EMBRACING TECH IN FINANCIAL SERVICES EPISODE: DEMYSTIFYING THE DEVOPS BUZZWORD

TRANSCRIPT

Host: Tim Broome, Technology Advisory Practice Lead, Financial Services, Accenture Australia and New Zealand
Guests: Ajay Nair, DevOps Principal Director, Accenture Australia, and Harshu Deshpande, Emerging Technology & Liquid Studio Lead, Accenture Australia and New Zealand

Tim: DevOps, is it a tool, is it a process? Or are we actually talking about a culture? Join me and my expert guests as we discuss where DevOps is at right now and where we see it going in the future.

Presenter: Welcome to Embracing Technology in Financial Services, a podcast brought to you by Accenture. In this 16-part series, we will hear from experts to uncover the latest in technology and trends in Financial Services. Now, here's your host, our practice lead, Tim Broome.

Tim: I'm really looking forward to our conversation today, I'm here with Ajay Nair, who I've been told is our not our DevOps lead. Hello, Ajay.

Ajay: Hello, Tim, that is absolutely right. I am not our DevOps lead, but I look after our DevOps practice. By looking after, I see myself as an enabler.

Tim: Excellent. And I'm also here with Harshu Deshpande, who is our emerging technology and Liquid Studios lead. And you're here for a third time, how are you doing?

Harshu: Great, thanks for having me back. You got to be careful, I might be looking at hosting this show soon.

Tim: This is my seat, and I'm keeping it. We're like one of those couples who is not yet in a relationship, we're not quite sure.

Harshu: Oh, yeah.

Tim: All right, down to business. So, we're here to talk about DevOps. So, Ajay, we've not worked together for a while, what's going on with you?

Ajay: What's going on with me? Look, I've been around various different blocks since we last spoke together - to 2016, yeah, 2016-ish - been around a few different industries solving this wonderful philosophy, a journey called DevOps. Outside work, I've been doing a few concerts, and I think we have one coming up, Tim. Where Metallica-

Tim: We might have one on the horizon, yeah.

Ajay: That's right. So, yeah, just keeping busy and...

Tim: Brilliant. How about you, Harshu?

Harshu: It's conference season, so been doing lots of summits and conferences, doing talks. Just did one yesterday on the Cloud Strategy Summit, talking about how organisations should look at scaling their cloud journey as they look to go beyond just lift and shift workloads to doing mass migration of workloads into the cloud.

Tim: Yeah, see an endless stream of LinkedIn updates of you speaking places.

Harshu: Yes, absolutely.

Tim: That's good stuff. Okay, so topic for today, DevOps. Now, I think we've all been in and around DevOps for many years now. I - probably incorrectly - a couple of years ago thought that it was here, it had landed, everybody was fully adopted, and I think I've come to realise that, that's a long way off the mark. How do you see things, Ajay?

Ajay: For me, I would love to come in and sit in the shoes that you wore, and what you saw, because it's so far away from reality. So, I'm quite keen to understand why you thought the problem was solved. Similar to how Harshu, back in the day, told me the agile problem was solved, "Everyone's agile now." So, the way I see it, people have been on the journey, right? And at varying different parts of the journey, and just like anything - like the hype curve - people get onto something, everyone assumes you're sitting in a nice car, going to the right place, and then things break down, and... there's a lot of challenges right now, so people are in a place where they need more steer than ever before, because they've started on a journey which they don't necessarily fully understand how to take through to the end, or... or there is no end, in this case. That's what makes it harder.

Harshu: I think what I've found interesting is IT... like, the traditional way that we were doing IT has very much changed now, and there's so many different factors that the IT departments have to face with, right? There's new ways of working on the agile, lean practices, there's DevOps, there's cloud, and then there's all this lightweight architecture and modulisation of your architecture, and digital decoupling. And it almost feels like they're stuck trying to meet all the different demands and getting confused around how those things interplay and work together. And are you seeing... are they trying to solve this all in one go, holistically, or are there still silos of organisations or parts of the organisation that are doing this all individually and separately?

Ajay: It's a combination of both, right? For me, I'd almost want to steer away from the word "DevOps," because I think what's happened and where it's taken the industry, where it is today, the confusion is largely because people have been trying to solve something without fully understanding the system. So, I talk to a lot of my clients about "You're a system, your organisation's a system, apply systems thinking to this." So, the problem you're trying to solve really is, how do we make the system more efficient? Whether you do that through DevOps, the cloud, or... they're all means to an end, and therefore, focusing on the system fundamentally and asking questions about why does the system operate the way it does? Why does the system allow me to deploy untested and go into production? That's what I should be solving for. Now whether I do that through getting teams together, so people or process, or tech implementations. They're all part of the same goal, fundamentally, as far as I'm concerned. The buzzwords everywhere has obviously made... you're right, Harshu, where people are saying, "I want to do a bit of that, and a bit of that, and a bit of that," and they're quite confused as to where they currently are.

Tim: And when you're talking about confusion, do you think there's... we've got DevOps, and then we've got DevSecOps, then we've got BizDevOps, and then various combinations of all of that.

Ajay: And don't forget SRE (Site Reliability Engineer), in there, as well, somewhere.

Tim: Site reliability engineers?

Harshu: Absolutely.

Tim: So, do you think that is making things more complicated, or making it harder for people to figure out, "Well actually, what am I supposed to be doing? What's the right answer to my problem?"

Ajay: To me, there's two sides of this, of buzzword bingo, right? So, on one side, it's terrible because vendors have jumped on the bandwagon and said, "Yeah, here, you can get a DevSecOps tool, or a DevOps tool." Like it's a tool, right? And therefore, a lot of customers feel like, "Oh, by doing that, I might be on the journey, or I'm solving the problem." But they quite often don't end up doing that. So, the risk you run with buzzwords is people... it depends on how you interpret them, it depends on what you do with it. But on the contrary, buzzwords are a reflection of what's missing in the DevOps journey, so DevSecOps is a fantastic example, right? No one's ever said, "Security should not be embedded as part of your DevOps journey," or they assume that it is. But then, as people consumed what they presumed to be DevOps - or in their definition - quite often security felt left out. So, the security committee came in, and it's a great... that's a good example of why the buzzword is awesome, because all of a sudden, it gets security a place in this thing that people are trying to do. So, it forces, or brings people of the same ilk together. You go to a DevOps conference - again, that's a fantastic outcome of a buzzword - because now you have people in varying different places in their journey, all coming together in one place to listen to how different people have experienced the same thing. And it broadens their horizons, and their thought processes and what they need to think about. And I think that is a good reason why the buzzword thing... I don't have a problem with it.

What people need to be careful about is how you... don't get caught up with the buzzword. Go back to the system, the buzzword gives you some sort of meaning, or something to hold onto, which is great. But you've got to distance yourself, as well.

Tim: Do you think that's an evolution of probably where all this started off, probably 15 years ago? "I'm going to automate a piece of code deployment." And that was a segmented piece of the puzzle. And then we've grown out... though, separate to that, you'd have a group say, "Well, I'm automating testing." Okay, well actually, but that's all part of this code migration process. Security, for better or for worse, is one of those things which is often left ‘til late in the process anyway, probably not surprising that it's late to this party.

Ajay: Well look, that's a... well, I think it was never meant to be late to this party. I just think now it's forced into the thing, right? So. By talking about DevSecOps, we're now actually saying that "No, they very much have the same seat on the table," and that's what that's done. As far as I was concerned, that should have always been the case. Like, DevSecOps fundamentally focuses on, "How do you secure the application? How do you secure the infrastructure, and how do you secure the pipeline itself?"

Harshu: I think I found this interesting yesterday - you were at the same conference as I was - and they had the Cloud Infrastructure Summit, and Cloud Strategy Summit right next door to the DevOps Summit, and one of the key elements was, how do you achieve scalability of your cloud infrastructure, given all the security concerns and things?" And really, the answer there was DevSecOps, right, is enabling things like policy-based security and compliance, so that you allow the teams flexibility and agility to innovate and use cloud, or use it in a simpler consumable manner.

Ajay: Correct, and do it a lot earlier in the lifecycle as well, as opposed to being this one check at the start to see that you're compliant, and then some tests at the end that they've actually gotten... "How do I do it in a static way, in a dynamic way?" And I think it's brought it closer to the developer than it previously was.

Harshu: But do you think then, therefore, we'll start adding more words? Will it be DevSecNetworkOps, and all the other elements that are also required to enable agility? Will the words just keep on expanding?

Ajay: It's interesting you say NetworkOps, because I think... I've been in a few recent conversations where telcos are looking at, how do they apply DevOps in the network space? Because if you think of the advent of cloud, and how infrastructure became... well, is like code now. Software-defined networks do somewhat the same thing. So, you're kind of dealing with network devices that you can deploy or configure like code, and therefore they are asking the same questions. Now, whether it becomes DevNetSecNetOps, or whether you just say DevOps for networks, for me it's the same thing. Look, all this while we've been talking about the technology underneath, but not talking about how these organisations need to think differently and work together.

Tim: Yeah, and that's... one of the questions I got, was... I said, we've talked about technology, we've talked about tools, is DevOps a technology? Is it a process? Is it a culture? To me, underneath it is... there's a culture. If people have the right mindset, it almost happens anyway. It's when people are given tools and told, "This is the way it works," that we often see failures and struggles.

Ajay: 100% agree. It is a mindset, and I think it's a combination of different things, but it is a mindset, fundamentally. Having the mindset alone is not good enough, you still need the process, and the organisation to be structured in a way that allows you to leverage the mindset and actually use it as a catalyst. Which is why tools process, the organisation, the skills you have in your organisations, the talent strategies - all that stuff matters on this journey. But yeah, fundamentally, if you get all that right and you don't have the mindset or the culture shift, then we have enough data to suggest that people don't achieve the outcomes that they set out to achieve. That's not saying they don't achieve positive outcomes, it just means what they envisioned for the organisation in terms of reducing cycle time for the business, didn't hit the mark because the mindset wasn't there.

Harshu: And I find it interesting that in a microservices design, they talk about this concept of Conway's Law, right? So, a system that's designed reflects the organisation that built it, right? And DevOps - everyone talks about culture and tools and processes, but then everyone forgets about the organisation that sits behind it, and the hierarchy and structures. And fundamentally, if you still have Dev separate to Ops and they have different KPIs, you've not met the fundamental tenet of what you set out to do in DevOps. And, so many times I'd go around and see that the basic fundamental thing that's missing is just the org change that's required, and that's always the last thing that comes.

Ajay: And it's a tough thing to do, because you're changing something that has been set out for, in some organisations, for hundreds of years, right? So that's why it's also a tough thing. Actually, yesterday in the conference that you spoke at, Harshu, I went and attended one of the DevOps talks by Nigel Dalton – he was the ex-CIO of Realestate, I think. It was such a good talk, and he talked about how, in REA, they actually had an example of getting the sales and marketing guys to sit down with the operations engineers, or the developers. And for the first time, you have business - and I'm going back to your buzzword, BizDevOps or something that you said. It's actually getting them working together, because then it was so clear for the sales and marketing guy, that by virtue of me doing X, Y, and Z, or selling X, Y, and Z as a product, it actually goes through these 64 systems.

Ajay: He had no idea of the complexity, and therefore, together they changed the way they operate the product, or they changed the way they think about things from a business angle - how easy that can make everything else, and vice versa. It gives an operations person or a developer - or anyone in that system - a better understanding of what they're building in the first place. It's a tough thing, though, because you're changing how they... what they've done for years and years and years, and I think that's part of the journey.

Tim: And from a, probably the more traditional DevOps lens, I'm seeing organisations adopting in a team basis, and then trying to figure out, "How do we centralise it at an enterprise level?" And it seems to consistently struggle hugely. What's your experience on that side of things?

Ajay: To me, having a DevOps team is a complete anti-pattern to... you almost run the risk of creating another silo, which is the DevOps team.

Tim: Yeah, totally agree.

Ajay: Right? And what I think again, going back to systems thinking... we have a platform team underneath, right? So, if you think of an organisation where you have a platform team that looks after building things that your application teams consume.

Harshu: Your key enablers.

Ajay: Your key enablers, right. So, it could be the tool link, could be the infrastructure, the cloud - all that stuff sits in one place. And their job is to make sure that my consumers of this service can consume things as easily as possible, and I think that should be the sole motive of that team. So, calling it a DevOps team is a bit of a weird one, because it shouldn't be. You don't want a bottleneck within that team. People within each delivery team, that the teams themselves are consuming, should really be DevOps teams, because that's when you have cross functional people with the right skills that can take code from... or even product ownership, for that matter, where you have business working with those teams to actually take that from idea to inception, through to, all the way, into production. I think that's the key thing, as opposed to creating another silo.

Harshu: All right. And how do you find the whole platforms team? In effect, when you started looking at things like standardisation of platforms across an organisation, do you find that that stifles innovation, in some cases? Because if there's a new tool or a new product that comes out that the team wants to use, but the platform team hasn't approved it or hasn't enabled it, do you find that there's a bit of a step where they're kind of confused as to whether they should use the new tool, or standardise in that battle?

Ajay: Not if you have the people with the right mindset in that platform team, right? Because there's never going to be one platform, let's face it, for the next 100 years. It's not like one organisation's going to stick to a certain set of tools or a platform. What I think should happen is... there was centralisation and decentralisation. Our understanding is that it's extremely tricky and complex, but it's extremely important. I've been in an organisation where they've said, "You know what? We're just going to go decentralise everything. Everyone, pick their own thing and run with it." Yes, it's fast, but the amount of confusion it causes in trying to bring work together to understand where something's at is ridiculous. And then, let alone, the license costs and what not. So, if you think about it from the other angle... if you centralise everything, it actually constricts people, or forces them down a certain path. I think it should be quite open, where we will centralise in certain points, and decentralise in certain points. And where we have centralised, if there's better suggestions, those application teams should feed that back into the platform team, have a mechanism where they can tweak it, learn, and try something new. It doesn't have to be one way or another.

Tim: So, actually, I just want to loop back a little bit. We talked around the organisational challenges in and around that, and I understand that you've got a capability that you do called a DevOps Simulator, that I think tries to challenge and work through some of these challenges. How does it work?

Ajay: Right, so we call it the DevOps Simulation, it's actually... so there's a company called G2G3, who built the simulations. So, they're not technical in the sense that you're going to need a laptop to run this. You can think of it as a change management exercise. When I say change management, I mean human change management. So, when you talk about being on the DevOps journey, for example, what it means to work together in teams - I think that's what it's trying to simulate.

So, it actually gets a bunch of people together in varying factions of your organisation in a room, and we simulate what it is to work together, and the advantages of working together. You might think it's super obvious what the advantages are, working together, but we go through different simulations. So, like a simulation where... and it's all gamified. So, it's quite real, it's quite intense. Like, you still have...

Tim: So, what's an example of a scenario?

Ajay: So, a scenario is, you will have a business user, you will have some developers, you'll have two development teams, you might have a test team, and you'll have service management. And this exercise basically simulates a real-life environment where the business wants something developed, there's people running helter skelter because the dev team is in one corner, the test team's in another corner, and the dev team then picks it up. They actually solved some Tetris-like puzzles, that's what development is, it's not a real... writing Java code or anything like that. And then you've got to get it tested, and all the way out through the door. In the middle, there's incidents coming and hitting you, and that's causing chaos, and then it gets the group... and after that, you would have produced, say, a 30% output, but then it gets a group together to think about "How would they do things differently?" So, the team then says, "You know what, actually? I would go position myself with that team, because we kind of need to talk together."

Tim: Oh, yeah.

Ajay: So, it kind of gets you to understand DevOps as it truly should be. Ignoring the tech - the tech and other stuff are all things that enable you to do what you set out to achieve - this kind of forces you to think about what you're setting out to achieve, and it forces you to think about breaking organisational silos. So, it's very, very effective. The reason we do that is because - I saw this when I went for the DevOps Summit in Vegas, they were running one of the simulations there, and I was invited to look at it. I sat there and looked at it and said, "This is so cool," so that's why we said, "Let's pilot it here." So, we're now trained in it so where we can run this for our clients.

Tim: Okay, and so it's completely divorcing us from the tools, the technology, and just focusing on, "How do people work better together?"

Ajay: That's absolutely right. And it's a lot of fun, actually. Being part of it was super fun.

Tim: Awesome. Really interesting. Okay, so I think we've talked through a range of different challenges and opportunities out there. Let's say you're walking into a bank, and they've got challenges, not sure what to do next. How would you start, and where do you focus?

Ajay: So, for me, my first core focus is to understand the purpose of... when they say they want to go down a DevOps transformation - or whatever it is they call it - why are they trying to do that? Because I really need to understand the why. It can't be "I need more tools or more automation," but it's what's-

Harshu: Or 50 deployments a day-

Ajay: Or 50 deployments a day, or 100% automation. It is, "what are you fundamentally trying to achieve for your business? Is it innovation, is it cycle time, etc, etc?" So, nailing down the why is very important for me. I think once we get a handle of that, that enables you to figure out how you approach it. One of the most important things I would do, personally, is not boil the ocean. It's not a race. So, for me, it's very important to break it down, pick pockets, application teams, both across, and friendly areas in the enterprise where you can start doing some things, and learn from it, and then do the next bit. So, a transformation doesn't have to be one big thing, it is many smaller transformations.

Tim: Yeah, and I think we've learned that big transformations aren't the way forwards.

Ajay: That's right. It just doesn't work. So, that's what I would do, I would find a cluster of applications where I can get a good business outcome by virtue of getting them to a better... because if I go and make one thing really fast, but that's now one piece amongst 20 other things, I'm still not fundamentally exciting business to say, "Hey, well, we were slow then, we still are slow. Although hey, guess what? This particular thing works really fast." It's of no use, it's a link in a chain.

Ajay: So, I'd rather pick an area where it's a smaller subset of things where I can get some value, implement stuff both across people, technology, and look at all those things together. Not just the tech part. How do I get the people working together, run some simulations?

Tim: Yep, yep.

Ajay: Get them working in a certain way.

Harshu: Picking up on that point around change management - especially the human change management - I find, you know, we're pretty much asking people to move across two different dimensions really quickly. We're saying, "By the way, all the stuff that you've been doing for the last 15, 20 years is all old and bad, and here's all this new, good stuff around emerging tech." And we're also saying, "All the ways that you've been working for the last 15 years is now wrong, and here's a new way." So, you're pulling people across two axes at the same time, and that's a very fundamentally different way of doing things for them. And that requires a fair amount of care, to be able to take those people on that journey to help them adapt and change as times move on.

Ajay: And you can only do that if you're doing something that is constrained and small, right? Because you're absolutely right, I think the question leadership should be asking is, "How do I build a system that allows people to feel safe to do exactly that?" Not just make people do it, how do I actually create a system? So, what is it about the current system that stops us from doing it? And therefore, let's address that bit in this area. I think that's how I would start, and that involves tech as well as process and tools, and... but people are at the core of it-

Tim: Yeah, I think that's a really great point. I mean, that connects into - Harshu, what we were talking with Andrew the other week - this disruption. Behind disruption, there's people, and unless you get the people to understand why we're doing it, how we're doing it, what's in it for them, you're not going to get adoption.

Harshu: Yeah, absolutely.

Tim: It's been great talking to you both, it's been a great conversation. I can certainly see that there's certainly a lot of evolutions still to happen in this area, and I'm really looking forward to getting involved in one of those DevOps simulations at some point in time. I think that would be very rewarding. Look, great to speak to you. Thanks very much, Ajay.

Ajay: No worries, Tim, thanks for having us.

Tim: And Harshu.

Harshu: Thanks, Tim. Thanks for having me back again.

Tim: Pleasure, thanks very much. Goodbye. If you have any questions about today's podcast, and want to get in touch with me, please email tim.broome@accenture.com. For information on all our podcasts, please visit accenture.com/embracingtech. See you next time.

Presenter: You've been listening to Embracing Technology in Financial Services. You can hear the entire series on the Accenture Vision App and SoundCloud by visiting accenture.com/embracingtech. For more information on all our podcasts, please visit accenture.com.

[End of recording.]