Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

Title: **The Seven Deadly Diseases of DevOps**

Description: DevOps has been a boon to software development, marrying development with operations and stressing shared responsibility and accountability. Now, forward-thinking companies are realizing that security should be part of the DevOps mindset—thus the term DevSecOps. The name reflects the long-neglected, but tremendous, opportunity for collaboration between DevOps and security. However, there are pitfalls to watch out for in integrating security with established DevOps methods. In this week's episode, Mike Kavis and guest John Willis, Vice President of DevOps and Digital Practices at SJ Technologies, dive deep into DevSecOps. John shares his insights on the "Seven Deadly Diseases" of DevOps—the worst of which is ignoring security—and provides tips for companies to help avoid them and finally make security a first-class citizen in the DevOps environment.
Disclaimer: As referenced in this podcast, "Amazon" refers to AWS (Amazon Web Services).

Duration: 00:30:23

Operator:
The views, thoughts and opinions expressed by speakers or guests on this podcast belong solely to them, and do not necessarily reflect those of the hosts, the moderators, or Deloitte. Welcome to Architecting the Cloud, part of the On Cloud Podcast, where we get real about Cloud Technology, what works, what doesn't and why. Now here is your host Mike Kavis.

Mike Kavis:
Welcome to Deloitte's Architecting the Cloud Podcast, I'm your host Mike Kavis and I'm here with John Willis, who is one of the four faces on the Mount Rushmore of DevOps.

Mike Kavis:
So, John, welcome to the show. We go way back, but maybe our listeners don't know you, so tell us a little bit about your background real brief, what you're up to, and we'll get into the meat of this conversation today.

John Willis:
Yeah, thanks, Mike. So, you and I go back a pretty long way too, right. So, my name's John Willis. I go by the handle Botchagaloupe, maybe you get it in the show notes. I'm easy to find, so I'll give you the really short version. I'm pretty much 40 years – this year, 40 years in this industry. Gone through numerous waves. Going back about twelve years ago, I got into kind of distributed computing in kind of its latest form, if you will, starting with working for Canonical, one of the first private clouds. I think that's about the time I met you, Mike. And then went on to Chef, was early on at Chef, so spent a lot of time kind of infrastructure as coding, if you will. And then built another company as a principal that we sold to Dell, which was multi-cloud management. And then my last startup was about three years ago. I sold a company to Docker. Spent two years with Docker. And now I'm back just kind of freelancing, doing independent work, mostly working for CIOs right now, so doing kind of meta-skill transformation, helping companies understand how to get past the kind of the scalable brick wall of DevOps.

Mike Kavis:
Yep, we do go back a way. First question, and you know, we love buzzword bingo and there's like twenty different DevOps buzzwords. One of them is DevSecOps, which I don't really like the word, but after talking to you, I accept it. But what we're here to talk about today is security in DevOps and the intersection of them, and security finally becoming a first-class citizen in Dev shops. And you're doing a lot of work in this area. You've been talking a lot about DevSecOps, so just tell us in general your thoughts on the term, and then what are companies doing now.

John Willis:
Yeah, I think you hit the major point of what this term is doing, right? And a lot of times when I'm speaking on either something about cyber or introduction to DevSecOps, or I do this seven deadly diseases of DevOps, which the worst deadly disease is security compliance theater. But one of the things I'll jokingly say – actually I'm kind of serious and joking – I'll say if after my presentation, everybody that wants to argue about the name, go ahead and meet at the right-hand side of the room, because I'll be in the left-hand side of the room trying to work on the problems and moving the ball forward. And the thing that when the name came out, I didn't coin it, and I was actually like most people, like why do we have to keep trying to change the DevOps name? Can't we just accept it's a metaphor for two groups working together collaboratively? And so, I was in that camp of like every time either Gartner or somebody tried to come up with a better – what they thought was a better name for DevOps, I would totally get annoyed and then try to kind of push it down, because the one single name has created an incredible movement of commerce and transformation.

And then two years ago we called it DevSecOps, and all of a sudden, 70, 80 percent – this year probably 80-plus percent of the room were security professionals. And so early on, I realized this name had gravity.

And even as I went to some of my clients early on when this DevSecOps name started to catch gravity. But the thing was, early on when I started testing the waters of this name, I'd go to some of my largish clients, or peers, and I'd say have you heard this thing called DevSecOps. And they would literally leap forward and say, "No, tell me more." And I realized this term had gravity. It caused – just that simple S-E-C in the middle, people understood exactly what it meant.

And so, the term has done us well in the last couple years because we literally – the term has really sent out an invitation, and for the most part, even though it's still kind of early, we're seeing tremendous opportunity of collaboration between security and kind of people who normally were thought as kind of pure DevOps supply chain. We're finding a lot – like fantastic stories of – and companies I'm working with where the – how they deliver the software in the pipeline is not – is now inclusive of security control points. So, things like – and by the way, not just like that they always did vulnerability scanning, but now it's like every point in the kind of pipeline of software delivery, not only do you have something DevOps-ish, but source control, test-driven development, you have a build, you have a (Inaudible), you have all those things. At each one of those junctures, you also now see people taking a systemic approach, a systems approach of security. So, it isn't just like, "Wow, we've got security covered cause we do artifact vulnerability scanning." People are having discussions about full lifecycle security completely intertwined and embedded and collaboratively working with everything else that's in the software delivery pipeline.

Another great example is we're seeing a lot of organizations move away from having a vulnerability issue ticketing system and then a bug-ticketing system. James Wickett, one of my co-authors DevSecOps handbook, says a bug is a bug is a bug is a bug. So, if (Inaudible) is your kind of feature and issue system that you're using, then you find this kind of maturity level where people just put the vulnerabilities, the defects, all those things are really treated the same way. And so I attribute that to the term that a lot of people still kind of hold their nose on, and I've just said, you know, hey, I just want to move forward, and again,

Google DevSecOps and you'll find a plethora of great emerging opportunities of people who are doing the right things, finally getting security to break out of the kind of silo last mile, hey, let us look at it, no good. It's this integrated collaborative discussions.

One of the things that I'm spending a lot of time talking about, and this hurts people's heads, but moving people away from sort of what I call subjective attestation of your security. Attestation is like I'm proving – like the CAB, you know, Change Advisory Board, is a subject kind of attestation system. A human says I'm going to do these things and I would like to put into production and here's what I think's going to happen, here's my back-out plan, hey, CAB authority board or people, this is kind of human-to-human, can you look at this.

And then, okay, somebody who's responsible for production environments will be like, yeah, I think this is okay, looks like they got all the – then you put it in production. And then when an order comes up, you're looking for that kind of change ticket, and what you have is a subjective attestation. Right? And what I'm saying now is what I'm really excited that people are kind of moving away from that to more of an objective attestation where you actually build the kind of – not only the security and DevOps – DevSecOps control points in the pipeline, but you actually create like kind of crypto events that say no human did this and here's the kind of – I don't want to scare everybody by saying blockchain, but it's a distributed ledger of a trust-based system that tells me that no human touched this code when it left the development environment, and it has done all these kind of attestations that are far more valuable from an audit and less theater-ish than a human-to-human.

Mike Kavis:

You know, DevOps to me has always been from idea to idea running and then how do we get rid of the bottlenecks in-between. And I think we started with CI, let's get the builds right. Then we went to CD, let's get the environments right. Then we started shifting testing, let's just automate testing, and I think the big bottleneck now, one of them is security. And we're trying to deploy ten times a week or some number like that. We can't have all these CAB meetings anymore, so we have to do it through automation, and I think the big shift in the mindset is once you have the automation, you have to trust the automation, right. How do you trust the automation? Continuous security monitoring. So, it's a big mind shift change. The companies that are moving to DevSecOps, are they embedding engineers in the development shops or are they still a silo out there kind of dictating?

John Willis:

You know, of course, like everything else, I think if I had to swag how many people would be really "DevOps," we would say just maybe 20 percent of most enterprises. And so, like DevSecOps would be in the 5 percent range. So, we're talking about a trajectory of a growing path. So, you asked me are they embedded. I would say that the people who are reasonably mature on a DevOps kind of roadmap or path would be building more of build-run type themes, like we used to call them two-pizza teams now. I prefer to use this kind of concept of a build-run team where each service has all the componentry of people that are required, we have a product owner, a software engineer, Ops, and now security. So that would be a model that people are adopting, but there's still – there's an SRE model as well for cyber. And then there are people that are just still trying to create cross-functional teams. I think that's – if that's what you have to do to get the ball rolling, God bless you, but ultimately, I don't like cross-functional teams, but sometimes that's the ignition to start a cross-functional – to get – to kind of – the symmetry between the different groups and certainly pull security out from kind of a castle or a silo.

But I do want to go back to one point. So, the embedded thing is interesting, but again, the more interesting part is it's been easier to do automation from what we call DevOps side, and we keep struggling with how do we get rid of the CAB. We can tell people you've got to get rid of the CAB. And the barrier to the CAB is trust, proof, attestation, right? And I think as even now in DevSecOps, I think this kind of notion of what I'm calling automated governance, or I like to call it objective attestation, is finally an answer or an alternative to the CAB. What we haven't been able to say is we can say get rid of the CAB. Why? Because it's – high-performing organizations don't have CABs and you can't move fast. And they're like yeah, but I still have an audit once a year, right? And I've got to be able to pick a change and show some provenance. And unfortunately, the legacy provenance model that we have is this subjective attestation. Hopefully I'm making sense here to people.

Mike Kavis:

You are, and kind of the beauty of automation is you could have real time auditability. And that's where all this is leading.

John Willis:

That's how we steal the CAB. We can't just tell people to get rid of the CAB. They say why. We say because it's not a healthy way to do your high-performance organization. Yeah, but I have this constraint that I'm going to get audited and they're going to ask me questions. And none of this is easy, but when you start putting the attestation into the supply chain and the automation, now you can say, hey, auditors, I want to kind of train you on a model that's a little different and, by the way, it's not subjective. It's actually objective. And then separation of duties and provenance and all those things become literally hashes. And if you're setting it up as a trusted model, again, what level of trust you want to use, I mean, something like a blockchain. And again, I have customers that are doing this with pure blockchain. It doesn't have to be that, the point is that's – it is the automation, to your point, but it is that we have to start thinking about automation driving objective attestation of the things that we're doing in the pipeline so that we can prove (A) no human touched it since it left the development desk and that this change or feature adheres to the – it has the evidence, a trusted evidence, a non-interruptible, a non-human interruptible evidence that it did all the things that had to be done. Had to have vulnerability scanning, had to have security (Inaudible), it had to go through artifact, all those things. And here is the hash that proves the trust that we've done this.

Mike Kavis:

Well, the good news is we haven't adopted DevSecOps engineers yet.

John Willis:
In a perfect world, a successful DevSecOps movement will eliminate the name and we'll just go back to calling it DevOps.

Mike Kavis:
Yep. So, the next question is on an area – we talked about this last time we met at the DevOps summit last year, and you've been speaking about it a lot. As you go into places and you start interviewing people and there's a lot of work that's not visible to people. You call it invisible work. So, tell us about that. I think that's in part of your seven deadly diseases of DevOps but tell us about the work you're doing there.

John Willis:
Writing the DevOps handbook was like intense study of lots of different companies. In fact, the book has 48 case studies. And what happened like two years ago, the first company I came into, what I kept finding is I kept peeling back the prescriptive nature of what I thought transformations had (Inaudible). You know, I came in first doing the kind of lean value stream mapping, like everybody would talk like that's tool number one. But what I found out is that when you're going into these companies that are institutionalized, they're siloed, they've got persistent institutional memory muscle. Like I said, I'll repeat they're siloed. You know, one of the things about value stream mapping is you say, okay, let's pull all the people that are associated with this value stream, and let's try to gather as much truth as we can about the value stream to understand what it is and how we can improve it.

And so, when you go into a company that's kind of still legacy, kind of hasn't broken out of the mold of silos, what you're really doing is creating an artificial environment. I'm a huge fan of lean value stream mapping, or particularly value stream mapping in the DevOps prescription, but it's – that's like – that's not the starting point, I found. Because what you're doing is you're pulling people out of their natural habitat, and you're asking them to give you truth. Now, you get tremendous truth, but you don't get the whole truth, Igo in. Usually when it's really successful at the CIO level, and I say give me a few weeks with your organization, and I want to take a sample set. Three development teams, an infrastructure team and a cyber-risk governance team. And I don't want the ones that are raising the hand that say, “Oh, me first, me first.” And I don't want the ones that are passive-aggressive. I want that middle Goldilocks zone. I think value stream mapping is something that's tremendously effective as a second phase, but like I said, I want to find the real truth. So, what I do is I have interviews. I just have discussions with the teams in their natural habitat. So, if development team – it's a marketing development team or another one is like a finance development team, I want to meet that team without their managers and without any other teams in that meeting (Inaudible). So, do Development Team 1 on Monday, Development Team on Tuesday, Development Team 3 on Wednesday. And I'm learning from each team about what more to gather the next team. And then Thursday I bring in the infrastructure team, again, independent of the other teams. And then Friday I like to end up with some combined risk/cyber, depending on how you classify the structure.

I use just flip charts. Low tech, no computers, it's just me and usually an assistant, and I'm just gathering as much as I can on these kind of flip charts. I mean, literally by the end of the week, I have 100, 200 flip charts. But this tells you why I want to start; it's like at the primordial level, if you will. First thing I do at a team is I draw a box and I say where does work start. And on average, teams try to argue with me, for about an hour on average, that that's not a good question. And I play Colombo, like well, yeah, it might be a good question, but do you mind answering it. And they're like, well, John, you don't understand, it's a complicated system here, we've got this. I'm like, yeah-yeah, but where does work start? John, you just – there's lots of pieces.

And then finally at some point I'm like, okay, Monday morning, there's like this breakthrough, well, if you must know – and now you start opening up the question of like, why do you take certain work, how come certain work you actually create tickets for, certain work you don't. Oh, we never make tickets for that team. That's Bob's team and we always know that their work is going to be pretty seamless – well, you don't really know, right. And here's the thing. So, what I call the first deadly disease of the seven deadly diseases, they're really just patterns – I call them diseases to make it sound cool when I'm presenting, but the first one is that, on average, most companies capture less than 50 percent of all their work. You've got a $1 billion budget IT department. So, this is the classic scenario. I go back to the CIO. You know, good CIOs are like, okay, they're going to crush platitudes.

In an industry where we're always screaming about data this, data that, I mean, every organization I go to in the last two years is somewhere in the range of 70 to 30 percent capture rate. And like I said, I can average and say it's around 50. We've got to fix that. You know what the second deadly disease is? There’s no continuity on ticketing systems. You've got SharePoint, you've got e-mail, you've got Remedy change tickets, you've got Remedy requests for services. Some people have written their own issue tickets, and they all have different contexts of what they're supposed to be there for. And so right off the bat, you've created a framework for people to not even to enable to capture the work.

I won't go through all the sins, but the third sin is you don't have proper inventory of what you've got. You know, when you ask people questions like when you get a critical or high vulnerability, what do you do with it? Their answer might be if it's infrastructure, we know where to go, but if it's applications, yeah. And why? Because we don't have an accurate service catalog of ownership of services, whether you call it CMDB or service catalog. Well, I already know part of the answer, which is half the work you're doing, half the installs that people are putting in the system, there's no documentation for it. And there's things that are going to be quasi-development that steep their way into quasi-production, which then becomes shadowy. Like I know it's hard in a $100 billion, $40 billion, $200 billion market cap company to change things, but my God, man. Capture 95 percent of your work. Do that first.

And one last thing that I think this all blends down to is the seventh deadly disease is security compliance theater. One other thing is, in the second week, I put teams together. So now I start kind of smashing truth out of people. You know, Joe, you said that if you – anytime you make a ticket as a major status, it's going to take at least a day. And Bill here who runs the CAB says that's baloney. That happens one out of every 50 times. And then this engineer guy pops up and says yeah, yeah, it's probably my fault. There's certain things that we have to do manual steps because things like Tibco and Charles River don't have source control, so we have to do automated – and then you start unraveling these kind of folklore-ish truths.

Anyway, it sounds kind of hand-wavy, but again, ultimately what you kind of magically end up with is sort of a blueprint of the real truth of your organization, and then that real truth is sort of the answer like you got through five value streams and everybody's patting themselves on the back because we DevOps-ed five teams. And then you go to the sixth team and for some reason now it seems like a square peg in a round hole and you're yelling at the team like get onboard and you don't really know. And the truth is you really haven't uncovered this whole truth, and that is the answer of why most companies get to somewhere about 8 to 15 percent success with their DevOps transformation and then hit like this horrific brick wall. So, what I try to do nowadays is try to uncover the whole truth, or an underlay of truth, before you look at value stream.

Mike Kavis:

And to your last point, somebody with a tool – you know, the tools aren't a means to the end, right. You should probably know what your issues are, so you know what to use the tool for, right.

John Willis:

Yeah, if you don't capture half the work, like, my God, what are we doing here? We're on the dark side of the moon.

Mike Kavis:

The last question, we'll try to keep it to five minutes, this is where I'm spending a lot of time and I talked early on about the bottlenecks, and I think a big bottleneck now for companies that are three to four to five years in is figuring out the run part of it. Because traditionally we ran stuff in the data center with a separate group, CABs, all this stuff you were talking about. And then we start building all this automation, this code, and some of the groups want to maintain their empires, and we're trying to shift all this Ops to left. We call a lot of their stuff the new operating model, cloud-operating model. So, we're spending a lot of time there, but at the same time that we're trying to figure out how do we reorg and change processes to run this stuff in the cloud, people are moving up the stack in the cloud, right. The higher and higher levels of abstraction where there is no infrastructure to manage anymore, we look at serverless, you do on any of the big cloud vendors now and you get streaming functionality.

You mentioned blockchain. Amazon just released blockchain ledger database as a service. You used to have to go build that for six to eight to twelve months. So, my question is, as we're becoming more and more abstract from the plumbing, are these companies aware of the big shift in responsibility of operations to the devs, and how are they handling that? To me, that's a big bottleneck right now is you get companies out – no matter how good or bad their data center and infrastructure is, it's known. They know how to deal with a crisis. Then you move to this greenfield cloud thing, you build all this stuff and it's great, then you try to scale and all of a sudden something breaks and it's like we don't know what to do. And then the cloud gets poo-pooed, oh, the cloud. But it's not the cloud; it's the engineering, the Ops side of it.

John Willis:

No, no. And I think there's three principles that people have to get on right now. So first off, stop calling it a cloud. Basically, the world – I'm going to say Kubernetes and containers. I don't care if it's – it's going to be clusters that are scalably-managed and a new form of compute that is going to be completely different than the classic virtual or operating system model. Again, we don't need to go into a whole tutorial on the difference between a VM and a container or how some people might run functions under containers, or completely divorcing containers, but at the end of the day, I would say that organizations need to put first front of mind this idea that the world is moving to massive, or clusters of workloads that will be managed through some orchestration. Today it is clearly Kubernetes, and for the most part, the compute model is containers, with a sideshow of serverless.

Mike Kavis:

Let me jump in here re al quick. I would raise it up higher, and I would say we're entering the era of platforms. Whether it's a Kubernetes platform or platform, or even a highly extracted service.

John Willis:

But I think platforms might narrow our view. That's my worry there too. It's like cloud. Like because then we get into is it IaaS or PaaS. Honestly, for somebody who's gone through the cloud wars like me and you, it's nonsense. Start thinking about clustered work. R ight now, if you had to ask me, I'd say start thinking about workloads that run on the orchestrated models, mainly Kubernetes. There are some alternatives, and they are this container or – I use the word serverless differently than most people – a serverless instantiation of compute, which I would say arguably is compute containers. I think when you start using platform, you build in these frameworks. I think you have to get the gravity of your organization and start thinking about the world, if it goes on this progression that it's on, will be mostly clusters of these very kind of atomic compute instances, whether containers or – and again, I'm not saying the mainframe still won't exist and people will have major workloads in VMware, but most of the gravity and direction (Inaudible) of where you're going to go is this new thinking.

And so, the reason I point that out is then you don't have to talk about cloud. Because once you start talking about clusters, Kubernetes and containers, again, to use simple two words, then your focus is better suited and then what type of framework it lands on doesn't matter. Is it Amazon? Yeah. Is it GCP, you know is it bare metal, is it Packet Cloud? You know, the more people start to silo, I'm going to build a cloud operation and cloud management team and then, well actually I've got a problem, because I've got my Google cloud management team and I've got my Amazon. It's just a framework, and again, the abstraction is going to be clusters, again, Kubernetes and containers. That's principle number one.

Principle number two, to kind of the shifting workload of how people have to deal with this. I think that the world will kind of gravitate to build-run teams, so

you'll have basically most services will be these circles of multiple individuals, you know, your product owner, your network, your development, your security, your operations, all in that team, call them two-pizza teams.

And then kind of third principle is the run side of it. So, they're build-run, but they're really build-run in the sense that they do everything they can to create the service, make sure of the service's reliability, create all the operational control points when it runs, but then you build in SRE teams and then you have kind of SRE for network, SRE for this. And then I like what Damon Edwards says is that becomes this self-regulating system of balance.

So, if you don't have a lot of time to go into error budgets, SRE error budgets and SLOs. But, I think the predictive nature of where I think most people need to be, at least as we sit today, is to really understand that directionally or kind of the gravity of where I think the industry is telling us to go is to really understand that stop worrying about cloud, worry about managing clusters of compute nodes, and then start really getting a developer mindset to move to more build-run team, and then how do you arbitrate the operationalization of those as the build-run teams actually set up the ability for those services to evidence their controllability, manageability, serviceability, basically control points. And the SRE team creates the structure and is enabled to deal in anomalies that break out of those things, and then the SRE build-run team becomes this self-regulating system. Again, that's probably the quickest explanation of kind of SRE-balanced build-run team structure. But I think it really is important for people in leadership roles to start – the evidence is that some variation of those three ideas is where all the kind of momentum and where I think most of the views of the leadership, the right leadership in our community are saying people need to point towards.

Mike Kavis:
Great stuff. Appreciate your time today, John.

John Willis:
Yeah, it's fun.

Mike Kavis:
Yeah. Where can we find you on Twitter and where specifically can we find your Seven Deadly Diseases of DevOps presentation? Or any podcast you may have.

John Willis:
I did a webinar with Electric Cloud. I'm actually giving my first full presentation next week in Australia, so that will be up under Dev Ops Talks Sydney. And then the best way to catch me is this horrible word, which I love nearly and dearly, called Botchagaloupe, B-O-T-C-H-A-G-A-L-O-U-P-E, and that's Twitter. That's my GitHub. Most of my presentations are on GitHub, so I'll have – as my GitHub project called My Presentations under the Botchagaloupe ID for next week will be up there with the video and all that.

Mike Kavis:
All right, man, thanks again. That's our show for today. You can find this podcast and more from myself and my colleague, Dave Linthicum, just by searching Deloitte On Cloud Podcast. We'll see you next time on Architecting the Cloud.

Operator:
Thank you for listening to Architecting the Cloud, part of the On Cloud Podcast with Mike Kavis. Connect with Mike on Twitter, LinkedIn and visit the Deloitte On Cloud blog at www.deloitte.com/us/deloitte-on-cloud-blog. Be sure to rate and review the show on your favorite podcast app.