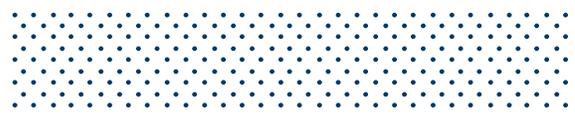


# Architectural Governance Pendulum



CAPTECH TRENDS | PODCAST | EPISODE 5 TRANSCRIPT



## **Vinnie Schoenfelder**

Hello and welcome to CapTech Trends, a place where we meet with thought leaders and subject matter experts to discuss emerging technology, design, and project methodology. I'm your host Vinnie Schoenfelder, Principal and Chief Technology Officer at CapTech Consulting. Today we're finally getting a little more technical and we're going to have some more technology-related podcasts in the near future. Today we're going to talk a lot about enterprise architecture. Historically, it's been heavier on governance across an entire organization and it's moved to be part of the actual functional delivery teams. And we're going to talk about where we think it actually should be, somewhere between those two things. What are some of the advantages and disadvantages of those two different extremes? I have with me today, Jack Cox and Matt Comer, who are both Fellows at CapTech as well as Bill Klos, a Technical Director at CapTech. Welcome.

## **Jack Cox**

Thanks. Good to be here.

## **Matt Comer**

Thank you. Hi, good to be here.

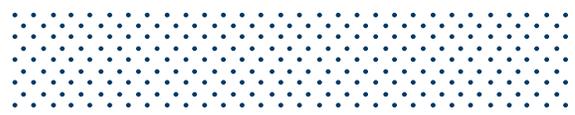
## **Bill Klos**

I have to say real quick, I always smile when somebody says, "Oh, you're on a technical project now," if I'm doing enterprise architecture, because a lot of times I don't really consider that a technical project.

## **Vinnie Schoenfelder**

Well, that's funny, so yeah we'll get into that. I'm going to restate the topic to go a little bit deeper, but then I'm going to come right back to it; I think it's important. So, from a topic perspective, you know, we've had a middle-tier technology shift from monolithic tools and frameworks to more lightweight, containerized, serverless environments. And we've seen a similar movement in enterprise architecture where a centralized EA has migrated to having a more distributed architects working within functional teams. And there was a reason for this, right? There was pain and dysfunction in the old way. But what we're finding out is that there's some pain and dysfunction in the new way as well, and we've traded one set of problems for another. And so, we believe that pendulum needs to be somewhere in the middle.

We'll talk about what those pain points are and what we're seeing from our clients. But I'm going to go right back to your point, Bill. And that is one of the things that CapTech has always



believed is that architects, even enterprise architects, keep their hands dirty and still write code and still contribute to that sort of development. So, I know there's some organizations that have more paper architects, but I guess in this discussion it's fair to talk about which side we're coming down on. So, I think we're talking about architecture where we keep our hands dirty. Is that what you're referring to?

### **Bill Klos**

Yes, there's a lot of video of Vizio architecture, that type of thing.

### **Matt Comer**

Well, but to be fair, an architect does have to think about things other than just the technology. And I think that's where it can sometimes get a little bit confusing, right? It can seem like you're doing a lot of nontechnical work, when in reality you're trying to figure what are the right set of tools and technologies and platforms that are going to this organization's needs. And some of the things that drive that are not necessarily purely technical, right? And so that's where it can start to feel a little soft sometimes, but it's important.

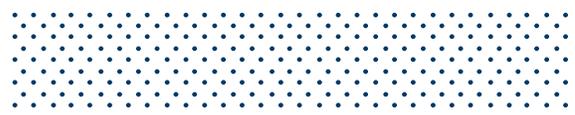
### **Jack Cox**

Yeah, I agree. It's important, but it should be a red flag to an organization. And when the application developers are going to the architects and explaining the technology to them, that's where you see hands-off architecture taking you into that situation where they don't know these new tools that the developers want to bring in and use, right? They don't know what they do, they don't know what the ramifications of them are.

### **Vinnie Schoenfelder**

So, most people learn through those discussions though. So, it's fully appropriate for our enterprise architects to listen to application architects or solution architects and become informed on those things. However, from a patterns and framework perspective, they should know where they fit into the larger enterprise design patterns and where they can be used and leveraged. So, there's a back and forth between it. But to your point, Jack, it shouldn't be educating on the basics and the fundamentals. It should be educating on what's new, what's coming out.

So, guys jumping into this, what are some of the things from sort of the old way that the enterprise architecture –we saw, you know, maybe five to 10 years ago – that people decided to move away from? Is it more change in methodology with Agile? Was it more of a toolkit change moving to the Cloud? Or was it a move because there was pain and dysfunction in the



way that those teams were affecting the delivery?

**Jack Cox**

I think the move was started by almost a shadow IT going on even within the IT organization. If we want to get things done quickly and we're constantly running up against friction with enterprise architect, how do we reduce that friction with enterprise architects? Why don't we bring them into the teams rather than having them be separate from the teams and try to reduce the friction that way?

**Matt Comer**

Well, I think it was too much of a focus on the G word, right? Governance. And although there's nothing wrong with governance over application of rules, again, if those rules aren't necessarily well grounded or aren't keeping up with the goals of the organization and the technology is being used etc., it just becomes an impediment. And then people start to figure out ways to work around those impediments. And then somebody gets smart and says, "you know, let's just get rid of these impediments," right? Let's stop walking around them, let's just tear them all down. And I think in some ways that's what's happened.

**Vinnie Schoenfelder**

And the documentation of those rules too. I mean, so much documentation that no one could even sift through.

**Jack Cox**

Yep. And it's never up to date.

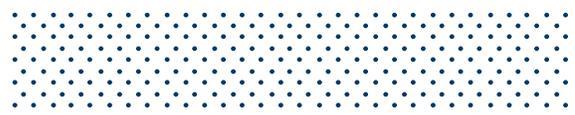
**Bill Klos**

I was going to say that the biggest casualty I've seen is the abolishment of the AR base, no more architectural review boards. And trying to replace them with quicker teams that make decisions and pivot better, but it's all as a result of what you guys already mentioned.

**Vinnie Schoenfelder**

Right, and sorry if we're stepping on each other a little bit. We're practicing physical distancing on these podcasts. Jack and I are colocated six to eight feet apart, but Matt and Bill are joining us over WebEx. So, we're doing the best we can with that.

So, another thing that I was thinking about with the way we used to do it was these canonical models. I remember I would go into organizations and there'd be an enterprise architect



function of trying to think of every possible API end point that the organization could need and building that canonical model so that there is a single way of thinking about the major attributes of an organization. And these are multi-month long or over a year or couple of years projects to try to do this, having the systems or having those APIs in place for when they're going to be needed. That seemed fraught with problems. And, moving towards this, we'll create the APIs just in time as we need them, maybe a sprint or two ahead of the functionality. Has that played a part in this migration as well?

### **Matt Comer**

Well, I think it's the same. It's, another description of the problem, right? And I think it highlights that one of the big challenges that we've tried to address with Agile delivery is that perfect is the enemy of good. And some of what we saw happening with a lot of that, canonical elevation of the data model and trying to imagine all of the services we may ever need and all the data they might need to return and etc., is that it's very, very hard to think about all of that stuff – to answer every question, to come up with a perfect data model. In fact, you're probably never going to do it, but the harder you try, the longer it'll take. And so, sometimes figuring out when the time is to say, "you know, this is good enough," is the key to making sure that doesn't become such an impediment.

### **Vinnie Schoenfelder**

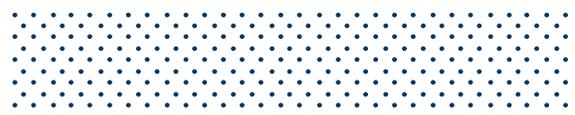
So, as we've moved to having the architects come into the delivery teams, what has stayed? Are organizations still have these large enterprise architect groups? Have they thinned out? Have their responsibilities changed? Have they gone away completely? What do you guys think at the clients?

### **Jack Cox**

I'm seeing that InfoSec has kind of spun off architecture. They're kind of taking the role that the ARB used to have of making sure that you're using tools that are the right tools, that they've been tested, that it is a known quantity they are bring into your organization. So that's one area of the ARB that's moved to another group and maybe that's a better place for it.

### **Matt Comer**

Well, I think it depends too on which client you're talking about, maybe which industry you're talking about because we certainly see still some organizations that have more traditional project delivery, SDLCs, and certainly still have a greater involvement from enterprise architects and more governance. But we're also seeing organizations that have none of that, right? There are some organizations that we work in that it's almost "do whatever you want."



The extreme focus on the delivery, on the “get it done” or “whatever the business wants” and very, very little governance around what tools should you use or how you should build applications.

### **Vinnie Schoenfelder**

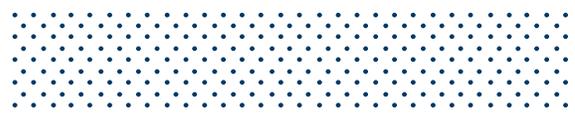
Let's jump into that Matt, because that kind of speaks to the new environment. What are some of the client issues we're seeing? What are some of the complaints we're getting from people who have moved really far in that direction? I know some of the things I'm seeing is inconsistency across the lines of business. For instance, where the data I'm collecting from minor faces may be different in form or incompleteness than yours. And what can you do over time is create inconsistent data models that require an awful lot of preprocessing to make them useful at all in a Machine Learning AI environment. So, if there was just some consistency in that it would greatly benefit your data. Also seeing delays in release cycles, lack of reuse. I know Jack, you've been on some projects recently. What pain are you seeing?

### **Jack Cox**

I'm seeing a lot of archeology going on and projects. You think you need to do this thing, but you think there's someone else in the organization that's doing the same thing. You try to find that person. And you spend days or weeks talking to everybody in the organization because everybody thinks also that someone is working on it and there's nobody who can help you find that person and you finally found them, they're in the basement. They've got their red stapler and they're working on this thing and you realize that “crap, it's different.” It's not the thing that you wanted. You've got to go do it yourself anyway. But you were told not to duplicate effort. So, now you're looking for a mysterious API that someone else has developed. It ends up not being right. So, having a central repository of that information is a great artifact of the old way, but it's something we kind lost in the new way.

### **Bill Klos**

And I guess I see it from a more basic perspective, though, too. It's a simple matter of even licensing. One of my clients had six different relationships with Salesforce and they had six different instances of Salesforce and it became a monumental effort and trying to get them to figure out how they're going to combine all that. And it was partly because the business decided they need to move quickly. And then once one individual business unit hears the other business unit is doing it, they want their's too. So now you have multiple copies of this or whatever software you want to inject in there.



### **Vinnie Schoenfelder**

Yeah, that reminds me of all the orphaned environments in the Cloud where people move test data out and lose track of where it is and the organization is spending tens of thousands of dollars a month keeping data in the Cloud that's not necessary because there's no organization around that.

### **Matt Comer**

Well, I'm seeing something that sounds counterintuitive, and that is that it actually slows us down sometimes, especially getting started on work because there's no guardrails, right? How are we going to do this? What technologies are we going to use? What languages are we going to program these services in or this front end? What's the JavaScript single page web application framework? What's going to be our backend from a database or data storage and SQL database? Whatever perspective, there's just so much green space that you spend all this time trying to figure out how are we going to do this. Whereas if you had a little bit of guidance, you could get started.

### **Jack Cox**

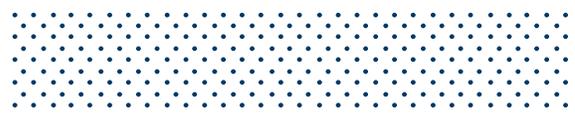
And if you have those common standards, you could have common starters accelerators to get you get the team going. So, they're not only making all those elementary decisions, but they have, "this is the the CSED pipeline for react flow for our environment, it works." We don't have to spend days getting that up and going.

### **Vinnie Schoenfelder**

Yeah, the big one for me, I keep coming back to, is reuse. I know, Matt, you and I were on a project probably 10 years ago and I remember looking at the services we needed for a couple of the apps we were working on and having a discussion with you about course grain versus fine grain objects. And when something is a single record coming back or a small set coming back to include all the fields that you may need and include grain objects because then you get easier reuse when people refactor the interface for the front end. It's not a change for you on the backend. And, of course, when large data sets come through, it's fine grain. All that thinking through, the building of those objects in a reusable way, I think we lose some of that when everyone's building their own services. Now I realize from a microservices standpoint, you can rebuild those pretty quickly.

### **Matt Comer**

But is that what you want to do? Why? So now we're focused very much, and again, this is one of those things where there's pros and cons, right? We're very focused on building things for a



use, right? Someone has a need, let's build it. And so of course that means that it's probably not necessarily reusable, right?

**Jack Cox**

Yep, Bill, you're going to say something?

**Bill Klos**

So, I guess I've always seen reuse as the holy grail because we've been talking about this for 30 years. And some of the patterns I've been seeing recently are more about the interoperability of services you have, just the ability to get information in and out. That's the next step for a lot of these companies. Reuse may come 10, 15 years down the road, but right now they just want to make sure there's a common way to get data in and out.

**Jack Cox**

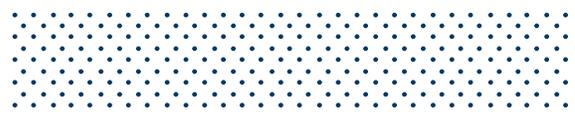
I think if you're not thinking about reuse, it'll continue to be the holy grail and you'll never get there. But I think you need to think about it at a different level, in the new way. Not we're going to reuse this entire code base, but maybe reuse, elaborate using API. So, you're on two different sides of the system, not just the whole system, but the entryway and the foundation of it.

**Vinnie Schoenfelder**

And it's knowing when to focus on reuse and when not to. A lot of times when you talk about your interface tier – whether it's responsive HTML or native mobile – it could be more difficult or less valuable even to think about reuse at that tier than it is at the middle tier where lots and lots and lots of different interface tiers are interacting. It's a one too many relationship. So, I think again, as we're having this discussion, isn't this the exact role of an enterprise architect to think through things like this to add value to the company? And is there still a role for them or is that going away?

**Matt Comer**

Well, it's a loaded question, right? I think we would all agree that there is, and what we're working on trying to figure out is how to do some of that, how to still have that value without it becoming the impediment that we were talking about earlier. So how do we provide the right guardrails, the right direction and guidance that's relevant, that's current, and that gives people enough of a framework to work in. That they can still be creative and get things done quickly, but don't have to answer that question every time of “how am I going to do this?”

**Jack Cox**

Yeah, I wonder if it comes down to the incentive of the architect. Going back to one of my truisms that all corporate dysfunction can be traced back to the compensation model. Our EA groups are compensated to make the EA group successful. They had an impact. They did all this documentation, got all these forms. Maybe that's the wrong incentive. Maybe the right incentive is "EA, you are responsible for making this project successful," or maybe two or three projects successful. And that's how they're graded and that's how they're compensated. Not on making the enterprise architecture group successful, but each individual team.

**Vinnie Schoenfelder**

Yeah, another thing that comes to mind too is application rationalization. So, to the extent that everyone's working on their own functionality, you can get a portfolio of projects over time that continues to sprawl. And you could lose control over that and then you have to work to kind of claw that back and look at how to modernize it and refactor their solutions. So, is it the role traditionally of enterprise architects to keep tabs of the portfolio of applications and know when and how to rationalize this?

**Jack Cox**

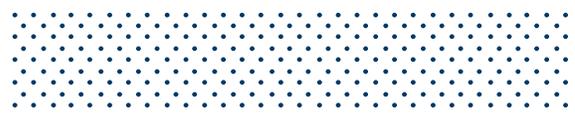
I think so, I think they're like the Lego build master. They know what all the pieces are so they can piece them together. When something, new problem comes along, "Architect, what are all the pieces that we have available that we can bring in to put this thing together to build it quickly?"

**Vinnie Schoenfelder**

Yeah and RPA is something that jumps to mind too, where if different business units pull in different that may not be consistent with the vision of the organization, where they don't want to take on the technical debt of doing RPA.

**Matt Comer**

Let me pile onto that, Vinnie. I'll say something maybe kind of controversial that I think the current state of affairs has led to a little bit of an explosion of technology in the enterprise. And I think it's important for us to remember that. So, I'm a technologist, I always have been since I was just a little kid, I love technology. If it were left completely up to me? We play with every cool new toy that comes along. And I think that's what we're seeing sometimes happening in the enterprise because there is no governance, right? And so obviously technology has a role to play in these creative, innovative solutions that people want to bring to market. But it's not the technology itself, usually, that we're actually trying to deliver maybe occasionally, right?



But most of the time it's something else. And the technology is just kind of helping to enable us solving that problem. And if we continue to focus on the technology – and we love technology – we'll just bring more into the enterprise.

### **Bill Klos**

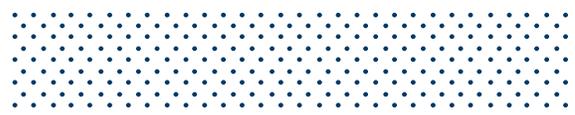
What I've been seeing is a little bit of what you kind of started off with, Vinnie. Enterprise architects were being pushed down to more of the solution architect level. And so that's where I see a crunch more down to where they had to understand the applications before at a high level. Now, they have to understand more how they fit within the overall landscape. But at the same time, the governance that they're bringing with them is less prescriptive. It's more about, "I don't care what Java script framework you used as long as it's one of these eight, I don't care what mainframe programming language you use, as long as you can create services with it. I don't care what kind of whatever server you've got, no SQL database, whatever else, as long as they can handle these types of things," and let the business units kind of figure it out. They've actually kind of pushed that accountability down to the business units to say "you guys wanted the control of this. You had shadow that were doing things that were made you go fast, which is great. Here's the guardrails we're going to put up for you. I'll help you, I'll help you build what you need to build. I'll help you design what you need to design. But at the end of the day, you are accountable for it."

### **Vinnie Schoenfelder**

So, let's take that, we've been talking about some of the challenges in the new way. There's an awful lot of good, right? These environments are more flexible, to your point, they're faster. Teams are empowered, transparent, and accountable. There's more technical talent in the role of the solution architects. The frameworks are typically more modern. People can stay up to speed faster because they're not being told what languages to work in. So, it is a pretty dynamic, fun environment for technologist. As you were talking to Matt, maybe a little bit too much so for those who are using technology for technology sake, but there's an awful lot of good in the new environment.

### **Matt Comer**

Like speed, creativity, right? That's the main thing. That's why we are where we are, I think, is we needed to do things faster and be more creative. And so that we certainly have done, we've realized that. It's just that the costs associated with it are real. And so, we have to figure out how to strike some kind of balance.



### **Vinnie Schoenfelder**

So right, what is that balance? If we were to say that there was a very good reason to move into this new environment, that we have talked about a lot of reasons why, but maybe some things were forgotten or maybe some things still have value. And it seems like what we're saying is, what's the minimal amount that enterprise architect group can do so they're not seen as a stage gate or a blocker or a naysayer or someone who's requiring tons of documentation, but someone that's just ensuring consistency and reusability across the organization? So, what does that look like? Is it a group of people? Is it an agile guild? How is that realized?

### **Matt Comer**

Well, I can tell you what it's not, right? It's not a gate driven process, right? It's not something where I'm having to check in all the time with someone to tell me that I'm doing the right thing or get permission, because we know that's gonna slow us down, right? So, it's more of a framework, I can work in a framework that I can know and understand and can work in, and that I can then also help to influence and guide. But if it's very prescriptive and every two days you've got to come and check in and get the blessing right then. But we know, that's going to impact the team's performance.

### **Jack Cox**

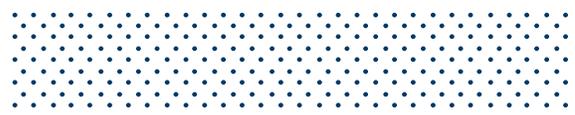
I think one of the things that looks like that the architects no longer control the landscape, but they understand the landscape, for the most part. And so, when somebody comes and says, "we want to do this," it's okay, that's good, it's going to fit into the landscape. It's not going to clash, it doesn't have the security issues, maybe it's a different thing in this architectural landscape, but it's good. We understand it, we can bring it in, we'd manage it, and have it be part of the longterm life cycle of the business.

### **Vinnie Schoenfelder**

Kind of comes back to the Bezos memo and enterprise architecture design patterns. So, when I think of the role of an enterprise architect setting the patterns and standards and saying, "this is how we should do things," as long as you meet that design by contract, as long as you meet that interface need, that should be enough. And then just hold people accountable for those decisions. Bill?

### **Bill Klos**

One thing that I've seen changing is it doesn't take 20 years to build an architect anymore. So the architects that are enterprise architects now, if they come down kind off the mountain and



then discard their robes, there's a lot of smart people at that application architect level, that solution architect level, that they can actually learn simply because when that person became an enterprise architect, they were still building mainframe applications. And the world has changed a lot and there's a lot to learn from that first-year architect, those highly, highly skilled developers. Getting those people together as more of an architecture cadre or cobble is actually been more successful, I've seen, than still having limited groups that are coming down from the mountain.

### **Vinnie Schoenfelder**

Yeah, I would argue it's still a craft, that as smart as you are, and as modern as you are, if you haven't been through a dozen different projects different contexts, you're not going to have that background to know the tone. Like Matt said earlier on, it's not making the technology decision, it's making the right decision for the organization that's happening to be using the technology.

### **Matt Comer**

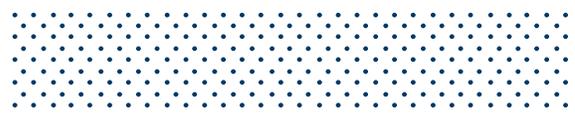
It's the vocabulary, right? You don't have a big vocabulary until you've read a lot of books. And so, you've got to have had some experience and some exposure to be able to understand. Because often what I find is interesting about architecture and design is that it's that "aha moment" that you have of, "Oh, I've done this before, right?" We've solved this problem before, we've used a tool like this before, we've seen a pattern like this before. Maybe it's a different technology, we called it something else, but it's the same idea, right? And then you begin to expand and say, "we can apply these ideas and new ways," but you have to have some experience to be able to get to that point.

### **Bill Klos**

And in the successful companies I've seen that actually start having architecture groups, there's almost like an internal user groups where they bring developers and they bring architects in, and they do things – they come up with problems, they help solve them over there and they kind of teach both ways, both directions. And I've seen that be very valuable.

### **Jack Cox**

Yep, I think one of the things that the new world looks like, a good world, is that the architects are getting their hands dirty. They're still occasionally writing code, putting systems together, doing dev ops, seeing how things work, seeing how their decisions impact delivery and impact operations.



To make another point, back to a point Vinnie made earlier, you mentioned the Bezos memo completely out of context. But anyway, the Bezos memo was a memo that Jeff Bezos wrote, back I think in 2002, back when Amazon just sold books and a few other things. And it highlighted basically six architectural principles that everybody needed to follow within Amazon. And there was a seventh principle that said, if you didn't follow the above six that you were fired. And it came down from the top and it was very clear memo, but it was six very simple rules. And from that AWS was able to be creative; now it's a standalone \$40 billion business because they architected things correctly.

### **Vinnie Schoenfelder**

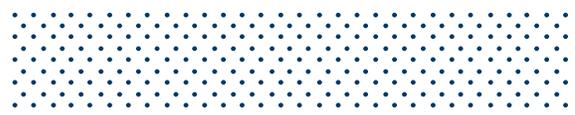
And coming off of that, they also hired intelligent people who were empowered to make their own decisions. There's some downstream there. So, what would we recommend guys to companies who have moved into this new space? They're feeling some pain. We talked about do you stand up an enterprise architecture group if you don't have one, or do you create like an agile and have a pocket of enterprise architects with responsibility, as Jack said, to a portfolio of projects? So, you don't have one person for one project, you have one person for 10 projects. These seems like normal steps that we used to do. Is it just that simple? Or what would be some of your first steps you would do to sort of improve the deliverability and governance that's happening right now?

### **Bill Klos**

Get a handle on your application inventory, find out what you've got, find out what talks to each other, find out what doesn't, figure out your sunset program and start putting it into execution.

### **Matt Comer**

Well, I think you've got to, as part of that, figure out what kind of organization you are. There are some organizations that can tolerate a lot more diversity of technology and of different approaches and tools etc. It's part of what they do, right? They maybe value the rapidity and the creativity and maybe they want to hire the very best developers, so they've got to allow people to express that. Then you've got some organizations that may favor a little bit more stability maybe more concerned about how do we support all this stuff, right? Maybe they're more risk averse from the standpoint of maybe security and data privacy. There's all these things that you have to consider, and knowing who you are is then going to allow you to set up a program that's going to let you have the right level of control, governance, not the right word even to use.

**Vinnie Schoenfelder**

As you were saying that, it made me think it also changes the type of people you're looking for for that role. You don't want people who only have the engineering angle, right? You want people who have enough acumen that can go in and understand the business problem behind it, behind the need, and know when to be pragmatic and when to be academic and where to be somewhere in the middle. And those are hard resources to find, those are the unicorns. But I think that's why as a small group that's serving multiple delivery teams.

**Jack Cox**

I'll go ahead and express a controversial opinion that I don't think an enterprise architect should be a long-term career role. I think it should be something that somebody does for a period of time and then goes into another role. Either goes to be a tech lead, goes to be a product owner, something like that, but they don't stay as an architect because I think if you stay there too long, that's when you start to crystallize, so of like politicians. You want term limits on these guys.

**Matt Comer**

So, you don't have any perspective on what's really going on that's right in the ground anymore.

**Jack Cox**

You lose perspective, you become ivory tower, you need to bring down the ivory tower and the architects need to be, again, getting their hands dirty in something other than just enterprise.

**Matt Comer**

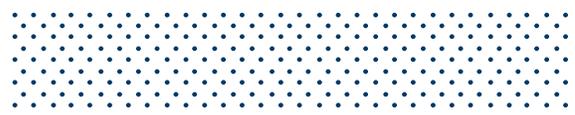
Do you think cycling through some roles, where maybe you have a little bit more of an enterprise architect style role, and then a little bit more of a systems or solutions, or project architect style role?

**Jack Cox**

Yes, product owner role. Yes.

**Vinnie Schoenfelder**

You would have a lot more empathy for the teams that are trying to deliver if six months ago you were one of those teams.

**Jack Cox**

It would also help that team interact with the current enterprise architects and know how to work with the process, how they think, so that it creates less friction for your particular project.

**Bill Klos**

How does that affect continuity, though? I mean, if you've got somebody say you're an enterprise architect, Jack, and then you leave and I come in, it's like, alright, go everywhere.

**Jack Cox**

Well, I think you would manage continuity by not changing out all of your architects all at once so that a stream flows through the architecture group. Sort of like the Supreme Court, we don't change Supreme Courts all at once, right? We do it one or two at a time, over generations.

**Bill Klos**

So, you'd have to die and then I become enterprise architect.

**Matt Comer**

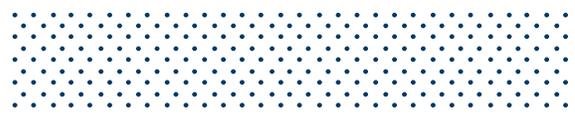
Well not the long, not the lifetime in term. But again, I think you've got to also have some people, or at least have a way that you can then train some people to understand some of the things that are important for an architect to understand, again, around nonfunctional requirements and business requirements etc. to say we've got to be able to support these applications, we've got to be able to afford them, and pay for them, we've got to be at a licensed, we've got to be able to do this. Do they perform well, do they scale? So, all the things that I've got to put a little bit of a different hat on than just a developer.

**Jack Cox**

To defend my controversial opinion, I think having the former enterprise architects be back into the team, they can bring that knowledge back into the team and make it more more owned. What's the right word? It becomes part of the team and not something artificial or external from the team that they have to fight against. It becomes part of the team's DNA.

**Vinnie Schoenfelder**

Well also, Jack, I think what's interesting there is that you could go from being an enterprise architect to being a solution architect on a team. So, you're still a technical leader and you're building technical people around you, right? You're making developers better. You could be



in new areas where you don't have the experience so you're gaining that perspective and then you can go back to being in a different project again and another year or two. Right. So, it's just a, it's, I think it's a neat rotational idea.

### **Bill Klos**

To me Jack, that's not your controversial opinion, that actually makes sense. The InfoSec one that you made earlier, that's the one that I don't want to live in that world.

### **Matt Comer**

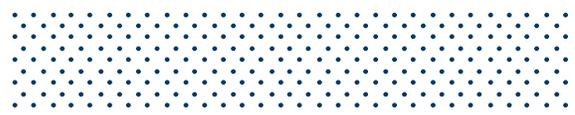
Well yeah, security and convenience are orthogonal, right? So, everything would get very inconvenient.

### **Matt Comer**

But talking about the ARB, I think that's a key idea around sort of this question you have of what could it look like or what should it look like? And it illustrates this idea of a process that's a little bit more collaborative and a little bit less prescriptive. And I think that overall, that's probably a theme of what we're all talking about, right? And that's that governance with a capital G, right, is part of what the problem was. And so, in today's world, there should be some rules, but rules are also made to be broken, right? Because I can remember back in the day if you ran a foul of some rule and needed it some sort of exception, well, you're not going to do it on this project's lifetime, right? So, we need to have guidance more so than prescriptive rules. And I think that a team based, collaborative, communicative type approach facilitates that.

### **Vinnie Schoenfelder**

And I think we mentioned the makeup of that team, right? So, we have a lighter weight team – with a lower case G on governance or just a smaller uppercase G. So, such that they are talking about frameworks and patterns and approaches and the correct ways to do things as an enablement team, but with the acumen to understand that the business need know where and when to be pragmatic and academic. And to Jack's point, I think you become that person. You create those type of people by working them into a lot of projects. That's the perspective you have to gain to have that flexibility and perspective. We also, kind of rounding out our recommendations here, need people in that role to understand the full portfolio of applications, to know where a reuse and consolidation can occur, so that we don't get this hyper sprawl of solutions and services. And I think it can function as part of like an agile guild that does architecture reviews, but maybe not doing code reviews, just sticking to the final. Does it fit our guidelines before we publish it? But up until then kind of let people go and have



the flexibility they need. Is that an accurate summary, did I miss anything?

**Matt Comer**

No, I don't think you did. I think that's a pretty accurate summary of the things we talked about.

**Jack Cox**

Yeah. I think you missed the part about InfoSec becoming the ARB.

**Vinnie Schoenfelder**

Yeah, that was purposeful. I followed Bill's lead on that.

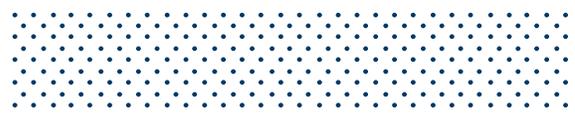
**Matt Comer**

One thing you said about architecture review though, I want to be careful about, is that one of the pain points we had in the past was that things that are gate driven, things that happen later rather than sooner, tend to be so much more painful. I want to know that I'm doing the right thing right before I start doing it. Or as I'm starting to do it, I don't want to find out about it when I'm nearly done. And that's one of the things that we saw as a big problem in the past, you would run a foul of something that you didn't even know about and ended up with either having to convince someone that it was okay, get some kind of exception, or remediate a bunch of work that you had done. And so, something we can be doing is ask more questions, get more guidance, get more advice.

**Bill Klos**

And Matt, I think that's where application inventory comes into play, because at the end of the day, what you end up doing is you put that all into a model, and whenever you want to make a change to any application that you've got in the stack, you can bounce up against that model to see if anything that you're going to do is going to affect it or what applications is going to touch. And at that point before you actually start doing your work, then you get a feel for it, everybody can take a look at. It's like, "alright this is going to impact me this, this way, this way, this way, and this way; and you need to make these changes, or have you thought about this?" So, there's a little bit, I wouldn't call it a gate, it's just more of a FYI before you start this project, here's some of the issues you're going to have. And hopefully by that time everything's documented correctly in the model and then you know what you're stepping on and what you're not.

**Vinnie Schoenfelder**



Yeah, I'll take that one step further. One of the things I think that enterprise architects do really well and it's really, really necessary, is they look at the teams that are doing "we need this functionality now," to your point, Matt, and that's what we're going to build. If you look at things like RPA, you know a lot of business units want to put RPA, in robotic process automation – Jack and I always joke that it stands for "really poor architecture" – it's integration at the glass, interfaces change, things break. And so, it really is the wrong place to do integration. However, there are legacy systems that are out of support and there's immediate pain and, right now with COVID-19, there were some certainly some situations where RPA could provide some immediate relief in a very short period of time with a lot less money.

When that happens, when that team does that, there has to be an enterprise architect group that says "we are going to put on our timeline now when that's coming back out, when we're going to build the services to do the integration at the proper tier or when we're going to modernize that system." So, RPA, in my opinion, can be thought of as as structured technical debt. And there's other types of structured technical debt, but if there isn't a single place – to your point, Bill about application inventory – there also needs to be an inventory of structured debt and plans for coming back off that and not letting it get out of hand. Because that team that built that functionality, they're going to go on to the next thing and that's not going to be their concern.

### **Matt Comer**

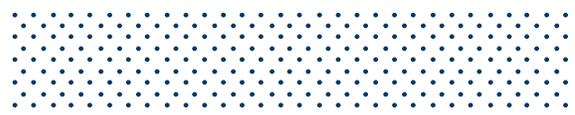
I completely agree. And that I think that's one of the things that we need to figure out how to interface everything we're talking about here with an agile delivery process, because we love to work in these agile delivery processes because they let us be more creative and be faster and respond and not get locked into solutions that ended up not being what anyone wanted anyway, etc. But they are fast moving and they do move on. And you do build technical debt. And although we all talk about the fact that we're supposed to have time allocated to burn that stuff down, I don't know about you guys, but I seldom see it actually happen.

### **Vinnie Schoenfelder**

It's the first to go besides testing, those are the first things that get cut.

### **Matt Comer**

So, we had to figure out how to plug some appropriate level of enterprise architecture into the agile delivery process in such a way that it doesn't break that process. It still respects the process, but you take the time to do the right things.

**Jack Cox**

I think there's a lot of compliance gates that need to be brought into the agile process and there's always friction around those gates. Again, bringing up InfoSec, there's compliance gates there. And we can have this same conversation about how do we keep InfoSec from being a gate that people hate addressing or try to work around? There's other types of compliance gates.

**Vinnie Schoenfelder**

We didn't even talk about data, making sure the data's clean and going into the data lake and all that sort of stuff. So, there's a lot of other points around that could be a whole separate podcast. I want to be sensitive on time. So any closing thoughts? Matt, Bill, Jack?

**Jack Cox**

Nope, good conversation. Always enjoy talking to Matt and Bill.

**Vinnie Schoenfelder**

Well, thank you all for joining us. Yeah, peasure having you guys. Bill, thanks for calling in from Columbus. Good to have you on the phone remotely. And for those listening we'll be doing more technical podcast soon, so please stay tuned. Thank you.

*This document contains general information for educational purposes only. In posting this document CapTech is not providing business, financial, investment, tax, legal, or other professional advice or services and should not be used as a substitute for those services or should not be used for any decision or action that may impact your business. This document is not a solicitation for business for any of the companies or the organizations included in it, nor does CapTech recommend or endorse the services or products provided by these companies or organizations. CapTech provides this document on an "as is" basis and makes no representation or warranty as to its suitability or validity for any specific purpose. CapTech is not responsible for any loss sustained by any person or company who relies on this document for making business decisions.*

*This contains CapTech's copyrighted material. If you wish to use copyrighted material from this site for purposes of your own that go beyond fair use, you must obtain permission from CapTech.*