The Modern Mainframe Podcast – Episode Two Transcript

THE FLOW FRAMEWORK WITH DR. MIK KERSTEN

Mike: Hey, everyone, this is Mike, and I'm excited to be speaking with Dr. Mik Kirsten today. He's CEO of Tasktop and the author of a recent book, *Project to Product*. Mik, thanks a ton for joining me.

Dr. Kersten: My pleasure.

Visiting Compuware in Detroit

Mike: We recently had the pleasure of hosting you and Gene Kim here in Detroit for the day. In speaking with our CEO, Chris O'Malley, I know there was a lot of reciprocal education occurring between you three that day on, and I'm wondering if there's anything that you've been reflecting on since that visit that you might want to share.

Dr. Kersten: Yeah, absolutely. I think that what Chris and the Compuware team did was absolutely eye-opening, both in terms of the transformation that's been happening over the past few years at Compuware—and how profound and interesting that is in how Chris and Team have led that—but also what it means to all those large organizations, many of whom are our customers at Tasktop as well, in terms of how their data center, their software stacks, how their mainframe investments are going to change over the course of the next year or two.

I think the one most profound thing that I took away from it, and that I've shared with many people since that visit, is the fact that we actually got this very visual view into what the future of these organizations' data centers and IT stacks looks like. And it's very different than what it is today. But Compuware has actually taken that step already and is actually sitting there already to be viewed by anyone who's interested. So, to me, that was just fascinating.

Compuware's Two-platform IT Strategy

Mike: Yeah, and with our data center, moving to basically just a mainframe-and-cloud strategy, I think save for 12 racks of servers left maybe, we call it Two-platform IT, and it really, like what you're describing, has been a major component of Compuware's transformation. And something that we see, and I think something you've pointed out too in your talk at DevOps Enterprise Summit, is companies tend to focus a lot on the wrong things in transformation, and it actually can cause more damage than they're already experiencing.

You created this thing called the Flow Framework. And from what I understand, it helps with focusing and bringing things together. But can you explain what that is and how it kind of improves an organization's ability to transform? What exactly is it?

Dr. Kersten: Yeah. Absolutely. And I'll just start that explanation by just recounting of it from my perspective of what I saw in Compuware's data center because I think it speaks volumes in terms of what the Flow Framework is trying to achieve. And my recent book, *Project to Product*, is really a vehicle for it.

So just to give you give our listeners some kind of sense for this, Chris took Gene Kim and I into this data center. And the funny thing was Gene and I, right before we were having breakfast at a Starbucks and were wondering, "Why is this on the agenda? Why would anyone want to take the two of us into a data center?" And I thought someone explained it to him. He thought maybe I'd got more detail on it. But we were pretty sure Chris wasn't going to waste our time, but we had no idea what we would see.

I was just hoping to see a whole bunch of interesting hardware at the least. But we walked in and we saw the Systems z running just massive and high-transaction loads of COBOL and then behind them—which by the

way to me are still super impressive the way those racks look, the amount of hardware and dollars in that dense a space, and this was not a huge data center, but it was not small by any means—we saw this graveyard right, this graveyard with these tombstones and what looked like these police outlines of the dead bodies that were actually racks.

I thought the most brilliant thing was that in each of those outlines of the dead racks, we saw the name of the system that was retired—some internal HR system, some other IT system—and we saw the dollar savings from what was retired. I, as someone who works with enterprise IT leaders a lot, have just always listened to these discussions of organizations, of CIOs wanting to move into the cloud, wondering how quickly they should do it and so on, and I've always been a part of discussions. And then I walk into Compuware's data center and I see, "Wow, this is done. The move is actually done."

Compuware has done this incredible thing of actually creating this Two-platform IT, where the core workloads, the core differentiators, the stuff around Mainframe and COBOL, is running in the data center, and everything else, all the x86 workloads, that are much better off running on Amazon or from some SaaS provider, is actually gone. The transformation is done, and that really just set the theme for the day. Because how do you take this bold an approach of burning the boats, of moving things that quickly, and roll that into an Agile transformation, a DevOps transformation, and actually see the results of being able to take those cost savings from doing all that work on-premises and funnel them back into software product delivery yourself or value streams to help you become more innovative?

So, to me, this was just fascinating that this project I've been trying to get organizations to think about had actually happened at Compuware. And there was this, again, very visible evidence of it happening. And of course, this data center graveyard is just one part of the story. That's such a visible and visceral part of the story. I think it's really important for leaders to understand what happened there.

One thing that Chris emphasized with me that really made an impression is this approach of burning the boats, of actually taking that step towards Agile, towards DevOps, towards cloud is critical, because the risk of keeping the current trajectory or not moving quickly enough is actually larger. When you can't redeploy all of those resources, whether they're cost resources, staffing resources, into your own software innovation, you're going to get left behind.

The Flow Framework

Mike: What you're describing is sort of what the Flow Framework is right? Is being able to bring all of those things—costs, DevOps, Agile, all these processes, your tools—together. I think you've also talked about, if

you're using SAFe for instance, and how do you have these different streams and processes going on? How do you get them all synchronized? Is that correct?

Dr. Kersten: Yeah, at its highest level, the Flow Framework is meant to capture what was already in Chris and his team's heads and make it easy for others to apply to the businesses. Because Chris knew that all these x86 workloads and all the investment going into these other applications that were running the data center was not core to Compuware. It was not the kind of innovation that Compuware customers had been yearning for and not getting for a very long, time to be frank. Until that shift happened, what, four, five years ago?

So really the Flow Framework is that way of looking at all of your product value streams and determining what value they're delivering to the customer, what the cost of each value stream is and what the flow metrics for each one are. Basically, how much business value you're delivering and where.

By the way, it's okay that for a very large organization—now I'm shifting out of Compuware into, let's say, a large enterprise IT organization—to have hundreds or thousands of different applications and software products in the portfolio. It's okay. Start looking at those; ask, "What's this costing me? Is this core to me? If I actually reduce the cost of this thing, could I redeploy that somewhere else?" And then for the things that should really drive your growth, your customer satisfaction, you're basing your place in the future as more and more of the market gets disrupted, that's where you probably want to put that investment, in things that will drive your top line.

You really need a separate way of understanding the value streams there to protect your bottom line, really maybe moving those very aggressively onto a lower cost platform such as a cloud provider, and again taking those savings and putting that into product value streams. They're going to drive your success and your customers' success.

And that's really what the Flow Framework is all about, is giving both the business side and the technologists a common language to achieve that. A common way of describing product value streams, describing what flows through the value streams, how you deliver business value. A way of being able to discuss together the flow distribution. For a particular product, how much you invest in, let's say, risk reduction such as security or data privacy versus innovations such as feature delivery for customers.

So, the Flow Framework is really meant to be this language that makes these kinds of decisions easier and more empirical. It gives organizations the data to make these decisions, to know which of, let's say, three hundred value streams to invest in, which ones to sunset, which ones to lift into the cloud and so on.

But what was so amazing about the Compuware visit is that's exactly the kind of thinking that we see evidence of at Compuware, and the Flow Framework is obviously intended to make that kind of thinking, that kind of business decision-making around software values streams much easier for anyone working with very large software portfolios to make.

Value Stream Mapping

Mike: Okay, and you kind of got into this idea also that I've been hearing a lot, especially from Gene Kim at DevOps Enterprise Summit in Las Vegas, of bridging that that gap between the technologists and the business leaders. It sounds like the Flow Framework is a way to codify that and make it understandable across these different players.

But you also talked about value streams, and at Tasktop, what you do is value stream mapping. Is that sort of a separate concept—or, not separate, but how would you describe it? How does it play into the Flow Framework? What is that concept, and why is it crucial to transformation? You started a company around it, so it's obviously something that you think is critical.

Dr. Kersten: Yeah, we see two critical parts of these transformations, and really looking outside of tech companies. Where within Compuware the technology stack is actually extremely elegant through this transformation and through the fact that Compuware has been able to both leverage the COBOL tools and move to some of the new tools and add some of its own tools such as zAdviser. You've actually got the thing that, or at least largely got exactly what, we're after for our customers, which is a connective value stream.

But the problem is, if you go into organizations that are one or two or three levels of scale above Compuware and of legacy even beyond the company, where by virtue of having been a traditional business—a bank, a health care company—rather than a tech company, we see something very different and very problematic in these organizations' value streams, and that's the fact that they're completely disconnected.

So, these organizations might have acquired a dozen different companies over the course of the last few years or decades. They'll have completely different sets of Agile tools, right? The Microsoft work is being done on the Azure DevOps stack, and the Java and web work is being done on Jira, maybe there's some Cloud Platform work bring tracked with Google's Issue Tracker. Then there's the service desk, which is from a completely different vendor, and then all these product management tools and so on.

What we've learned in trying to bring these companies to have this kind of product development flow is that the biggest bomb in their flow is that their value streams are disconnected, and the business has no idea what the value streams are beyond what someone's reporting in a project management tool. So, there are kind of two key problems that we see as the biggest impediments to these transformations:

1. **Disconnected Value Streams Don't Work**

   You can't achieve the kind of success you want with your Agile or DevOps transformation if you've got disconnected value streams, because those are your impediments to the ways of DevOps, of flow and feedback and continuous learning. So you have to know what's flowing where, and you have to make sure that you have automated handoffs between someone finding requirements for and application and someone actually doing the Agile development and deploying the software.

   So you need to connect all these different working intake systems rather than, again, having all these things be manual processes and disconnected between all the different specialists. So that's the one problem and that gets in the way of all the people building the software. That's what slows them down by an order of magnitude there, as compared to their startup or tech-giant counterparts or the likes of Compuware.

2. **The Project Management Lens Won't Work**

   So you've got this big disconnect. The second problem is that the lens that the business looks at that software deliver with is completely the wrong lens. They're using project management, which is great for building buildings and tearing down data centers, but it's terrible for highly innovative and highly uncertain work, which is exactly what software delivery is.

   So, the business is completely misunderstanding what's happening on the IT side because they're using the wrong tool, and really, the wrong management framework of project management, which is not what software innovators use.

How Tasktop Helps

So, we realized we need to help organizations with these two things. We need to help them connect their value stream network, and that's what we focus on so much in terms of our offerings, and that's really what Tasktop Integration Hub does. It allows you to connect what we currently support, nearly sixty different tools, because there's such a broad diversity of tools and specializations in Agile and DevOps. So, you make those tools work together by having, say, a feature flow from one cell to another so everyone can collaborate, and even if one person's using a service desk like ServiceNow, another person using a Microsoft tool and so on.

Then the other key thing is we enable these organizations to structure their value streams to align around products because software innovators focus on product delivery, not project management, right? You should focus on how much value you delivered with your software product, what the cost of the product is, where you invested in this product portfolio.

So, the Flow Framework really fills that gap between what we now call the tool network—that's all the tools that you've got in house or in the cloud—and we allow you to connect that to be a value stream network where

everything flows along these product value streams. And then the really neat thing is that's where the Flow Framework comes in.

Flow Metrics

Once things are connected you can actually get these flow metrics, like what your end-to-end flow time is from the moment a customer requested something to the moment it was delivered and running, whether it's running in your data center or it's running in your cloud, but the software's actually running. Because this is really the key thing we want to get away from it, which we have customers report once they start measuring using our tools, using Tasktop: where time is spent for a feature to get delivered to a customer.

And this is going to sound crazy, but at some organizations who think they've actually gone Agile and are doing DevOps—which means that the development and the deployment and operations teams are actually delivering fairly quickly (so let's say that they're able to do a feature in a two-week sprint)—they're taking 80, 90, 120 days to get a feature from a customer request or from the business approving it, to market.

And one of our customers reported that it was two and a half percent off the total flow time—and this is one of the core measures in the Flow Framework, the point from when you started working to when your customer saw value from the work you're doing—only two and a half percent of the flow time was spent on development.

Making Software Delivery Bottlenecks Visible

Mike: Oh, that's crazy. I mean, because you would assume if you're innovating, you want the majority of your effort obviously going towards that work. So, where is the majority of the effort actually being spent, technical debt and things like that?

Dr. Kersten: Well, that's not even counting the technical debt. This is just the raw time for the thing to get worked on. And so, the answer is the bottlenecks always tend to be upstream or downstream of development. So, in this case, it was a lot.

Downstream is when you have change approval boards and you don't have enough DevOps automation, release automation and so on. So, organizations are getting better and better at that, but still have some work to do. But what's fascinating is that in their efforts to do that, it's without looking at the flow time. You're just thinking, "Okay, let's do DevOps." You're ignoring the stuff that's upstream.

So, in this organization's case, it was actually the iteration with the business. Everyone's trying to make first rate digital experiences for the customers. Well, guess what? That takes work, right? You need UX people to do that. And if you have this preconceived notion, like so many of these organizations have, that "I'll hire more developers and get more software out more quickly," as many CIOs or Chief Digital Officers think, it can be pretty shocking when you hire more developers and you've only made that two and a half percent faster.

The fact that all the developers are waiting on wireframes from the UX people becomes a real bottleneck. And this is just one example; another company, it could be very different bottleneck that's upstream, so it could be related to regulation, could be all sorts of things on the business side, on the design side, on the same side.

So, the key thing is that, with the Flow Framework, you actually measure what's flowing through your value stream network. And then we can make those investment decisions and make the kinds of decisions that Chris was able to make, like, "We're not gaining any value from these x86 workloads."

The Mainframe Within Value Streams

Mike: My last question is, and I know you've touched on this a lot throughout our conversation already, but maybe to help our customers in particular at Compuware understand this, how do you see mainframe applications and processes fitting into a value stream?

Because I think a lot of times, mainframe people, mainframe teams, might have this view—and maybe it's not even them, it might be that their CIO or business leaders have this view—that the mainframe isn't modern

enough fit into these newer concepts that are actually very necessary for connecting technology and business and delivering innovation for customers.

So how would you explain that to a business leader or someone in a mainframe organization, how mainframe applications and processes can actually fit into a value stream?

Dr. Kersten: That's been a part of my conversations a fair amount because, of course, most large organizations have some mainframe workloads and are trying to understand this. I can't tell you how many times I've had the question come up, "Well, should we just dockerize everything tomorrow and turn everything into microservices and throw all this old stuff out?" And to me, it's just completely the wrong outcome, but it's also, even worse than that, it's the wrong question.

So, again, if you think in terms of product value streams, you will have some product value streams running COBOL workloads. If you think in terms of the Flow Framework, you need to think about, "What flow do I want from those? Do I want to reduce risk in them, or do I want 100 new features of this product value stream in the next six months for customers?"

COBOL workloads, mainframe workloads, are so highly optimized, they're doing their job, chances are, in many cases, you just want to reduce risk, reduce cost on those and basically manage those as they are while investing, let's say, in a more modern native cloud solution.

So, what I see smarter organizations doing in terms of thinking about both their existing mainframe investments and the new systems of engagement they're creating, their new user experiences they're creating, is actually to realize that these things will coexist. That Two-platform IT path Compuware has taken is exactly the path I think is the best path towards innovation, because these things, again, are working, their cost and performance are highly tuned already, and you're far better off investing in modern web technologies and microservices around that core of your mainframe than you are trying to replace it or find your path to replacing it tomorrow.

So again, the path Compuware has taken, unsurprisingly I think, is the path many organizations will end up on. And a way to think about it at a business level—and this is the whole point of the framework—is to identify those product value streams.

By the way, product value streams, a bunch of them might be just for internal products. Transaction processing can be its own value stream for a certain part of the business. The key thing is that in the Flow Framework everything's a product. Some of those are internal products to you. An SDK is just another product. Tech companies think in this way, where they treat everything that's part of their core platforms, even the delivery pipeline they think of as a product. Because then you think the right way, you think about adding features, reducing technical debt and make smart decisions about what to invest in where.

So, yeah, I think we've had plenty of proof points, first of all, to answer that part of your question, that the mainframe can exist in an Agile and DevOps world. Compuware is a proof point. I've seen quite a few customers have managed to take some similar strides.

And then, of course, just pick the technologies to map the needs of your value stream. Don't think about making sweeping changes and dockerizing everything. Where the value stream fits into your investment is, of course, a context. Another way to look at it that I really like is that Geoffrey Moore Zone to Win approach where you have a quadrant of what you invest in and allocate accordingly.

But don't think that the mainframe can't work elegantly in an Agile and DevOps transformation when it's driving some of your core workloads, and of course, where you can do things like wrapping microservices and another kind of services around that. And then potentially doing things like, if there's one part of your mainframe portfolio that you need to move off of, you can actually then have an incremental strangler pattern take hold of that. But it's just interesting how many organizations see that as the thing that's holding them back, whereas it's not.

Mike: Awesome. That's a lot of great insight, Mik. I really appreciate you sharing it. If you want to learn more about what Mik and Tasktop are doing, read his book *Project to Product*, and I'll put a link to that in our show

notes as well as a link to Tasktop, so you can see what they're accomplishing there. Mik, thanks so much for sharing your insights today.

Dr. Kersten: My pleasure, Michael. Have a great day.