accenture

CLOUD BREAKS BARRIERS TO AGILITY

VIDEO TRANSCRIPT

Speaker 1: Agile Amped shares stories of bringing agility and humanity into the workplace and beyond. Inspiring and provocative voices speak on topics from technology to business to living change. Dedicated to building a more agile world, Agile Amped is brought to you by Accenture.

Alalia Lundy: Welcome to another [00:00:30] episode of Agile Amped. I'm your host Alalia Lundy. Today, my guest is Adam Burden. Adam is Accenture's chief software engineer and a senior managing director leading technology services for Accenture in North America. Over his 28-year career, he has led many pioneering technology engagements, including serving as lead architect for the world's largest wholesale electricity market.

And I have the pleasure of speaking with you today. Welcome, [00:01:00] Adam. Thank you for joining me.

Adam Burden: Thank you. I am so happy to be here. I've been a fan of Agile Amped for years, and it's such an honor to have an opportunity to be a guest.

Alalia Lundy: Awesome. Thank you. So, wow, 28 years at Accenture. Tell us what has that journey been like?

Adam Burden: It's kind of like Forrest Gump would say, it's like a box of chocolates. You never know what you're going to get next. Right? My career started as a software [00:01:30] developer and a programmer working on a customer utility system actually here in Florida where I live now. And it's taken me on the way to roles running help desk, to operating testing organizations, to starting up an application modernization practice, to helping to lead our global software engineering business and becoming our chief software engineer. And now here I am running our North America technology [00:02:00] business. And it's been an exciting road along the way to be sure. But, indeed, it's one where those opportunities that are out in front of me, they've always been surprises about what happens next. And I relish in that difference in that career trajectory I've had that has given me that diversity of experience. It's been fun, interesting, and exciting.

Alalia Lundy: It sounds like you have definitely had a front [00:02:30] row seat to the evolution of agile product delivery in technology. And so almost a year, we've been in this worldwide pandemic. And although many organizations, large and small, have been forced to fast-track their digital presence in cloud-based technologies, there are some organizations holding back. You talk about this in your article “Breaking the Four Barriers to Cloud.” And I once heard you say, “cloud unlocks agility.” [00:03:00] How and what does that mean exactly?

Adam Burden: Yeah. I have had a front row seat to this. And it's been an interesting journey like as a software developer at my heart and then also as a leader working out with our clients in some of their biggest transformation journeys.

Cloud in itself, moving to cloud is not something that makes you more agile as an organization. There are lots of companies that look at cloud as a cheaper data center. It's actually the

[00:03:30] wrong way to look at it. Yes, you might be able to save some CapEx and some OpEx costs by moving some of your application workloads to cloud. But in the end, if you don't take advantage of the elasticity that the architecture offers, the ability to spin up test environments, the ability to do things differently from a software delivery pipeline, then you're missing the opportunity that cloud really gives you. It's about moving from something that's a fixed mindset [00:04:00] into something that's much more elastic in nature.

And it doesn't, you don't necessarily have to move to public cloud to do that either. You can do that in a private cloud too. But it's that transformation – like, if I just take what I have and I move it in the cloud, yes, maybe there's some incremental unit cost savings I get out of it. But if I really want to unleash the power of agile, I have to do more than just that. I have to change my culture. And cloud is the [00:04:30] elixir, the catalyst that helps to make that happen. Because absent that, if you want to do things in more of an agile software delivery fashion or to be able to expand and contract more readily as the market forces changes or otherwise, you're sort of fixed by the constraints that you've put around yourself. There's a cultural change in embracing working in cloud that's required to go along with it. But without those two ingredients together, you can't make that happen. [00:05:00] So cloud is, indeed, a reagent. It's the thing that really starts that reaction and allows our clients to do that, whether it's public, private, or hybrid adoption.

Alalia Lundy: Okay. So you mentioned a lot of buzzwords in there that I immediately think agile, agility. And so you said that-

Adam Burden: Sure.

Alalia Lundy: ... cloud doesn't make us agile. It unlocks it.

Adam Burden: Yeah.

Alalia Lundy: So for those organizations that are holding back, it brings to mind [00:05:30] where else is agile not penetrating? Is it that it's not penetrating? Are organizations not embracing it? What's the challenge that you're seeing in that space?

Adam Burden: Right. This is a big one. I think this is a really important thing that on any cloud journey for a big enterprise client, or even for medium-sized businesses, that they look at their set of business applications. It's very rare that you have someone that moves into cloud that's [00:06:00] an entirely greenfield business. Now, sometimes you have these born-in-the-cloud businesses like an Etsy or something like that that is just one of the pioneers in doing things in DevOps

and really embracing agile and extreme programming and the different evolutions of that.

But for an established enterprise, their value has come from, in many ways, the technology systems which they've collected and built over the years. And they just can't abandon that by moving [00:06:30] to cloud.

On the other hand, those barriers, that's exactly what also holds them back. So it's kind of like a “damned if you do and damned if you don't” type of thing. It's a big, expensive thing to go in and reinvent this, reinvent some of those apps, and it can be time consuming. But the yield that you get out of it, the agility that you get in your business is very, very different.

Let me give you an example. So let's say I'm a utility. Right? And I have a really important application that I [00:07:00] use in my business called an OMS, or an outage management system. This is the system that for most of the time, it's dealing with, oh, this one person has their power out, and another person has momentary outages or whatever. Let's say it gets five or six outages an hour or something like that. That's kind of a typical volume for a utility, maybe a bit higher than that. Now, a big storm comes in, let's say off the west coast of California or something like that. [00:07:30] And it comes in and it knocks out power to like a million customers. That system needs to immediately be able to expand into that.

Traditionally, when our clients have built these types of applications, they've built them as sort of fixed assets. And they say, "This application needs to be sized for its maximum level of utilization." Right? So you might have a couple of really big servers [00:08:00] to handle that million- and two million-person type of outage all at once even though for 99.9% of the time it's barely being used.

This is where if I were to take that application and I moved it to cloud and I don't make any changes at all, well, I'm going to reserve those instances, that same size and scale, what benefit have I gotten? Maybe I've kind of changed the way that the consumption of compute is bought or something, but my marginal savings on [00:08:30] that is going to be very limited. In fact, I would argue that the cost of actually moving, well, you probably would have trouble ever making the business case on something like that work.

The key is is to take some of those key applications, those vital systems like an OMS system as a utility, and you want to reinvent it. You want to break it into its component parts. You want to do something called service decomposition on it, so it can be maintained somewhat independently. But most importantly, you want it to actually [00:09:00] expand and contract as your business needs actually change. And that's the power of elasticity. That's the power of moving into cloud.

So if I take that application and I move it to say like a serverless technology on AWS or something, think about what I've unlocked when I've done that. Not only can I now have multiple teams working independently on adding new features like better root cause detection on when an outage occurs or faster outbound [00:09:30] calling or adding new channels like web features or mapping information that I didn't have previously. I used to have to do that in a big monolith-type architecture. It would have been really hard to do that. That's agile development and being able to do it at scale. And you've got these things. You're moving from a waterfall-type technique from something that's a monolith into something that's really broken apart that can be maintained in separate components.

But even better than that, it scales up and down. And [00:10:00] this particular example, this is a real example, we moved an application just like this for a client. They were spending $2 million a year on this application prior to its moving. After it moved, $20 a year.

Alalia Lundy: Wow.

Adam Burden: Two million dollars a year, $20 a year for the-

Alalia Lundy: Wow.

Adam Burden: ... infrastructure and the cost.

I want you to just put that in context, like that's what you get. Now, this is kind of rare that you have applications that have that kind of volume changes over short periods of time, and [00:10:30] it's really important that utilization is available. But think about that power that they've unlocked there. Now they've got new investment capacity they didn't have before. They have the ability to add new features independently inside of this application, not necessarily treat it as a big monolith. And they've unlocked better savings for their customers and more investment capacity as a result to add new features and serve them better.

So that to me is a great example of how these legacy applications [00:11:00] like this outage management system, big barrier to moving to agile. But if you treat them in the right way, if you do that surgery on it, you do that service decomposition, you move them to cloud so they can operate in truly an elastic fashion, and you've actually broken them apart into their component services parts, that you can open an entirely new level of capability and unlock tremendous value and savings at the same time.

Alalia Lundy: [00:11:30] I want to learn more. I want to hear more about the [00:12:00] conversation that you have with these organizations. You wrote an article titled “Renaissance of Custom is Here to Stay,” right?

Adam Burden: Yeah, yeah.

Alalia Lundy: These legacy systems that have not evolved with technology, what do you say to that leader, that organization that has these legacy systems that haven't evolved and they're afraid to touch them?

Adam Burden: Sure. Yeah. Well, I think you're asking about really two things there. Right? You've got the scenario where these legacy systems [00:12:30] which have been quite static and maybe you've lost kind of the tribal knowledge around how they were built. And then also you've got this renaissance of custom that's happening. Well, the thing is is that these things are actually converging. And let me explain what I mean by that. The legacy systems challenge is a really tough one. You take applications like this OMS I talked about, billing applications are often like this, big old ERP systems, that kind of stuff, typically big custom systems, [00:13:00] oftentimes...

And I worked on a system like this several years ago where it was written in Macro Assembler and it was an order-to-cash application on a mainframe written for a client. And for this client, they literally had one person that was still on their staff that knew how to write and program in Macro Assembler. It's almost an archaic language. And as our chief software engineer, I guess one of the areas of pride that I have sometimes is the number of different languages I've [00:13:30] written software in. And Macro Assembler happens to be one of them, but it's not one that they desired to continue writing software in.

Alalia Lundy: Right.

Adam Burden: So our job was to take that and actually move it to a more modern language, in this case COBOL if you can believe it. But that's what was best for them and best for this system, because it was then portable to other applications in places that do that. There are ways to unlock those applications. You can do things like static and dynamic discovery to help you learn the applications.

Now, here's an interesting fact [00:14:00] for you. Sometimes when you look at applications, not sometimes, but on average, when I look at applications legacy systems and we're looking at moving them from one platform to another, I want you to take a guess. So how much of the source code do you think actually runs on a typical legacy application, so like an OMS system or something like that? If it's two million lines of code, right, how much of that source code is still running? Let's say the application was written 30 years ago. Take a guess.

Alalia Lundy: [00:14:30] Fifteen percent.

Adam Burden: It's a little bit higher than that. It's about usually around 60 to 70%.

Alalia Lundy: Okay.

Adam Burden: But I want you to put that in the mindset though of where if you don't know anything about that application and you have to pick that up and you have to move it, right?

Alalia Lundy: Yeah.

Adam Burden: Well, you can reduce your effort by 40% or more by not moving the stuff that isn't even used. But if you don't know that application, if you haven't gone in and really discovered it, you're going to create a whole bunch of additional effort for yourself unnecessarily. So that's [00:15:00] really key. That discovery piece can teach you a lot about applications. And I think that it is really important to understand that they unlock the potential of how you can get inside of legacy applications that you really kind of have lost the expertise on to be able to understand them better. And we do this all the time for customers in Accenture. And we also show clients how to do it pretty regularly.

Now, I want you to take that and I want you to superimpose it on what's happening in this renaissance of custom. [00:15:30] Renaissance of custom is something that I've been talking about for a long time. And maybe if I keep talking about it long enough, it'll actually be true. But in this case, I really do think that we're heading there. So the renaissance of custom is really about this reawakening of “it's okay to build custom software.” I think back in the early '90s, late '80s or so, most systems, and that was when I was a junior programmer at the time, most systems were built custom. There just was not a lot of packaged software [00:16:00] to go out and buy to do business applications. So most clients looked at this and said, "We're going to build these things." And they kind of moved to this model of doing this, but it was expensive and time consuming.

And, frankly, at the time almost all software development was being done in a waterfall-type technique. You remember the Mythical Man-Month and that type of thing, right? It was a very different time and era of thinking about how do we do software development. But it was right actually for the [00:16:30] time. If you think about the constraints that were on us, we didn't have the benefit of cloud. We didn't have elastic compute to spin up lots of different software development environments simultaneously and things like that. And as a result, working in the style and structure we did really was sensible.

Now, cloud has really kind of unlocked some new capabilities there. Right? And now we're in an era where there's levels of frameworks that have abstracted some of the difficulties of writing software that really have kind of [00:17:00] made it much more focused on writing business logic as opposed to the plumbing underneath it. And by the way, that plumbing, that's exactly where most of the problems came from in writing custom software back in the day because it was hard and complex and error handling and pipes issues and connections and all kinds of like handshakes and other, passing data and other things. This was super hard stuff to get right. And when you start getting down to like register level stuff and other things, [00:17:30] you would spend ages tracking down bugs and problems.

Most software languages have abstracted away a lot of those complexities these days so that now, if I were to write something in cloud, I would probably choose to write it in a JavaScript-type framework language. I would probably write it in a serverless-type architecture. I would spend most of my time writing business logic as opposed to worrying about that underlying stuff. And that, that is what has really started to [00:18:00] I would say accelerate this rebirth of custom.

A lot of clients moved away from custom apps because they were expensive, time consuming, complex, and they didn't have the staff and expertise to be able to do that. So let's say for two decades that really went on. Now, people are looking at, well, geez, these packaged applications, they do a lot of good stuff, but these are also core to my business. I need them to move faster. I need them to evolve more quickly to the demands [00:18:30] and changes of my business. And if I can't maintain the application myself, I'm sort of dependent upon a third-party vendor who's done that.

So while I moved away from custom software into these more packaged applications in the past to reduce my risk, it's now reduced my agility, my nimbleness in being able to add new features. So now I'm at a mindset where I think that for certain types of systems in my enterprise, [00:19:00] let's say financial, human capital management-type applications, these are pretty consistent. You're not going to get a lot of competitive differentiation about writing your own payroll application. Let's face it. Right? It just kind of does what it does.

Now, what you will get a lot of competitive differentiation about is having a unique e-commerce application, having a unique customer engaging system that builds new types of relationships with your business partners, your customers, your employees. [00:19:30] So those kind of like customer experience, those customer-facing systems, those are the ones where that renaissance of custom is really taking hold. And it's happening because we've abstracted that hard stuff out. Right? We've taken that out. People focus on the business application layer. And it's faster, it's more reliable, and that's what's occurring there. And if you converge those two things, if you take those things together of, hey, we can rediscover how these legacy applications work, and it's easier [00:20:00] than ever to build custom applications, we're seeing people instead of replacing those old legacy applications with packaged systems, they're modernizing them. They're renewing them. Sometimes they're rewriting parts of them. But they're modernizing them because they can do it in a better way. And it helps them differentiate in their business. And I think it's that convergence effect that is really helping to propel this renaissance of custom out in the market.

Speaker 1: [00:20:30] Stay connected with us at agileamped.com.

Alalia Lundy: And so you mentioned that being able to do that, customize software internally, gives organizations more control, more flexibility. [00:21:00] It sounds like that's the perfect scenario for an agile implementation. But, again, [crosstalk 00:21:06]-

Adam Burden: I think it is. And it's ideal too. For many of these clients that are accustomed to more waterfall type techniques, it takes two years to get your first release of a big enterprise app delivery. I mean, business doesn't work that way anymore. Right? The speed of technology change will never be as slow again as it is today. That's a powerful thing if you really think about [00:21:30] it. And if you propel that forward and you look at it and say, "Well, if every business is a digital business or every business is a technology business and technology is moving faster than ever before, well, I sure as heck better be moving faster than a two-year delivery schedule to get that out." That in itself is driving the embracement of agile.

It's not right for every system in an enterprise. Many, many enterprises have two to three thousand applications. For some systems, they're pretty static, and they don't need to change a lot. [00:22:00] And for other systems, it doesn't make sense to write your own. But for that core set that does, those systems of engagement as some of the industry analysts refer to them, the ones that really engage with customers and business partners and otherwise - those are the ones extremely worth investing in that you can rebuild. Analytic systems, another great example, ones that give you insights or can use artificial intelligence in order to predict this. Building these in an agile fashion [00:22:30] these days is a must-do to be able to very quickly begin to deliver incremental benefits out to your customers and even maybe more important the business side of the house who's expecting that from you.

Now, that begins to get us into product-based development change and how that's actually happening. But I think it actually starts with some of those early agile projects.

Alalia Lundy: Okay. I wanted to talk about the, because you mentioned not everything is moving [00:23:00] or changing that quickly and maybe agile isn't necessarily penetrating there.

Adam Burden: I will say that agile's not right for every system. Right? You have to have the wherewithal to go back and invest in this application in a way that actually allows you to maintain it as independent components. Now, maybe for small departmental systems that can be maintained by eight, 10-person teams, fine. Right? You can do that. You can move into that.

But most big enterprise applications, they'll have [00:23:30] teams of 30, 50, 200 people actually maintaining them. Think about like some of the, like a scheduling and a freight logistics system. Right? These are at the core of some transportation companies around the world. They were built often 20 and 30

years ago in an era of scarce computing resources when waterfall techniques were de jure. Right? So they're never constructed to be built and maintained in an incremental fashion with lots of independent parts. They also [00:24:00] weren't built with a mindset of site reliability engineering or full DevOps pipeline that's actually helping you automate the software delivery process.

Going back and retrofitting on top of those systems is hard work, and it's a big investment. But for the core business applications, I feel very strongly that for most of our clients, it makes sense to go do that. Right? But there's going to be lots of other applications in their enterprise that you would sort of say, "Well, it's an invoicing system. [00:24:30] Does it really need to be maintained in more of an agile... Maybe we can do something more efficient, but does it really need to be updated and invested in to do that? What's the benefit out of that?"

So pick your battles in that space. Agile isn't for every application in an enterprise. It's for the ones that are really going to help you differentiate your business and the customers. And maybe you'll circle back eventually and go back and get all of them because it's the style and the technique for how you do software development in your enterprise. As you move [00:25:00] to more product-based operating models, most of your customers are going to want to operate in that fashion. Fine.

But for big established enterprises, don't assume that you have to do a hundred percent to be successful. You can be extraordinarily successful doing a small percentage of your legacy application estate and moving it into that type of development technique.

And by the way, I would add too that from an innovation standpoint, for most of my clients, this is where agile really takes off, right, [00:25:30] is they start with our legacy is what it is, right, and maybe we'll go back and do some modernization, like I was talking about earlier. But for our new applications, for the net new custom apps that we're going to build, we're all in. Right? We're going to do that agile from Day One. And that's where you see kind of the benefits really start to come to life, and the business sort of wakes up and says, "Oh my God, this is remarkable." Right? "Look at how fast every day we're getting new features delivered" and that type of thing. "Can we do this [00:26:00] with the rest of our legacy applications?" And it starts to pull the rest of the business along. So for those innovation things that you get out front of early, starting to demonstrate and show the benefits of agile there can be a huge way for you to create the momentum necessary to carry on the rest of your applications.

[00:26:30] I want to make one other point in legacy things that I think is really tough too is when you're moving to cloud, I think that if you move in a very nimble way, in a very agile fashion where oftentimes the mindset or [00:27:00] the pattern you embrace is often referred to as "move and then improve," this is a hard sell to a business. Right? It's a hard sell to go out to a business and say,

"Hey, listen, I need you to take like a third of your staff over the following 10 weekends to help us test this application as we move it to cloud. And, oh, by the way, once it's moved, the best that you can expect is it works the exact same as it does right now." So they're not exactly lining [00:27:30] up to say, "Oh yeah, that's for me." Right? You're going to, you want all this investment to go do that. You have to find other things that really would incent them and incite them.

And that's why getting them excited about showing them what the benefits are of incremental delivery and then saying, "Wow, if I could go apply that to my outage management system, imagine the benefits I could get." You're going to find organizations all over the spectrum, from we're 100% in on agile to those that actually [00:28:00] actively oppose it. Right? And you have to kind of pick your battles in there and understand where to go in an organization if you really want them to embrace it at scale. And my experience in the cloud area and the renaissance of custom has definitely taught me about how is it that you approach that in the different circumstances that many of our clients find themselves in, which is often radically different by industry as well.

Alalia Lundy: So I wanted to ask you, how does [00:28:30] embracing a product-based mindset help organizations to get there?

Adam Burden: I think the concept of product-based operating models and product-based mindsets, it's a bit overplayed right now out in the market to be blunt about it. I think that it is an important model to look at and say, "Listen, by bringing our [00:29:00] IT and our technologists closer together with the business side and having this concept of product owners and product sponsors and other elements working directly with our software developers to maintain services for the enterprise," I think that's a really good thing in concept. But I believe that many organizations who have looked at this and have started to embrace it have said, "It's not right for every part of the business." Right? " [00:29:30] It's not right to try and do this in every segment of our enterprise, especially for the ones that don't necessarily differentiate us out in the market."

It can actually be more expensive, to be honest, to do software development practices in this way. And you actually can get some more risk as a result too. You want to be really careful about some of that stuff. And I think that for the organizations that have embraced a product-based operating [00:30:00] model, let's say in their go to market segments of their business, they are very successful in using that. I think where others have looked at and said, "Hey, let's try and do this across our entire enterprise," they're somewhat disappointed with the results. It's more overhead. It's an additional cost. And the change management can be really hard.

But I do think that this mindset though of product-based operating models is another contributor to that catalyst, [00:30:30] the reagent, if you will, that really has helped move agile forward and embraced it more because for those customers that I have witnessed have embraced this, let's say they

of their analytics space or in their deposits business of a bank or other things and they did this in the right way, and they said, "Look, we're going to put together these teams so that there's just incredibly close communication between the business owner, the product owner, the way that the software [00:31:00] development chain, the prioritization of things is happening so there's no misunderstanding of those requirements anymore," I think that they see the real magic of that come to life.

For other parts of the business where it just, it's hard to make the case for that, it's kind of a square peg in a round hole sometimes. Maybe they'll get there eventually and that need will materialize over time. But for the customer-facing parts of the business, it's there. And I think product-based models, even though they've been a bit over-hyped in the media, [00:31:30] they serve as an important framework for how to do this and do it right.

Speaker 1: You are listening to Agile Amped.

Alalia Lundy: One [00:32:00] last thing I'd like for you to share with our audience. What gives you hope for the future?

Adam Burden: First, I think there's a ton to be optimistic about what's over the horizon. As a leader in our technology business, I see a lot of disruptive technologies out there that I think are just really, really exciting, innovations happening at the edge, new types of compute [00:32:30] technology like Quantum and others coming onto the scene and helping us solve some of the world's hardest problems. Maybe I would boil it down into the democratization of technology.

When you look at how history has unfolded, some of the biggest innovations in the world have only been affordable to the largest enterprises, the ones with the deepest pockets. Right? So think about something like Quantum computing. These days, if you [00:33:00] wanted your own quantum computer, right, you have to have billions of dollars in investments and others to even contemplate affording something like that and building your own basically to do it. But with the advent of as-a-service type of mindset and on tap computing services, you can rent a quantum computer like Braket, for example, a service from AWS, [00:33:30] for fractions of that cost. Right? And it's that effect where it allows anyone from a startup to the biggest enterprises to consume things.

What gives me optimism about all that is think about the democratization of this technology and the explosion of innovation that you get from moving from such a small segment of people able to access and afford technology to an incredibly wide community.

It's very analogous to [00:34:00] what's happened in artificial intelligence. And when you look at artificial intelligence, which had been around since the '50s, there really hadn't been a huge amount of innovation in that space until let's say 2000, 2005 or so. What happened around then? What occurred then that really started to explode that? Well, cloud computing came onto the scene, and it opened up AI to legions of people who would otherwise never have been able to train their models to do things. And it wasn't [00:34:30] just captive within big enterprises.

So for me, what gives me hope about the future is I think that technology will be democratized for everyone and it will give the opportunity for new innovations that are going to help us solve the world's hardest problems, whether that's challenges with climate change, whether it's challenges with education and healthcare, or its challenges with making sure that we're taking good care of our elderly and our sick population. I feel that [00:35:00] across the board, these opportunities are at our fingertips now, and it's because of this democratization of technology. So that gives me a lot of optimism about the future.

Alalia Lundy: I hear a sequel. Adam, I look forward to speaking with you again. This has not been enough time. But I really appreciate your time that you've spent with me today.

Adam Burden: You're welcome. I enjoyed so much coming on Agile Amped, and I will come back anytime.

Speaker 1: [00:35:30] Thanks for listening to this episode of Agile Amped. If you learn something new, please tell a friend, coworker, or client about this podcast. For more inspiring conversations, subscribe to Agile Amped on your favorite podcast app. If you have an idea for a topic or feedback on an episode, email us [00:36:00] at agileamped@accenture.com.