Deloitte.

Architecting the Cloud
August 2020

Architecting the Cloud, part of the On Cloud Podcast

Mike Kavis, Managing Director, Deloitte Consulting LLP

Title: **Amplify software resilience and learn from failures by asking how, not why**

Description: Critical failures happen to software systems; it's just a fact. Learning from those failures and focusing on building resilience is more important than ferreting out who's to blame. In this episode of the podcast, Mike Kavis and guest, Netflix's Jessica DeVita, talk about resilience engineering and why asking "why" something happened is not the best way to deal with system failures. Instead, Jessica, through introducing the concepts of human factors thinking and local rationality, recommends that software teams ask what happened and how, so that blame is not a part of the equation and teams can build a culture of trust that can help build more resilient systems. She also gives salient advice on how companies can start their own resilience engineering journey starting with unlearning that human error is a cause.

Duration: **00:27:51**

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. 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:
Hey, everyone. Welcome to the Architecting the Cloud Podcast, where we get real about cloud technology. We discuss all topics around cloud computing, but most importantly we do it with the people in the field doing the work every day. I'm your host, Mike Kavis, Chief Cloud Architect over at Deloitte, and today I'm joined by Jessica DeVita, Senior Applied Resilience Engineer at Netflix. Jessica, welcome to the show. It's great for you to be here and tell us a little bit about your background and some of the things you're working on.

Jessica DeVita:

Sure. It's a pleasure to be here, Mike. Thanks so much for inviting me. Yeah, so I am at Netflix, and I work with teams after they've had an incident or an outage with their software, and we really try to help these teams, and all of Netflix, learn as much as they can from these incidents.

Mike Kavis:
Sounds pretty cool. And you're taking some kind of master's program in this area?

Jessica DeVita:
Yes, that's right. I am a master's student at Lund University. Lund is in Sweden, and they have a program in human factors and systems safety, and so I'm just in my thesis year now, working on preparing that thesis, and it's very exciting. It's been a very fulfilling and rewarding couple of years for sure.

Mike Kavis:
Pretty cool. And what's interesting is that area of your focus, and I'll let you kind of define it for people who aren't familiar with it, is our systems have gotten so complex over time that we're looking at these incidents just as if we were an airline industry trying to make sure that airplanes don't fall out of the sky. It's getting that complex, and the systems are getting that mission critical that we're having to apply some of that same thinking. So, talk about the space that you're learning in and writing your thesis on and why it's important to software.

Jessica DeVita:
Sure. Well, as you mentioned, the complexity is what's going on. Our systems are complex, and they're so complex that they can't be held a mental model by one person. No one person has some complete understanding of these things, and it's all of the people who touch that system together who keep that system up and running. And basically, because of that, it is really important that we call in human factors thinking and approaches to understanding these systems because they fail in spectacular ways. And it's not a simple outage. You have an outage and it takes down other services. When we look at some recent incidents with Cloudflare, for example, is one that comes to mind. AWS has an outage and it takes down half of the internet.

But I think the real fear here that we have to work with is that software, it's not just about websites anymore. These are supporting hospitals, these are supporting human life essentially, so we have to treat that with the same care and attention that other industries have done for years, and that's why the aviation industry is not new to human factors. They've invested a lot. There's at least ten pilots in my human factors program at Lund. And all of us, though, are on this same journey where we're trying to change hearts and minds around some key fundamental traps, like believing that human error is the cause of an incident.

And there's other traps like mechanistic reasoning where we think, oh, if I just find this one eureka part and I find the problem with that fine, problem solved, we can move on. And things like counter-factual reasoning. And that's where you say, “Well, they should have done this. Well, if they had only done this, then the incident wouldn't have happened.” The problem with that kind of thinking, like all of these traps that I've spoken about before, are they fundamentally are not about the incident that happened. They're describing an alternate universe that didn't occur. We have to help people stay with what did happen.

Mike Kavis:
So, we're going to dive into that, because one of the reasons why you're here is I saw your presentation on AlltheTalks. You were talking about “The Lies About the Five Whys,” but before I wanted to get there, I've been dying for this question. So, I work with a lot of people, and we start talking about concepts like resiliency engineering and all these things, it always comes back, well, you're not Netflix. Well, you are Netflix. So, the one thing I wanted to ask, because I see so many weird definitions and implementations and expectations of what resilience engineering is. So, what is resilience engineering at Netflix, and what is the role and responsibility of someone in that role? And that doesn't mean that's the way it should be for everyone. That's what it is at Netflix, and I'd like to get that out there.

Jessica DeVita:
Sure. And I'll send you a link to a blog post we recently wrote that describes how we think about SRE at Netflix and our central reliability practice. Ultimately though, our teams in core – I'm on the core team – we do not build services, we do not fix your service for you. We specialize in incident response and paging in the right teams. And we're ultimately when you press "Play" on Netflix, and if it doesn't work, that's my team responding. Now, after the incident has stabilized, that's where folks like myself, J. Paul Reed, Ryan Kitchens come into play, where we help with the post-incident analysis. And so that's this whole learning from incidents that we talk about is how can we take what happened and get more than just a Jira ticket out of it? Can we learn things that will help influence how people work in the future?

Can we learn things that drive a deeper understanding of our systems and the complexity of them? So, you're right, we're a little different than other teams, but that just goes to the point that context matters. What works for us won't work for you. You can't just lift and shift, you can't cargo cult anything, but certainly you should be able to take what you like and leave the rest. But we are different. We operate a service that millions of people around the world love, and they want to be able to sit down and watch their favorite series, and they want to watch the entire thing all in one sitting. So, that's a little bit about how we do things. It's a little bit different than some other places that you might find, but we certainly think about resilience engineering extensively in the sense that resilience is about amplifying what's already there. You have talented engineers who know things about the system –how do we get that information and amplify it?

You have resilience engineering, another example of that is when you look at an incident and you see how the team's built a script on the fly to solve a problem. They stepped up and had some little bit of information that without that, the incident would have gone on longer. That is what we mean by resilience, and can you engineer it? That's an open question. Dr. Richard Cook, another giant in our industry, has a wonderful talk called, "The Resilience of Bone," and I think it very beautifully explains resilience engineering, but it's another flavor of human factors that, again, looks to discovering and amplifying the strength in your teams that is already there.

We're not adding it in, right, and it's not a tool. Now, there are teams that build tooling, and they end up calling it resilience, and that's fine. I don’t want to say that that's wrong, but resilience engineering is a practice and a field of study that's more than 20 years old, and we should, I think, invest in learning about what it means, and we don't want to –there's blog posts here and there that say, that write about resilience engineering, and what they wrote has nothing to do with resilience engineering. So, it's just more of that education piece and evangelism and how can we share that knowledge that people can understand what it means.

Mike Kavis:
Yeah, so back to previous discussion we were having, you talk a lot about some of the ways we used to solve problems or (Inaudible) instance don't apply anymore, and it's funny, I was just looking at a deck the other day that was pushing the five whys. So, your whole talk was the five whys and the lies about complex system failure, so talk about that. I won't dig into your thunder there, and you can probably say it better than me anyway but talk about why that is.

Jessica DeVita:
Sure. So, that talk was one I've been wanting to do for quite a few years now. Just go on LinkedIn any day and you can search and find lots of people still sort of peddling this approach. But as I mentioned in my talk, it's completely insufficient. It does not work for complex systems. It might work for some simple three-part system. I'm trying to think of, like a vacuum cleaner. I'm trying to imagine a system that the five whys would work for. And to summarize the issues with it, it's still a linear accident model, and it assumes that there's only this single path, and ultimately if you just ask these five questions, you'll suddenly get to the root cause. We know that there is no such thing as a single root cause, so that any effort that simplifies post-incident analysis down to these five questions is just inadequate.

And it doesn't tell you anything you don't already know. And if you take two people and you put them in two different rooms and you have them go through the five whys, they will not be the same. They will be completely different. And lots of really smart people have written this. Dr. Nancy Leveson from MIT has a beautiful paper that describes the issues with the five whys. And I think ultimately, we have to stop lying to ourselves and using that method. It simply doesn't work. And if you take even a step towards being more curious about it, I don't think you could find anyone who would find value in it if they were willing to take a close look. The issue, Mike, is that it's very seductive, isn't it? Oh, I'll just follow this method, this fishbone diagram, and, oh, it worked for Toyota. But we're not building cars. We're not building an assembly line, right. Nothing about the systems we build and maintain are uniform and predictable that would even make that an appropriate tool for analyzing a system that has failed.

Mike Kavis:
I was always amazed how this magic number of five came about. I mean, couldn't sometimes it take maybe three questions, sometimes two?

Jessica DeVita:
Right. Is it seven? Is it twelve? And also, let's fundamentally look at the word why, at the question why. If I ask you why the answer is because. You are explaining something. In post-incident response, I don't necessarily want your description. I don't necessarily want an explanation for something because it brings forth a defensive response. Instead, I want descriptions: “What did you see? How did you get alerted?” We have to move from why to how, and John Allspaw has written a great blog post called, "The Infinite Hows." And then my colleague, Lauren Hochstein, also has another article which I'll send to you as well. But I think, fundamentally, if you move away from the why question, you have a better shot at learning more from this incident.

Mike Kavis:
My take has always been the why is like you're on trial, so you're trying to get to blame, or as you say, root cause as if there's one total thing, and a lot of people are defense trying to make sure it's not them that's the root cause. And the questions you were asking were what happened, and that's more scientific research like Sherlock Holmes, the sleuth trying to figure out the mystery. So, that's kind of what I took away from that is the whys kind of puts people defense.

Jessica DeVita:
Yeah. I've never seen someone be asked that question and not have a defensive response.

Mike Kavis:
Yeah, I've always got, “Why didn't you?” from management, “Why did this fail and why didn't you make it not fail?” I'm, like, “Uhhhh…”

Jessica DeVita:
Sure, and then there are some armchair quarterbacks saying, with their exquisite technical knowledge well, “Why didn't you do X? Why didn't you do Y?” And that's that counterfactual that's so harmful. You're saying why didn't they do something, and like that's an incident that didn't happen. You wanted it to go differently. Of course. But what did happen that counterfactual approach of why didn't they, it's just laden with blame. You might as well just cut to the chase and say you guys screwed up and – again, Dr. Richard Cook has a papercalled, "Those Responsible Have Been Sacked."

Mike Kavis:
I read this great book a few years ago called, “Black Box Thinking,” and it was about why people make up their mind despite facts showing other things, and it compared the airline industry to the health industry, where the airline industry, they're all about making sure a plane never falls out of the sky, and in the health industry, it's a little different culture because everyone's taking out huge insurance to make sure they don't get sued—the doctors, the hospitals.

And, so there's a lot of hiding and blame, which means that the same problems keep occurring. So, there was an example in that book of people dying or getting very ill from basic procedures, and it just keeps happening because the culture there is more, “I don't want to get sued for millions of dollars.” So, the reason why I bring this up is what you do at Netflix can only happen in a culture where it's okay to expose the warts, it's those types of things. So, can you speak to that a little bit?

Jessica DeVita:
Yeah, absolutely. So, you're getting at something really interesting which we – the collective we – think about a lot. You will hear time and time again, “We've got to find out what happened so we can make sure it never happens again.” And, look, it's going to happen again. It's not going to be a repeat. Incidents don't repeat, but there are themes. If you don't invest in monitoring, you'll have less ability to see into these systems later on. But this foundationally, full stop, if you think that learning about what happened is going to prevent something, I think you're going to be very dissatisfied, because,

oftentimes, you found the problem and you put in a fix. Just two weeks ago, it's like there was a fix that was a trigger for another incident. So, I think we just have to be extremely careful about thinking that we can prevent.

But to your point about culture, there's a lot of workplaces out there where it is not safe to talk about what happened. And you'll go before some kind of room with 27 people sitting around pointing at you and asking those why questions and, like, there is deep and abiding fear. And those workplaces where you have to live with that, – I think it's just absolutely so harmful to our customers; it's harmful to the engineers who, by the way, the day before saved your bacon. And, so I'm really proud of the culture at Netflix that allows us to share our lessons learned so broadly. I've just never worked at a place like this. I'm just so thrilled that we have that approach, and I hope others will strive for that in their workplace.

Mike Kavis:
So, one of the slides you had in that deck was, “My answer to each of the five whys is because local rationality,” which I thought was clever. So, explain that to the folks who weren't there to hear you talk to that page.

Jessica DeVita:
Every action that engineers and operators take makes sense to them at the time, given what they know, what they see, the signals that they get from disparate systems, and their prior experience. Local rationality means that we accept that, that they did what they did because it made sense to them. If it didn't make sense to them, they would not have done it. And when we question why they did something and we don't learn the lessons of local rationality, the issue is you're dealing with all of these things through hindsight bias and outcome bias. Those engineers did not know the outcome. That was unavailable to them. Now, sure, you can look back and see, “Oh, well why didn't they take this path?” But in the moment, they can't see the outcome.

And by the way, if what they did – if it didn't make sense to them, you're not really – at that point you might be dealing with a criminal situation where if they maliciously did something, that's a different story. But local rationality means honoring the expertise of our people and that what they did seemed like the right thing to do at the time. And Sydney Decker explains local rationality really eloquently in his book, “The Field Guide to Human Error.” And there's an illustration in here that, essentially, from inside the pipe, when the engineer's looking, they can just only see what they see, but outside and afterwards, you now, it's like, well, it's all about hindsight and outcome bias at that point. But ultimately that's local rationality. You have to accept that people did what made sense to them.

Mike Kavis:
Yeah, and I think the other thing is if we tried to design for everything possible that could never happen, we'd never get code out the door, so you design for what you've experienced through your lifetime and in the time that’s allowed for you to deliver software, so it's a subset of that, and then you incrementally improve the system. So, it would be kind of foolish to expect things to be flawless.

Jessica DeVita:
Well, it's impossible.

Mike Kavis:
Yeah, and people or other systems are going to use systems in ways you never anticipated, so software is a continuous learning process to begin with. So, why not create a safe environment to learn, right?

Jessica DeVita:
Sure. Sure. And you can, to your point, you can design the most beautiful system, but as you mention, people will always use it in a way that you didn't anticipate. It will interact in ways with other systems that no one can predict. And there's a gap between your design, and it's like this last 10 percent. And what's happening there is that operators and engineers are making up for the gaps in your design. They have to work it out, right. You can have all this beautiful documentation all day long, but again, this design gap is that we're always completing that design in the field, in the context.

Mike Kavis:
So, last question. So, there's a lot of companies out there trying to implement some form of resiliency engineering. Everything's a journey, right. It isn't one day Netflix woke up and there it was, right. It's a journey. So, what advice would you give someone who's trying to grassroots lead that effort up through the organization? What are some of the key advice, key points you should give them to help them along their journey?

Jessica DeVita:
Sure. So, we'll start with, first of all, in trying to help someone go on this journey, there's a lot of unlearning that they have to do. You have to unlearn that human error is a cause. So, my colleague when I was at Microsoft, Nick Stenning, he and I gave numerous talks about this, but human error and the belief in that as a cause is the first thing you have to disabuse people of that belief. But how you go about that, I don't recommend throwing a book at them, I don't recommend arguing with them, and I certainly don't recommend sending some academic paper, of which we have many. But I would encourage you to be curious and start asking your colleagues, “Hey, so that incident we had last week? They said that the root cause was Joe clicked the wrong button. I'm curious what do you think about – do you think there's more to it than that? Because I've clicked that same button and it's never caused an outage before.”

So, Step one is curiosity and small conversations, one at a time. You can't change the world. You're not going to turn some culture around instantaneously. So, I would start with your colleague, then you hope that they'll start another conversation with another colleague. It's one conversation at a time, and I suggest starting with the human error conversation, just exploring that. And when someone tells you that that was a cause, approach that with curiosity. “Tell me more about that because I've clicked that same button and it didn't't…” Find conversations like that. The second piece is, learn about local rationality, and it's related to human error because it's like you have to accept that what they did made sense to them.

A couple more tips, which I've mentioned to you, mechanistic reasoning. Again, thinking that this one little component, and if you can just pull out that one part and look at that, that, somehow, you'll have the answers you need. I call it the meddling kids trap. It's like, “My plan would have been just fine if it wasn't for those meddling kids!” And another piece is perk up your ears when people say, “They did a really good job!” or, “Boy, they really mishandled that outage.” That incident it goes to blame. But ultimately, when you use “good” and “bad” and that type of normative language, you're doing that with the benefit of hindsight, which again, those people did not have.

Now, all of that said, pick up this book, “Field Guide to Human Error.” It's an easy read and it will help you in that journey of unlearning that I talked about. And the second piece would be there is a very brief paper called, "How Complex Systems Fail." That's by Dr. Richard Cook. Those two pieces should get you started. You're not going to boil the ocean, though, and you're not going to change the culture overnight. Because remember, Mike, culture is – it's you, it's me, it's the conversation we have, it's the e-mail I sent you. And we're always a mirror of the culture in a company. And that comes from a book I love called, “Fierce Conversations.” And I would pick up this book too, “Fierce Conversations,” because I think it gets to the heart of what I wanted to say, which is, like, let's set all this stuff aside and just talk, and let's just be real and talk about what our fears and concerns are and talk about making things better around here.

And I know I've gone on and on, Mike, but take a look at this concept called asset-based community development. Instead of starting with what's wrong, you have to start with what's strong. What is going right in this team? How did we show up in that last incident and really adapt? How do we amplify the resilience on that team that's already there?

Mike Kavis:
Yeah, what's the term a lot of folks use about incidents or opportunities? I don't remember the exact terminology, but they're like learning opportunities.

Jessica DeVita:
Sure. So, John Allspaw and Dr. Richard Cook have a company called Adaptive Capacity Labs, and a key thing that they have said and that I think conveys it so beautifully is you already had that incident. You didn't plan to spend that money, but you sure did. So, are you getting a return on that investment? And I don't remember the source, but there was somebody once who said the CEO said we just had a $1 million incident; can I get $1 million of learning out of it? And the point is you've made the investment already, even though you didn't plan on it. So, view it like that is, I think, Allspaw’s point. And, yeah, strongly encourage you to look at it that way.

Mike Kavis:
That's definitely a culture change.

Jessica DeVita:
It sure is.

Mike Kavis:
So, really good advice, and I've had the luxury of seeing you on camera pointing to pictures. And where can we find you on Twitter or if there's any blogs or slide shares or anything? Where's a good place to get your content?

Jessica DeVita:
Sure. So, you'll find me on Twitter under @ubergeekgirl. You might find some spicy hot takes there from time to time, but yeah, reach out and start a conversation. I'm happy to chat with anybody about these topics at any time.

Mike Kavis:
Alright, well greatly appreciate having you today. That's our show for today. You can find more podcasts by me and my colleague, David Linthicum, just by searching for Deloitte On Cloud podcasts at iTunes or wherever you get your podcasts. You can see this podcast and show notes, head over to www.deloittecloudpodcast.com. I'm your host, Mike Kavis. If you'd like to reach me, you can find me on Twitter @madgreek65 or just shoot me an e-mail at mkavis@deloitte.com. Thanks for listening, and we'll catch 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.

Visit the On Cloud library

www.deloitte.com/us/cloud-podcast

About Deloitte

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.

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.

Additional resources / content referenced during the podcast:

Lund University MSc. Human Factors and System Safety
Netflix SRE
Jessica DeVita 5 Whys and other lies talk and slides
Fierce conversations
Field guide to human error
Lorin Hochstein
- Why you can't just ask why
- Intro to Resilience Engineering

Adaptive capacity labs
John Allspaw
The Infinite Hows
Dr. Richard Cook
How complex systems fail