On Cloud Podcast
July 2024

The Deloitte Architecting the Cloud Podcast

Mike Kavis, Chief Cloud Architect, Deloitte Consulting LLP

Title: **Improving software engineering: John Willis on how Deming's theories can enhance the development process**

Description: In this first episode of a three-part series, Mike Kavis sits down with John Willis, DevOps expert and author of the book, "*Deming's Journey to Profound Knowledge*." John recounts his DevOps journey and why he wrote about Dr. W. Edwards Deming's management principles. He argues that Deming's theories can be directly applied to software engineering to improve the development process and that two of Deming's concepts—psychological safety and systems thinking—are absolutely critical.

Duration: **00:27:04**

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 and AI, but most importantly, with the people in the field who do the work every day. I'm Mike Kavis, your host and chief cloud architect over at Deloitte. Today, I'm joined by a good friend, John Willis, aka Botchagalupe, and John is the author of 12 books and is a legend in the DevOps and cloud space. And if you don't know John, at the end of this show, we will share his information and you need to read all this stuff and watch his videos. It's really great.

So, John, before we get going, it has been almost five years since we last did a podcast, and I was trying to go back in time to see when we did our first one and I think it was around 2008 or '09. So, we've been we've been chatting for a long time. It's good to see you and hear from you again. And since I've got you, today we're going to focus on your new book, *Deming's Journey to Profound Knowledge*. And then we're going to record on the intersection of DevOps and AI. So, for this one let's dig right in. And tell us what inspired you – well, first of all, what's this book about and what inspired you to write this book?

John Willis:
Yeah, I think, Mike, it's been a great – I don't know if you remember when we first met, you were one of the first successful commercial implementations of a product built on Amazon, as far as I know. And I had a pretty good universe. And you reached out to me as a consultant. I'm like, "Man, I'm just learning this stuff" and you're like, "So am I." And I helped you a little bit on your design and architecture. But yeah, and that was crazy years ago.

Mike Kavis:
2008, man. 2008.

John Willis:
Yeah, there you go, man. That's pretty crazy. As you said, and we've known each other for a while, I've gotten into a lot of things, early cloud, that nonsense where some people called some of us "clouderati." But pretty soon into that, I got back to my roots. Cloud was interesting but I've always been an Ops person, systems administration, and DevOps exploded. DevOps has been very, very good to me. And through going to the first DevOpsDays and then Damon Edwards, who's one of my primary partners in crime when it comes to DevOps, almost all things – we ran the first DevOpsDays in the US. And what happened was Damon invited Gene Kim to come to that event. That was in 2010. And I was on a panel, and I knew Gene. But I guess I just didn't know what he looked like. And sometimes I'm just like, "Go here, be on this panel, do this."

And I'm on the panel and I'm having a conversation with Gene and I get off the panel and Damon says, "You know who that was?" Because Damon knew I was a big fan of Gene's *Visible Ops.* And Gene in operations was a god. And he goes, "Do you know who that was that was sitting next to you?" And I'm like, "I don't know, panel guy number four?" And he's, "No, that was Gene Kim." I'm like "Oh, my goodness." And I literally race around the conference to find him. And he's already gone.

And so we agree to meet at South by Southwest only a month or so later, I think it was. And we met; we had coffee. And anybody who meets Gene, they realize what a beautiful human. He just genuinely wants to help people. And we're going through all that and then he tells me about his book that he's writing. The book is *The Phoenix Project*. And I asked for an early copy. "I'd love to get involved." We even started talking about writing a companion book, which evolved into *The DevOps Handbook*, which I'm co-author with him. And he said, "Well, you know what? I think you really should read this book first." And it was a book by Eliyahu Goldratt called *The Goal*, which is a classic industrial engineering book. I'm like, "All right, yeah, sure."

And I read *The Goal,* and I was like, "Oh, my goodness, this guy is in my soul." And so, I read *The Goal*, I read *It's Not Just Luck, Critical Chain.* I just consumed his work and I became a big fan. And then, eventually Gene gave me one of the early versions of *The Phoenix Project* and I read it. And for those who don't know, Gene purposely wrote a modern-day rewrite of *The Goal, The Phoenix Project*, which is a modern implementation. But the story was methodically designed to be the same story, a Goldrattian model of storytelling.

And long story short, we ran the second DevOpsDays. And now by this time I'm considered a Goldratt expert, and I'm one of the few people on the planet that's actually read the early version of *The Phoenix Project*. So, we've got this down. And we're running an open spaces on theory of constraints, which is one of Goldratt's primary ideas or methodologies or philosophies, I guess. I don't know. Pick one. And a guy named Ben Rockwood, who I'm a huge, huge fan of. He's incredible, well-read on operations, operations research. He runs this thread on theory of constraints. And Deming comes up and Ben says, "Oh, John, by the way, this all goes back to Deming." And I'm like, "No, man, Goldratt. Don't confuse me. I'm a Goldratt fan. I don't want to hear about this guy." And he challenges me, just like Gene challenged me. He said, "You really should go look at Deming's 14 points.

And I Googled it and I went and looked and I was like, "Oh, my God, this guy like is spouting the essence of DevOps." And what's interesting is he was doing it 50 years before DevOps. If you think about how revolutionary DevOps was changing the way work habits went for – in IT between developers and operations and silos and this this guy was talking about this kind of stuff at least 50 years ago before DevOps was coined. And I was like, "all right, well, now I'm going to really dive into this guy because this is fascinating." And then – no disrespect to Goldratt and I'm still a huge fan of his, but Deming was just – it's like opening up Pandora's box.

I decided to create a presentation, another recommendation by Ben Rockwood. And I called it, "Deming to DevOps," which I wanted to go through the history of how did this guy learn and think about this stuff that ultimately was lean, agile DevOps? And that was an incredible journey. I did some early presentations and then I had this idea with Gene because Elliot Goldratt did an audio-only book 20 years after he wrote *The Goal*, and he called it *Beyond the Goal*. And I pinged Gene and I said, it was about five years after *The Phoenix Project* went out." I said, "We should do a *Beyond the Phoenix* project." And Gene loved the idea. And it's a book – it's one of my 12 books, if you will. And I covered in depth Deming. He covered Goldratt. He covered the why, if you haven't listened to the book, it's a fabulous book, that Gene gets into the how Goldratt came up with the book, how they architected *The Phoenix Project*.

Anyway, another long story short, Gene told me at that point, "Hey, you should write a book about Dr. Deming." And I put it in my back pocket. I started accumulating all these great stories over the years, get a little serious, do some research, create some directories, some files. And then along comes the pandemic. And I realized, "You've been working on this thing for 10 years and are you ever going to write it?" And then I thought, "The pandemic – if I'm ever going to write it – If I can't do it now during the pandemic –" because I was getting back like 100, 150 hours a month from travel that I wasn't traveling, because I travel quite a bit. And so I said, "Yeah…"

I did some research. I found some really good editors who helped me find some people, some great consultants, primarily a great consultant who's attributed for helping write the book, Derek Lewis. And during the pandemic I wrote the book. All during my research, I found these great stories. And every book written about Deming was so formulaic. It was basically a little bit about his biography and then all about that author's ideas about Deming's theories. I mean, good books for Deming's theories. But I thought there were such great stories. I was finding the people he worked with, people he met, people he influenced, people who influenced him. And I'm a big fan of Michael Lewis: *Moneyball, The Big Short, Flash Boys*. They're biographies but you don't know that they're biographies. And they can teach my mother-in-law or an operations or a day trader. The same narrative; they just learn it differently.

I thought, "Man, if I could write a book in a Michael Lewis style about this guy, because he's got great stories, and there's some complexity in his ideas –" or simplistic complexity, which is a Goldratt theme, that some of the most complex things are actually really simple. And I was like, "I'm going to do this." And I spent a solid year writing the book and then took probably another 18 months to get it…When you think you're done with a book, there's a whole other world of revisions. I had eight, nine revisions and then there's going through the edit process. But ultimately, it came out last August, the audio, and eBook. And then in January, the physical – the paperback came out. So, it's been a good ride. It's one of the more, other than my kids and my wife, I mean, probably one of the most fascinating things I've done in my life.

Mike Kavis:
Man, it's coming up on a year already.

John Willis:
Yeah.

Mike Kavis:
That's exciting. I've written two and I don't know how you write 12. And like you said, you're almost never done. It's just so much work. So, hats off on that. So, now that we've got the background on that, can you – one of the topics in there is the four components of Deming's system of profound knowledge and their relevance to today's management practices. So, keep in mind that a lot of the audience may not know anything about Deming, so make that simple and explain that to us.

John Willis:

That's a good point. So, just to give you the heads up on Deming: It's called system of profound knowledge, which at this point, if you've never heard of Deming, you're like, "Who is this guy calling his idea profound knowledge?" But he's very particular about his words and he didn't mean it in an egotistical way. He meant it in that there's this body of knowledge that if you can harness it, and the interesting thing about who he is and what he did is it’s a really cool story. One is you might know him through quotes. I mean, the data quotes, "In God we trust; all others bring data."

You've heard his quotes. You may not remember that it's him. He's usually attributed as the person who was sent to Japan after World War II by MacArthur. It's a much more expansive story; it's not that simple.

NBC does this documentary. And this is how I start my book on this story because it's fascinating. It's called, "If Japan Can, Why Can't We?" And it's this whole thing about Japan, what they did after World War II, how they revolutionized their ideas and concept for manufacturing. It ultimately gets coined as lean at some point a little later, after this documentary.

And then, all of a sudden, the whole TQM, Total Quality Movement gets born based on this documentary. Again, I'm putting it in very simple terms. The book goes into a lot more complexities of this narrative.

Anyway, so the principles behind these ideas that he used during World War II to help us win the war, the quality movement during the war, which is, actually, most of it's classified. And then, what is the revolution that put America back competitive, is basically based on this idea that he wrote about in his last book called *System of Profound Knowledge*. The other thing that was fascinating about Deming is he was writing about complex systems and complexity and the truth of the matter, in the '50s, when he goes to Japan and helps Japan, he says the same thing to them, "If you follow my ideas, you will be a world economic power in five years." And they do.

And the principles behind this is the idea – again, he understood complexities in systems thinking in the '40s and '50s. If you think about the science behind complexity science, I mean, most of the real research starts happening in the 1950s, '60s and '70s with Lorenz, the butterfly effect, all that stuff. Deming's talking about this stuff in the '40s and '50s. And what he said is in order to understand complexity, you think of it like a camera. To take a perfect picture, you have multiple elements of a lens. One is aperture, the light exposure. One is depth. One is the focus. And he said, "So, how do you take a perfect picture?" It's a complexity. In my book I use the kiss, the famous kiss after World War II where the sailor grabs the nurse, and he kisses her. That's a perfect picture. That isn't just happenstance. It doesn't just click. A great photographer tries to get the right exposure, get the right focus, the right depth. How do you get the faces?

And so, Deming said that when you're dealing with complexity in everything, you need to think of complexity of using a lens. And he called this thing system of profound knowledge, which is made up of four elements. And I'll run through them really quickly. In the book, I spend a lot of time trying to figure out where he got all this from, because he didn't invent any of this stuff.

So, the first element is what they call theory of knowledge. We know this as scientific thinking, scientific method. It's how do we know what we think we know? If we know anything about DevOps, lean, agile, we know the biggest mistakes we make are biases, where we believe something works a certain way. We don't test it. And we're like "I've seen that before. Let me do this." There's been numerous accidents and postmortems on plane crashes and stuff that talk all about this stuff. So, the idea that you really try to understand that you don't know, you have a hypothesis, you have an idea, but just eradicate that idea that you think that you think not you think, that you know what it is.

And so, again, the scientific method, back to epistemology, you can think of it as correlation versus causation. So, when we have these biases or correlation, we look for patterns to say, "This causes this. Okay, I know what it is," as opposed to causation or more, maybe specifically the difference between inductive or deductive reasoning. Anyway, that's knowledge. There's a lot there.

The second element of this complexity lens, system of profound knowledge, is understanding variation. And at the core, the way we build toasters and nuclear power plants uses these ideas based on statistical process control and control charts. Unfortunately, we don't do any of it in IT, which is disheartening. And the idea then is how do we understand what we think we found out? And it's all through statistical models and something called analytical statistics. And at its core it's the ability to separate the data from the process. So, now you can decouple the bias. I'm just going to show you data, statistical patterns of data.

And again, this stuff, if it sounds like, "Well, this guy, what is he talking about?" I mean, the way your nuclear power plants, the way your toaster, the way your car was created all use this technology. And so, there are patterns that have been developed over 100 years that basically say, "If the data looks like this, and it's between an upper and lower control limit, and it has these patterns, there is something not right." And it's mind-blowingly simple, but it uncouples unbelievable complexity, because now you're able to have statisticians or analytical statisticians look at data, turn it over to subject matter experts and say, "The data is telling us something here. What does it mean to you?" "Oh, my goodness, that's when we changed the such and such; we updated version three to four."

All right, third piece, psychology. And this is where it gets really brilliant because the first two are more technology, but the fact that he added in this idea of psychology, and again, this is the third of the four lenses – is understanding motivation, understanding decoupling biases. And then you start seeing them all merge together. Epistemology or a theory of knowledge, looking at analytical statistics to uncouple what the data and what – the things that might be happening in the real world. The real world might be software. But now we're adding in like, “OK, the human is involved in this complex system.” And humans have, we'll call them frailties or biases. We’re weird animals. We decouple us sometimes from an organizational behavior structure as intrinsic versus extrinsic. it's not binary. Intrinsic motivation actually has eight or nine psychological classifications. So, sometimes we say, "Oh, this is simple. I just look for my intrinsically motivated employees and I treat them that way. And I looked for my extrinsically motivated employee and…" Easy as pie. And I try to go into that a little bit. But the point is you're applying that additional element into the lens, like the aperture or the depth.

And then last, but certainly not least, is appreciation of system, which is systems thinking. So, you put all this together. If you know anything about system thinking, it's working inside out. There's always a bigger piece to look at. How does what I do, or if I make a local optimization here, am I looking at the global effect or global optimization? So, that is as quick as I can do it. But again, if you're really interested in it, I would definitely read the book. But that is the system of profound knowledge.

Mike Kavis:
So the question with all that is yourself and Gene and Damon and all these guys who started championing this DevOps, how do you take all those things that you said, which are making successful manufacturing companies, and apply it to successful software building?

John Willis:
Yeah. What's interesting is you see all these elements all over all these successful implementations and success stories around DevOps because there's two things that are consistent to every successful, and all four have to be knitted together to work at optimum. But two that are clear is basically systems thinking. If we've learned anything about how we optimize things, you think of a software supply chain. If we just go in and optimize one piece of it, and this is what *The Phoenix Project* was written about, the theory of constraints. You can't just change one thing and not realize other things might be affected. Or the idea that sometimes in a global organization, some parts of the organization have to be suboptimal to make the aggregate optimal. These are hard concepts. So that's a very simplistic way of describing some of the benefits of systems thinking.

And then on the book end, front end, which is how do we think about what we do? Do we just make rash decisions or knee-jerk decisions about we get a breach… There was a bank one time that I worked with, they had a breach, and the CEO ordered them to put a req out for 150 new IT security people. But the point is you don't even know why the breach happened, where the breach happened, what all that stuff is. And the knee-jerk was, "I want a mandate. We need to hire 150 new –" and this is a true story. And I mean, anybody who's listened and been in IT or doing this stuff, you could have hundreds of these stories.

So, the idea that theory of knowledge, it at least gives you a guardrail, because if you start off with the idea that you don't really know what you think you know, all you have is a hypothesis, then you can follow what's been successful since the scientific revolution, theory going back to like Francis Bacon and those guys. It's how we solve almost every scientific problem including the vaccines for COVID. It's scientific method.

So, yes, scientific method, systems thinking. And then, again, using a little bit of analytical statistics and like understanding the human element of how we deal with change. We can't just tell people to change; we've got to understand what motivates them to change. Yeah, I think it's all part of almost every success post-DevOps success story that you've read, seen, or witnessed, or been involved in. Most of those, if not all those elements are present in that success story.

Mike Kavis:
Yeah, I'll give you an example. I read *The Goal* back when Solo was the big thing, because what we were trying to do is tie all these stovepipe systems together, because there was redundant data in each, because they were all built in a silo. And we had this one SaaS user who used to request these mammoth data sets and they would consume so much resource. One day, I sat down with the gentleman and said, "Well, what are you doing?" "Well, I pull this out, and then I run this data set, and then I run that data set, and I run…" and it was this process. And I said, "Well, what are you getting at the end?" "Well, I'm getting this." "Well, why don't I just give you that?" And that eliminated weeks of compute power.

I mean, it's kind of that system thinking, is take a step back from the micro world and say, "What problem are you trying to solve?" And then solve it more holistically. And it saved so much compute and storage and time. And then the guy freed up all his time. He was an analytics guy. He could actually do analytics now instead of running SaaS jobs his whole life. So, those types of things, and we see that in the DevOps pipelines to streamline things.

John Willis:
Absolutely. Absolutely. And you know what else is really interesting, Mike, is the whole idea of psychological safety. You can tie all that back in too, because think about two of the four elements, probably all four of the elements, again, it definitely applies to psychological safety. But the idea that if you're using and living by this concept of theory of knowledge, is that you don't really know what you know and everything is a hypothesis and a an experiment, then that allows the person who's been on the job for three days of any diverse background to be able to raise their hand and say, "I don't think that. I think that's going to work differently." Because if you're in a scientific mindset, some people call it an engineering mindset, but you've already anti-brainwashed yourself to those biases. The example without theory of knowledge and theory of psychology is the young person at two days says, "My last job, we had, that didn't really always work that way." And the senior person is saying, "What do you know? You've only been here three days. Come back and talk to us in a year from now."

The psychologically safe environment is "Wow, why don't you go experiment that? See if you're right." And again, we've learned a lot of that from really a lot of the cloud titans and these young kids who are engineers. In fact, a lot of them were physics people who live and die by scientific thinking. So, yeah, so there's that other benefit of it creates the – the diverse psychologically safe environments that we – look, hopefully everybody agrees we need to have that success.

Mike Kavis:
Fascinating stuff. I'm going to have to wrap this here and we're going to do another one. We're going to focus on Gen AI, but for today, fascinating stuff. Tell us where we can find the book and where we can find – I mean, you've got so much content. You have a podcast. Where can we find all this great stuff?

John Willis:
Yeah, so John Willis, if you LinkedIn "John Willis Atlanta or John Willis DevOps on LinkedIn, I've been running two newsletters these days. So, almost everything I'm doing is LinkedIn newsletters. Sometimes I periodically point back to a blog, and I do have a podcast, but you can find everything from my LinkedIn profile, again, John Willis Atlanta, whatever. And the one is all about Deming and profound knowledge. I try to cover anything that is interesting in

the ideas of system thinking and psychological safety and all those things. And then I have a new one. So, I'm writing a new book. We'll talk about this probably end of this year, but it's *The History of AI*. And so, I've created another newsletter where I'm kind of thinking about all these ideas called "Attention Is All You Need." And so, those are the two places that I'm finding most people are responding to. That's where I'm putting most of my work.

Mike Kavis:
Cool. Well, that's our show for today. I hope you enjoyed this podcast. Make sure to like us, leave a review, and subscribe. You can also check out our previous episodes wherever you listen to your favorite podcasts. And you can always find me on Twitter, madgreek65, or reach out to me directly via e-mail at mkavis@Deloitte.com. Feel free to give us suggestions on topics you want to hear. Thanks for listening.

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.

Visit the On Cloud library
www.deloitte.com/us/cloud-podcast

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.
Please see www.deloitte.com/about to learn more about our global network of member firms.