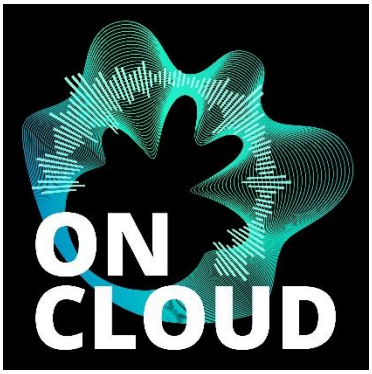

The Deloitte On Cloud Podcast

Mike Kavis, Chief Cloud Architect, Deloitte Consulting LLP

Title: **Calming AI chaos: DevOps leader John Willis on how SRE can help constrain shadow AI**

Description: This episode—part two of a three-part series with DevOps leader John Willis—focuses on “shadow AI.” Similar to the more-familiar “shadow IT,” shadow AI is the unconstrained expansion of AI that creates operational chaos and adds to technical debt. It occurs when organizations fear embracing AI but workers use it anyway—though without governance. To fix shadow AI, John says, companies should embrace AI, constrain it with site reliability engineering, and leverage it to meet business goals.

Duration: **00:17:06**

Mike Kavis:
Hey, everyone. Welcome back to the On Cloud podcast, where we get real about cloud technology. We discuss all the hot topics around cloud computing with the people in the field who do the work every day. I'm Mike Kavis, your host, Chief Cloud Architect over at Deloitte, and I am joining with John Willis for part two of a series of discussions with John on various topics. And John is the author of 12 books, starting number 13 and is a legend in a DevOps and cloud space and I think will soon be a legend in the AI space.

So, this is the second session. We talked a lot about your book on the first one, but the focus today. But first, welcome to the show. Real quick, tell us what you've been up to.

John Willis:
Yeah, sure. Yeah, so people who listened to the episode about my sort of work with DevOps and Deming and all that, about two years ago, somebody showed me some of these derivative, early versions that were sort of tools that wrapped around their API, and I went, "Oh, my goodness, this is insane." So, I asked some questions about Deming and it gave me a paragraph, and I was like, "Oh, my God, this is going to change my life for research."

And then, all of a sudden what happened was after my friend demoed it, I went home and I started – the second paragraph was like, "Oh, my God, this is terrible." Third paragraph was even worse, this whole concept of hallucinations. So, I learned enough about how this thing worked, that – while was it seeming so fascinating but then why did it go off the rails so quickly? If you asked about like Abraham Lincoln, GPT-3, it give you four pages of accuracy. But if you asked about somebody named Mark Burgess, who basically invented the whole concept of infrastructure as code, it would give you a couple of sentences that were accurate but then it would start doing some weird things. It would say, "He got a degree from Cambridge University." No, he didn't. He got a degree from University of London. But he had a physics degree, so it took the most probable answer. I understand statistics pretty well and I started really gluing together the idea of probabilistic things and how these engines work, and it makes really good guesses.

So, anyway, paid attention to it. Fast forward a year and a half ago, I was introduced to the concept of a vector database. And a friend showed me a demo of what he had done, where he kind of, not eliminated hallucinations, but got incredible accuracy from a very specific corpus of data. And I'm like, "Okay, this is what I wanted in the first place because now I can load in research papers and all this stuff, and now I can do research and have high accurate answers." So, I sort of dropped everything. I picked up some clients. One of my clients is MongoDB. They have a vector database. So, I started learning and building workshops, and I have some workshops I've been running for enterprises, trying to teach enterprises how to deal with all this stuff, it became really clear to me that the complexity of this technology is far worse--better or worse, depending on whether the glass is half full or half empty from your view—than what cloud was.

And so, I think the interesting thing that you picked up on is the shadow AI. And having a background in operations and infrastructure, I think about what most of us do who sort of title ourselves either sysadmin, DevOps, or just general infrastructure and architecture: We're protectors. And if we're kind of

good at what we do, we learn from our prior mistakes. And I started thinking a lot like, what were the mistakes we made with shadow IT? And that just opened up a world of, "Okay, now I'm on a mission because it isn't even about making money."

Now, I did say earlier I'm writing a book about the history of AI, so that couples really well. That's sort of my evening time work, which helps me better understand how I explain to executives. So, we saw shadow IT with cloud. And what happened? What happens every time there's these massive transformation opportunities? With the internet some of us who are really old enough to know, corporations tried to block the internet. They tried to block open source: "You can't run any open source."

When Web services started, you could only use like WebSphere or WebLogic and what happened? People just started grabbing Apache and throwing up websites and. And then the cloud was this sort of real sort of, "Oh, my God." We told people they couldn't use cloud and then what happened. We've taken 10 years to clean up that mess. Probably more.

Mike Kavis:
Yeah, I always say a multi-cloud strategy is not a strategy; it's a reality, because all of a sudden you have 10 clouds. It wasn't planned. So, the multi-cloud strategy is usually a reaction to what happened.

John Willis:
So, it's a great segue into why do we get those reactions on every one of those massive change shifts? Because we, as leaders or organizations—how we decide to deal with these transformations, we either react with "No, the answer is no!" – block, stand in front of the door – "You don't get in," or we ignore, sort of put our hands over our eyes and say, "Yeah, it can't get that bad." Or we embrace. Unfortunately, more often than not, we don't embrace.

And so, what happens when we block? When we block, it's classic Goldratt. We're going to get workarounds. You've got misaligned incentives in every organization. One group is telling you, "You can't use this." The other group is telling you, "You have to get this done by this date. And you have to do it efficiently at low cost." You either don't do it, you don't do your job and get fired, or you create a workaround that sort of seems plausible to say, "Oh, well, I did this, this and this. You wanted me to do that?" Sorry. Or we ignore where the mandate is, or there is no mandate. Leadership just says, "Ignore." And what happens there? We get the, "Do now, ask forgiveness later," syndrome, "Well, you didn't tell me I couldn't do it. That wasn't clear that I couldn't. So, I did it."

Or we embrace--unfortunately, it doesn't happen enough—is where we start to learn and we start to get feedback and we start to learn how to adapt. Because when we block, we don't earn anything. When we ignore, we don't get any global learning because there's little pockets that may share, "Oh, how'd you do that?" "We did –" "Oh, wow. Let me – show me." "Can you send me that?" And that gets worse because it becomes this massive sprawl.

So, when I'm looking at all this stuff when related to the Generative AI, block is not going to work. Did we learn anything? It isn't working. I have some data points here. Ignore is potentially as dangerous. And so, if we've learned anything, let's embrace. Let's stop this nonsense about, "We can't do it." Embrace doesn't mean we do everything. Embrace is we take a strategy of learning and experimentation. If you listen to my last podcast with Mike about the Deming's idea about theory of knowledge and scientific method and, "Let's come up with theories," like "If we do this, we think this will happen, but let's test to make sure this and this doesn't happen." And so, the light bulb went off about how much worse this is. This is probably – and I've got data to back this up—three orders of magnitude more complex, higher technical debt than shadow IT, i.e., cloud.

And here's an example. Back in cloud days, if you were backed with 100,000 employees, I would say best case you had five percent, 5,000 people using cloud because it was very IT-centric. In fact, to use cloud back in the early days there was no website. You had to use code and APIs. So, the funnel of the sort of the massive technical debt that we incurred over 10 years was a very narrow funnel.

And so, all of that debt that came out of like shadow IT was probably five percent, if I'm being gracious, it was five percent of a 100 percent organization. Now, with generative AI we've taken the programming mostly out of the picture. It's moved from program to model where the model is sort of the program. Now, I was estimating that 70 percent of an organization, but you've got the five percent in shadow IT, and now we're going to have 70 percent in shadow AI. And by the way, this is a much more complex domain in the cloud. This is all probabilistic, it's statistical models. Most of the code is not written by large scale industries; it's written by academia. We can go on and on and on about that. But then LinkedIn put out a survey just recently, a work trend survey, and they confirmed from their survey that 75 percent of knowledge workers are using generative AI and 78 percent of them are doing what they call BYO AI, bring your own AI.

So, I'm sort of on a mission. I'm working on a paper right now through Gene's organization called, "Dear CIO," with a bunch of sort of founders of DevOps; we're like, "Hey, CIO, let's learn the lesson this time." Because the thing we also are seeing is a lot of these CEOs now are hiring chief AI officers, because, "I've got to get somebody who knows this; we've got to stay competitive." And I don't disagree with that. I've heard sort of CIOs tell me it's do or die. But all of that debt, the GRC, the risk, the brand protection, the things that we do really well in infrastructure and operations will get thrown out of the window if you sort of silo and proxy all that to somebody who has no idea how to run a bank, has no idea to go through a full scale audit for a financial institution, has never gotten a severe enforcement action, and actually add the complexity of code, the fact that we're actually making sort of the same mistakes over again.

And the "Dear CIO" is like, "Dear CIO, if you ignore or block, we know what that's going to do." But if you sort of ignore or let the responsibility of protecting your brand, protecting risk, all those things, to somebody who doesn't really understand that domain because you got to move fast and the industry's telling you need to hire a chief AI officer, let me tell you what it's going to look like for you in a year from now. You're going to probably have 400 different AI vendors. They're all going to be using different temporal versions of their different point in times, like they implemented LinkChain at this point; these people, they did this. And you're going to have an in-house mess .You're going to have 30 versions of 10 different orchestration engines. You're going to have 10 versions of 25 different vector databases. You're going to have a mixture of expert agents run amok because you didn't sit down and try to learn and embrace and at least come up with some architectural designs."

The mistake we always make is, "Oh, cloud, that's new. That has nothing to do with IT. Let some cloud folk figure it out." "Data. Oh, you don't need data operations. It's big data. It's big data. It's magic. Create data lakes." Okay, where did all the data come from? How did it get there? How do we survive an audit to explain why we gave this answer? So, this is where my head is.

Mike Kavis:
So, this is a huge challenge. Because the reaction to that is to lock it down to the point where it's unusable or unpleasant to use. And that just creates more shadow IT. So, it's like, you have this balance. You don't want people to go willy-nilly and create all these threats but you don't want to lock it down to the point where it's like, "I'm just going to go use my own thing because I can't get work done with all these controls." So how do you balance that?

John Willis:
I think one of the things I talk a lot about in this paper that we're writing in and my presentations that I've been giving and I will continue to give, they're called, "Dear CIO". Everything goes back to Deming. If you follow some of the people who sort of learn from Deming – and again, Deming learned from a lot, so it's always the shoulder of giants, but it becomes really clear that constraints can create flow. It's not an obvious idea that the more you constrain something, the more flow you get. Think about like highways and roads and queuing theory and all that stuff. And you don't even have to go that far.

So, the thing I think about a lot is how have we successfully done constraints for cloud or cloud-native or just all the things, the new sort of technology stacks? I would say that SRE, site reliability engineering, has been one of the most successful things, and at the core of SRE is creating constraints. So, if you read some of the original papers by Tom Limoncelli, and then you have the original SRE book, Google basically – so, SRE comes from Google, basically. At Google they created constraints around technologies. You could only have two kernels. They were very limited to the amount of sort of pipeline types you could create. If you were ever going to use like language frameworks, they kept it down to sort of one or two.

In fact, it was this sort of idea, and I think it was Tom Limoncelli's book about this idea, was a team in Google that would literally go around and say, "Hey, why are you running three kernels? You need to be on these two." "Okay, I'll take care of it." "No, no, I'll stay in your office until you're done." IAnd then think about SRE. A proper SRE package is, "I've got an application. I don't have to have it SRE-managed in Google." A well-formed SRE in any bank or anything should operate like this. But to get SRE managed, you have to put some constraints on what you do. One, I need you to instrument and show me how I can manage it. Two, these are the two types of pipelines that we use for software supply chain. These are the two observability tools that you can use. And we create these constraints. And what you find is people come in resistant. "Oh, no, I've got to have this product. The whole world will end." All right, then we don't support it for SRE. And then look at your costs for supporting it. "Maybe I'll try of those two. Oh, actually it works better than the one I had."

So, I think there's a lot of lessons learned of what we did with cloud native, cloud, how we run large-scale IT today. We've created these constraints. And one of the ways we've done that is through SRE and the idea that in order to be SRE-managed that you're going to have to buy some constraints because we can't manage it at scale. Take this idea of a vector database. I've been telling CIOs, "You're going to blink and next year you're going to have 30 or 40 vector databases." And think about how antithetical that is to the way operations, infrastructure operations should work. One, pick two, create an incredible relationship with those vendors. Depending on how big you are you can get kid glove handling depending on your spend.

Mike Kavis:
John, thanks so much for your insight. We are going to wrap up today's 'part two' session and talk to you all in two weeks where we will be finishing up our 3-part series! I hope you enjoyed part two with John Willis. Make sure to like us, leave a review, and subscribe. You can also check out previous episodes where you listen to your favorite podcasts. I'm always on Twitter, madgreek65. Or you could hit me up on e-mail, mkavis@Deloitte.com. Feel free to ping me and let us know what topics you want to listen to. And that's it for our show today with John. And until next time, thanks for listening to On Cloud

Operator:
This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor.

Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

About Deloitte

As used in this podcast, “Deloitte” means Deloitte Consulting LLP, a subsidiary of Deloitte LLP. Please see www.deloitte.com/us/about for a detailed description of our legal structure. Certain services may not be available to attest clients under the rules and regulations of public accounting.

Copyright © 2024 Deloitte Development LLC. All rights reserved.

Z