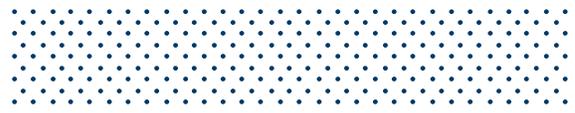# The Aspects of High Preforming Agile Teams

CAPTECH TRENDS | PODCAST | EPISODE 2 TRANSCIPT

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.

In our previous podcast, we discussed how Agile, DevOps, and Modern Architecture have prepared development teams to continue to deliver in this socially distant and remote working environment we find ourselves in, obviously as a result of the COVID-19 pandemic. This generated a lot of conversation and so we want to go a little deeper this time into what makes a high-performing team and, specifically what makes a high-performing team effective with remote delivery. Joining me again is Dennis Bowne, a Principal at CapTech who leads our services and who has built our Agile practice area, as well as Paul Given, a Managing Director at CapTech who currently leads our Agile Delivery practice area. Welcome gentlemen.

Paul Given

Hi, this is Paul.

Dennis Bowne

Hi, this is Dennis Bowne.

Vinnie Schoenfelder

So, as I said in our previous podcast, titled "Successful Project Delivery During COVID-19," the premise was that a lot of our clients are asking us what they can do to be effective in their project delivery during this time. And what we have seen is that companies who have invested in Modern Architecture, DevOps, and methodologies like Agile over the past 10 years and done the hard work to really make that effective in their organizations are weathering the storm pretty well. And those who aren't are struggling a bit. And we provided some thoughts and opinions on what are some immediate steps they could take to shore that up. And it's interesting because it's a little counterintuitive. Agile is a methodology that promotes co-location, and we're in a very different environment from that now.

But the aspects of it – transparency, responsibility, a cadence of delivery with sprints – these are all things that promote good project development teams and effectiveness. It's one of those things where whether it's methodology or architecture, if you do them well and you invest in them, they paid unexpected dividends. No one planned for the pandemic by implementing Agile, and yet it's still very helpful and beneficial. But you guys wanted to go deeper on remote teams. So Dennis, why don't you help us set the stage and let us know what
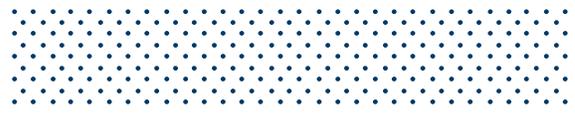
we're going to be focusing on today.

Dennis Bowne

Thank you. That's actually a great lead in. We did talk about, as we set the stage, that almost all of our teams have been disrupted in some way. Even those teams that are fully remote, they're working in this new environment where they have kids and dogs in the background and they're not in their normal work environment nor are their coworkers.
We did get some questions after the last podcast around what does a successful team look like? So we want to focus on some of the highest-performing teams that we've seen in this environment. And most of those are Agile teams. I'll call out, and I know Paul and I both agree on this vehemently, that you don't have to be a pure-play Agile team to do some of the things we're going to talk about. But what we want to give you is the example of what that high-performing team typically looks like, what we're seeing. And then we'll go into a little bit of detail around in this environment, are there some smaller tweaks and things that you can do to make your team more effective?

So with that, I'll start off, and I'd love to have a back and forth with Paul, who really is an expert on this, not only for CapTech, but in his previous roles he's led Agile transformation efforts for many of what are now our clients. So, one of the big things I talk about, always, is long live the team, right? So, in order to have a high-performing team, you have to build that cadence and trust and working relationship among the members of the team. So, if you have teams today, let's make it real personal for right now, don't break them up. Don't make big changes right now. They've got a cadence, they've got a working model. Even if it's not ideal, let's not make changes. And in the long run, what that looks like, is you're investing in keeping teams together and bringing work to teams that have the ability to deliver a product for you, not assembling teams every time a new ask comes down the pike. Paul, I want to hear from you how you feel about that and what you're seeing.

Paul Given

Yeah, Dennis it's really important. Teams that have been working together for a longer period of time have built a trust amongst each other. They have established communication patterns and established signals to each other when they disagree. And that healthy conflict is going to come to play when they move remote. So, the ability to, well, "wait a minute, I thought you said this, or that ability to say, I thought we were going to do this, that's what came out of standup." But when something changes in the middle of the day and you don't see the change because you're not in the office, they have the ability to reach out to each other and effectively communicate that conflict, versus just kind of plowing forward. So, they will stop, adjust, and
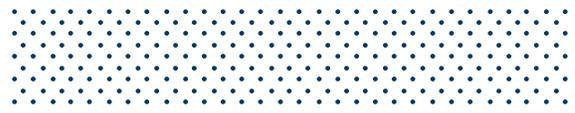
react much more quickly in a much more constructive manner than a brand-new team that has never worked together with each other.

Vinnie Schoenfelder

So Paul, that seems kind of obvious from the standpoint that you just get to know your coworkers, right? You know the idiosyncrasies, you know what they're saying behind what they're saying, or the question behind the question. It's just a comfort level of working with people. Is that what you're saying? Or does it go deeper?

Paul Given

Vinnie, it is what I'm saying. But it does go deeper. So, Vinnie, think about the first time we met and we had a good conversation about unit testing, but we were unsure where each other was coming from and we didn't really have that "oh yeah, we both get it and we both understand how important unit testing is." Now, when we see each other in the hallway at CapTech, we're like, "Hey, we need to look at that." You and I are on the same wavelength and there's body language and communication there. Now, when we move remotely, you and I already have that relationship and I can hear it in the tone of your voice. And it's happens much faster – that common communication already happens much faster. Whereas if I were having that conversation with another CapTecher who's on that project, who doesn't know me from Adam, then guess what? It's going to take a little bit longer to have that conversation. Now it's still an important conversation and we need to have it, but it just happens more quickly and more naturally.

Dennis Bowne

Yeah, and I think one of the other things I'll add to that is that, in addition to just being more comfortable working together, you also understand the skills and the talents of the individuals around you and who is the best suited to work on each piece of work that's coming through your pipeline. So just the handoff and the building of the Lego bricks that make the product you're building are a lot simpler. And it sounds intuitive, but so many organizations are "projectized" and they'll say, "Alright, we've got a new thing we need to build, let's assemble the team that we need to do it." Instead, it's "what's the best team that we have that can build this the most efficiently?" That's the mindset we want to take.

Vinnie Schoenfelder

Yeah, to kind of close this one out, when you think about a project and project delivery, you think about the roles you need and who the best people are for those roles. And what you guys are saying is the relationships between those existing people in those roles is equally as
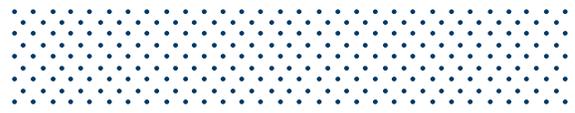
important. What I'm reminded of is that, in sports, like take basketball for example, the best team, the one that can win the championship, is not the All-Star team, right? You can assemble a group of All-Stars and they're going to lose to the best team who doesn't have the best players, but they've been playing together all year. Right. So, it's a very similar analogy.

Paul Given

Yeah. If you want to go fast, go as an individual. If you want to go far, go as a team.

Vinnie Schoenfelder

Right. Okay, Dennis, what's next on your list for effective teams?

Dennis Bowne

Yeah, so there's some healthy interaction on this one – the highest-performing teams typically are multifaceted teams. These are teams that can build a feature for your organization. For instance, an anti-pattern would be that you have one team that does the front end, one team that does the middleware, one team that does your backend connectivity, etc. And then they're having lots and lots of handoffs, handshakes, dependencies that they have to manage. A more efficient model is to have teams that are able to build your product, or at least a feature of that product, as a complete unit. So, you have all the different folks on the team. Yes, it's not a homogenous team where they all speak the same language necessarily; it's a group of all front-end developers that are jiving and working with each other. But as a team, you have all the skills you need to assemble that package. And what we don't recommend right now, so taking this to the today, is, "Great! I heard what you guys said, I'm going to rip down my teams and suddenly build a bunch of feature teams." Now's not the time to inject that change, but small changes where you start to inject another person to a high-functioning team that was just missing that one role, that might be something you want to try. What do you think, Paul?

Paul Given

First, I would say, like you said earlier Dennis, is don't go changing everything. So, if you have a front-end development team and you have a middleware team, and you have a persistence team, that persistence team can be a team that works with a system of record or anything else. But you have kind of three layers of teams and they're all now all of a sudden working remotely. I wouldn't change those teams. What I would really do is focus on, "How do those teams coordinate amongst each other and what are some new norms you can develop to enhance that coordination?" Because that coordination between those teams, when they're in the office, they're probably close to each other and they act as kind of one big team of teams. Well, how do you create that? So, we might only have a Scrum of Scrums amongst those teams
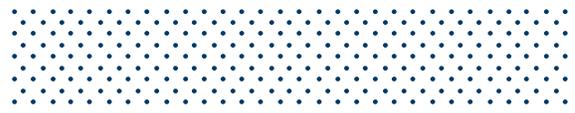
twice a week when you're in the office, where you probably want to have that daily when you're working remotely, you might even want to integrate more often. So, if they really typically only integrated in the middle of the sprint or the end of the sprint, you might want to move to getting fast feedback amongst the teams. Let's do that integration daily. And you're going to go to some DevOps techniques and things to get there.

Vinnie Schoenfelder
As you say that, and these are good specific examples – things people can do to improve remote delivery – but are there additional things we have to do? So, what's the expectation of the impact on velocity? Are we saying that because we have to change the way we do things – maybe add more communication – that we're going to have a particular impact to our current velocity?

Paul Given
Yeah. So, I have a very particular belief on that – and people will argue with me on this and not others –but it's a proven pattern of teams that finish early, accelerate faster. So, when you have a big change to how a team works, don't try to commit to your full velocity, commit to maybe 80 percent of it. Make sure you can get that done and then pull things in. And that also goes for teams of teams where large integration is important and really critical. So, you know that you're going to hit new issues and new impediments, and we don't know what they are, but let's anticipate that they're going to be there. Let's plan a little less, let's commit to a little less in our sprint planning, in our PI planning, even a shorter horizon and let's make sure we can get that high-important/high-risk stuff done and then we can pull stuff in and realize our new true velocity.

Dennis Bowne
Yeah, I agree with that point. And I think also by not committing to a full load where everyone's at breakneck speed throughout the entire sprint, when you have changes that create efficiency, you have the room to make them. So, Vinnie to your question, sometimes you're going to have to make changes to the way your team operates and works and you have give them the spare bandwidth to do that. Yes, if you were to not have any change, you could see you could slam in a little more work, but over the long haul, that's going to keep your team at its current rate and we want them to be faster, better, smarter.

Paul Given
We've just seen time after time, that teams that actually allow themselves some slack in the system, actually accelerate faster and are more innovative because they're given the ability to
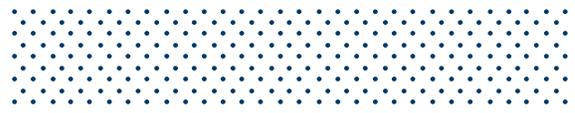
deal with the issues that come up. And the innovation in dealing with those issues is what makes a team high performing.

Dennis Bowne

And that's also a good segue to one of the other points that I'd like to talk about. Paul, I'd love to get your opinion on this – along that line, building empowerment into your team, right? So, the team that plans, estimates that work, and commits to their work rather than having that work given to them, estimated, and committed on their behalf, has more ownership, has more skin in the game, and almost always those are the high-performing teams in my experience. What do you think Paul?

Paul Given

I agree 100 percent, Dennis. Teams that own their own work and basically are developing the solutions for the problems that they have really good context around, are high performing. There's another component to that culture of commitment though. They're going to commit to solving those problems, but there's an organization around them that's committing to removing their impediments for them. And that's something that you have to look at. If you look at that – the cycle time for removing impediments for teams, new environments, things that are outside of their control – if you look at how that's going to change as everybody works remotely, not just the teams, but the leaders for the teams. If you start to understand that and understand how it's impacted, that's a good leading indicator that working remotely is working for your company. Vinnie, what are your thoughts?

Vinnie Schoenfelder

Well, what concerns me when I think about this is a team can be fully committed to delivering something, to your point. But I'm thinking about companies who are new to this type of methodology. There's a culture outside of the team. And so, I've been on a lot of projects where the individual project teams are adapting to Agile and they're taking it seriously and they're going through the training, and they're making the paradigm shift in their head, which you need to make. You can't just map it to other methodologies you have used before, right? You really have to make the shift. But other parts of the organization have not. So, you know, procuring hardware is still taking a very long time. Getting things through security is still taking a long time. Getting things through legal is taking a long time. And so, these are external impacts that limit your ability to affect that change and honor that commitment. But those things seem like they were more difficult to change quickly, especially in this environment where we're remote. So, what advice can we give people if they believe that they've got high-performing teams, but they're struggling from an organization that's not
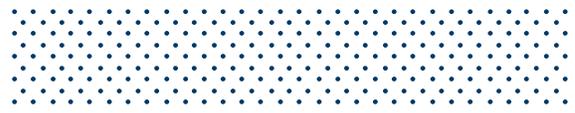
adopting the same principles?

Paul Given

Yeah, so, my advice is, find those leaders who are willing to remove those organizational impediments. There is usually an executive champion there and that executive champion is usually willing to engage in those conversations. Reserve time for those conversations. Have them more often. So, Vinnie, like you said, its compliance, getting hardware, getting environments set up, all of those systems around the team that are necessary for a team to deploy their valuable product. All of those things have a traditional process that may not be working as fast as the team or may just have different way of working from the team. So, generally, the product owner and the stakeholders want to see the product delivered as quickly with the highest quality as possible. And if that's the end alignment for everybody, bringing those others along with you usually requires some leadership, counterintuitive leadership to say, "Hey, let's look at the system as a whole and let's understand it." In a remote world, it's very difficult to lay all that out and work through it. So, you've got to take off small chunks at a time, really small chunks. Figure out what the most impactful impediment or organizational impediment is, and just attack that one for a couple of sprints and keep working for it. We don't know what the future holds, we don't know how long this is going to last, but let's just get started on getting through those things a little bit at a time. And as the organization sees that little bit of progress, more people jump on board and it tends to accelerate.

Dennis Bowne

Right, I think the other thing I'd say is that there's two types of changes: there's big change and there's small change. So, for instance, if you have a big change you're trying to enact – such as organizational change, changing the mindsets of the way your business groups or the supporting groups that are part of the process with that team – you want to pilot that in scale with champions that can prove to your organization this works. There's also small change, just moving to an empower model of working with that team and saying, "We're going to look at the work that's coming in. We're going to actively be part of the estimation process and we're going to predict what we can and cannot get done and get a couple people to help you with that process, who are going to change their mindset and be more open to being led by the team to what can actually be accomplished." That's not a really big organizational change.

Vinnie Schoenfelder

So, Dennis, I know this topic is broader than the current COVID-19 situation we're in, but we are in this situation, right? So, as you talk about big change and small change, I'm pessimistic
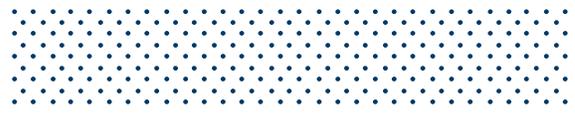
in that I don't think a lot of companies are going to take on an Agile transformation organization-wide right now, while people are working remote and it's probably focusing on more immediate demanding needs, right? So, as we go through and you guys are making these recommendations and they sound great, but they also sound a little bit easier said than done, right? So, what are the things you could do, short of the big stuff you're talking about? What are the first three things you would recommend to somebody who is a little bit behind the curve on this, doesn't want to pitch a full transformation, but does want to deliver more effectively?

Dennis Bowne
Yeah, that's a great question. I think the very first thing is, if you're really looking to adopt some of these principles wholescale, a lot of the functional changes and moving to an Agile model should be reserved for organizations that are already on the pathway. It's going to be much easier to nudge things along and try to enact better principles, than to, right now, enact a full transformation and change. I actually would not recommend that for those teams that are heavily Waterfall or just don't function in the way we've described today. You can look at your current state processes, and we did cover some of this in the last podcast, so I've referenced people back to that. There's small individual things you can do such as having regular touch points throughout the day, there's the culture – creating more-open transparency and visibility, not just inside the team, but to your stakeholders. These are all things that you can do to enhance your current processes no matter what methodology you're following. For Agile teams, if you're not doing some of these things, if you don't have that team-level commitment, if the things we're saying sounds like this is something that I could get benefit from, it's worth talking about making some minor enhancements and changes with the team on your side, not enacting those and putting those on your teams.

Vinnie Schoenfelder
All right, so that was a lot and good. But Paul, why don't you give me in one breath or two, three things they could do – like Bing, Bing, Bing – and then we can talk about them, but very summary.

Paul Given
Sure. So, my advice would be to focus on fundamentals. So, if you have a budding Agile team, focus on three fundamentals. Make sure you have a strong definition of ready with a deep-ready state backlog. Make sure you have a strong definition of done so the team knows when they're done. If you're not Agile, focus on your fundamentals. If you're Waterfall, make sure you have strong requirements. Make sure you have a strong testing approach. And go small;
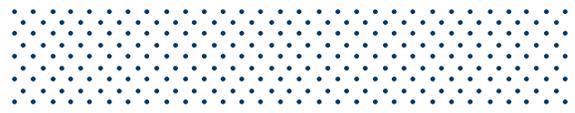
smaller projects are easier to coordinate and communicate.

Vinnie Schoenfelder

So, yes, thanks. And that kind of leads to my next question based on this, which is if you're talking about larger organizations, how would you recommend that they affect this change in the organization? Do you start with one project and make sure that it works well and then propagate it out? Do you make broad-sweeping changes? If you do start with smaller projects, or onesie twosie things, do you do it by most impacted the organization or the least risky cause as you're introducing change? I'm struggling with how you would take these things and propagate them at a larger organization.

Paul Given

So, there's two approaches there, Vinnie. One is – and I've seen this in a remote world, as well as in the office world – if you think about a Waterfall project that is not going well at the end, what happens? The testing team is there and what do they start doing? They start working defects and there's a whole bunch of defects. And how do they work those defects? Well, there's a daily defect call and the developers go fix the defect and they test it and they get through it. They move to an Agile model. So, when you have a burning issue, typically even in a Waterfall process, companies move to a more iterative, Agile development to get to something that is of quality that they can release.

So, whether you pick a high-risk project or a simple project that's not necessarily as high risk, that gives you some ability to fail, if you're rigorous and you follow the fundamentals, you'll be successful. Looking at the company, then it becomes, what's your risk tolerance? If it's critical – you've got to get done well – you may not actually convert it to an Agile model. You might just take an innovative approach to getting it finished. If you're in a highly regulated environment and you're developing medical devices, you're not going to put a piece of invented software on a respirator and do this in an Agile manner for the very first time. No, that's probably not a good idea, but let's pick something that's maybe a support system inside your company that you know can work well and then figure out and learn from that. So, it really depends on the company's risk tolerance.

Vinnie Schoenfelder

Gotcha. I want to be sensitive to time here. Dennis, is there anything else that you wanted to touch on?
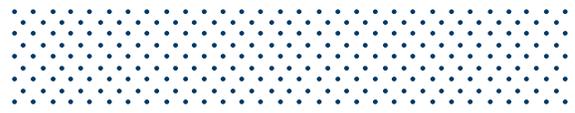
Dennis Bowne

No, I think the biggest point is that all of these lead to creating teams that are consistent, that are predictable, and that have good transparency with their stakeholders and among each other. Whatever your condition is right now, if you can try to keep those principles top of mind, that's going to help your teams perform.

Vinnie Schoenfelder

Great. Well, gentlemen, thanks for your time, appreciate it. This is great information. And for those listening, please stay safe, practice social distancing. And I hope this information helps you have higher, productive teams. Thank you.