied to a different value stream. So there's always exceptions. You're never going to get this perfect, like everything just lines up in one perfect. But we really strive for 80 to 90% of it to be aligned that way. And to then also have a product manager that lines up with that value stream that's the steward of that investment. And we even treat our platforms that way. We have a product manager of the platform value stream. The platform value stream is the infrastructure. And then also, product manager of security. And that investment is across all value streams, so that one's not perfectly lined up because we're making cross-product investments in security.

Mik Kersten (46:11):
Absolutely. But I think it's this kind of thinking, and even though, like you said, it's never aligned, it's never perfect. But it's, bringing back to the start of one of the points you made, Scott, that's what causes the right architectural decisions to be made, where value stream is architected for easier operations and observability. And you make the right kinds of decisions, like implementing feature flags rather than forking the code base, or doing other crazy things. So yeah, I think that's so key to this approach is having the value stream alignment and including service ownership as part of that actually drives the architecture, and I think some of the points we've talked about less here, the team structure and ownership in the right direction.

Mik Kersten (46:54):
So we're at time here. But there's probably another three podcasts worth of content that you've outlined here, Scott. We will absolutely point people to the work and the resources. But anything else you want to leave our listeners with in terms of heading in this direction that you've been on for years now and have come to such a good spot on?

Scott Prugh (47:14):
Well, a couple things. I appreciate the discussion. I always learn something in this and always come away, I'm like there's certain tweaks I should make in both the way I talk about this stuff and maybe the slides. So you might see a few changes. That being said, I want to emphasize my gratitude for this. And then also, to the listeners, you can find me on Twitter at @ScottPrugh. You can DM me if you have any questions. You can also find me on LinkedIn. But DevOps, it's really important. It can really transform your organizations if you really focus on improving the technical practices, work management. And also

very important, the people and culture. Those are really the three domains that make up DevOps. And improving all three of those things can really move your performance up a level into the high performance category.

Mik Kersten (48:03):
Awesome.

Scott Prugh (48:04):
Thank you, Mik.

Mik Kersten (48:04):
Yeah, no. And exactly. I the whole point is that you're able to elevate these benefits with the organization. I think this is a great framework for thinking about that. So thank you so much, Scott. See you again soon.

Mik Kersten (48:20):
A huge thank you to Scott for joining me on this episode. For more, follow me and my journey on LinkedIn, Twitter, or using the hashtag #MikPlusOne or #ProjectToProduct. You can reach out to Scott on Twitter, @ScottPrugh, or via LinkedIn. I have a new episode every two weeks. So hit subscribe to join us again. You can also search for Project to Product to get our book. And remember that all author proceeds go to support women and minorities in technology. Thanks. Stay safe and until next time.