Architecting the Cloud
May 2020

# Architecting the Cloud, part of the On Cloud Podcast

## Mike Kavis, Managing Director, Deloitte Consulting LLP

**Title:** **Software testing in the age of cloud—new ideas and opportunities**

**Description**: Testing has always been a critical component of good software development practice. However, it's often short-shrifted or siloed. As a result, app performance can suffer. In this episode of the podcast, Mike Kavis and guest, mabl's Lisa Crispin, discuss the evolution of testing and how new development ideologies like DevOps have shed light on the importance of testing. Lisa also talks about how the cloud has transformed software testing by making it faster and automating much of the process, so that testers can focus on crucial human-centric testing such as security and accessibility testing. Lisa also touches on how cloud-native apps changes the testing process and gives her advice on how companies can learn more about testing and use that knowledge to improve their software delivery process.

**Duration:** **00:23:48**

**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 back to Architecting the Cloud Podcast where we get real about cloud technology. We will discuss what's new in cloud, how to use it, and why, with the people in the field that do the work every day. I'm Mike Kavis, your host and chief cloud architect over at Deloitte. Today I'm joined by Lisa Crispin. Lisa's a coauthor with Janet Gregory of four books on agile testing, which she'll talk about here shortly. She was also voted by her peers as most influential agile testing professional person at Agile Testing Days. She's a testing advocate, works at mabl to explore leading practices and testing in the software community. So welcome to our show. You are the second ever tester on my show.

Angie Jones was the first, and actually that's probably how I wound up getting connected with yourself. It opened up a whole window of really smart people in that area and I've been following ever since, getting smarter in testing. So tell us a little bit about yourself and tell us about these great books that you wrote.

**Lisa Crispin:**
Well, I'm honored to be on the podcast. It's always an honor to do anything Angie did. She's awesome. And as you said, I've coauthored several books related to testing and agile development. The first one was way back in 2002. I first joined my first extreme programming team in 2000 and coauthored a book on testing extreme programming. And then later on Janet Gregory and I paired up and wrote our book, “Agile Testing,” followed by, “More Agile Testing.” And our latest book, just out last fall, is, “Agile Testing Condensed,” because our first couple books were 550 pages each, and we felt people might like a shorter version to read to get a nice introduction. So our newest book is only 100 pages. And I've been a hands-on tester on agile teams all that time, and my career in IT goes way back beyond that.

But I currently am also a testing advocate for mabl, so it gives me a chance to learn what's going on in the community with regard to testing, and I do feel like the cloud and DevOps and continuous delivery, that's where the future is. And so I'm trying to help testers get involved in that, and I curate a community website called TestingInDevOps.org with all kinds of articles and podcasts and webinars and so on – resources for testers to learn about DevOps and continuous delivery and how testing fits in with that, and also a lot of awesome guest blog posts from people involved in continuous delivery and SRE and all that great stuff. So that's my passion for right now.

**Mike Kavis:**
Cool. Well, those are all the topics we talk about here. I’m going to start with the Delivery Conference, the first one that happened out on the West Coast. Where was that? Somewhere in California, right?

**Lisa Crispin:**
Seattle, Seattle.

**Mike Kavis:**
Seattle – okay, Seattle. And I followed a lot of it when they came out. Unfortunately, I couldn't be there. But that was a pretty interesting one for me because it was all about the pipeline, all about delivery, deployment. And the amount of people in your profession, in QA and testing that was there, made me pretty happy because there's this misconception that once we automate everything in our CI/CD pipelines we don't need testers anymore. And this conference was loaded with people with great ideas on how to improve the overall process. So tell us a little bit about your experience there. Then I want to go into talking about your presentation there, which was about the importance of the whole team ownership when it comes to quality and testing.

**Lisa Crispin:**
Yeah, it was exciting to see so much emphasis on testing and quality at this conference, which was the first conference – I think maybe the first conference devoted to the technical aspects of continuous integration and continuous delivery, and a lot of the leading lights like Jez Humble and David Farley, Liz Fong-Jones, Nicole Forsgren, were there, all my heroes. It was really exciting. And quite a few testers – Jez Humble and Maryam Umar also did a talk on quality and how to measure quality, which is something we all struggle with. Like, how do we know we're delivering the quality that we want, and how do we know we're building it in? And it was really interesting; we were running the range from pipelines for legacy systems, a lot of stuff on security, using value stream mapping with our pipelines, and things like that.

So, you know, it's all related to quality, of course, and I was happy to see a lot of testers attending. So testers are getting interested in this, and, you know, everything that's new. Back when extreme programming was new, they thought they didn't need any testers either. [Laughter] And I'm all for generalized specialists, but we do need specialists, even in the DevOps culture where everybody's responsible for the product while it's being created and coded and tested and after it's delivered. That all involves testing, and having some testing specialists to help the team learn all the different skills they need for all the different testing activities I think is really important.

**Mike Kavis:**
Yeah, your talk. I listened to it. It was pretty interesting, and you know, it was all about we all own quality, not just this team or this person over there. So tell us about that.

**Lisa Crispin:**
Well, of course the whole team approach to quality dates back to extreme programming, so it's not a new concept. And DevOps I feel like was kind of a response to – originally agile development was intended for small, co-located teams, and it tended to be smaller companies that adopted it. And I always worked on cross-functional teams that included responsibility for releasing and operations, so we were always responsible and –as well as building the product, we were watching it in production and making sure everything was working well. But in larger, enterprise-type companies, the silos remained, and a lot of software delivery teams were just heaving code over the wall to the operations team saying, "Okay, your problem now. Get that released and in production and take care of it. We're going to go on and build the next feature." And DevOps's response to that was to say, "No, we need to work together," especially now that infrastructure is now code. Everybody needs to learn to code.

And the people delivering the software, the people coding it, need to learn about operations-side things. They need to be concerned with operability. They need to instrument their code so it can be monitored in production, so it can have observability into how our customers are using the product and what problems they might be having and how we can fix those quickly. So we need everybody working together, because you need a huge range of skill sets to make that happen. And while we can all get some kind of meta-knowledge about everything, nobody can have deep knowledge of every single specialty. So having this diversity of people on the team with different experience, different skill sets, I think it's really critical for continuous delivery, because there are so many challenges, and we need everybody's ideas on how to overcome those challenges.

**Mike Kavis:**
Yeah, so there's a lot of talk, and there are some recent really good books about moving from project to product, right? And I do a lot of work on ops model – what's the right operating model to really do what you just talked about. It's hard to do when we're all aligned in our domain silos with our own goals and

objectives, right? So what we're seeing is full stack teams, those types of things. So from your perspective, what are those new operating models that you're seeing? Is there still a QA center of excellence but people are embedded in teams? What are the different ways you're seeing companies as they mature through DevOps and CI/CD pipelines? How are the QA resources being integrated into teams? What have you seen?

**Lisa Crispin:**
Well, personally I've worked in unicorn land for the last 20 years, and so I've been on cross-functional teams that have been able to get rid of those silos. But I definitely see lots of companies, especially larger enterprise companies – they haven't really made that transition. They want to. They know it's the good direction to go. We know that handoffs involve a lot of waiting and queuing time, and that's waste so we don't want that to happen. We need to be able to deliver things to customers. Our goal is getting small changes out to our customers frequently, like continuously, and so that each small change is a low risk because we're not changing much at once. But we need to do that every day or several times a day. And we do need to get everybody involved in that.

I still see a lot of anti-patterns. I mean, it's all context-specific. There may be some domains, let’s say safety critical, life critical, money critical domains. Maybe they need those separate quality organizations kind of overarching to make sure every team is doing what they need to do. When I hear people talk about having their DevOps team, that's often a red flag to me because DevOps is meant to be the culture of the whole organization, the whole IT organization. But, you might want to have DevOps specialists to help make that transition to building a DevOps culture because it's not easy to do.
So, you know, it's hard to really judge when you look at an organization and see what they have, but you don't know where they are in their journey. You don't know what their business domain is and what their risks are. And I think it's really important that every organization take a look at their risks, take a look at what's valuable to their customers and what customer problems they're trying to solve, and organize themselves accordingly and keep experimenting with new approaches. And hopefully that leads to more people collaborating and fewer silos and handoffs.

**Mike Kavis:**
So two things you said, where they are on their journey, and experiment, and that's what people need to realize. You've got to start somewhere and start shifting some things left and learn and keep continually changing till it gets better and better.

**Lisa Crispin:**
Yeah, one of the main messages I brought away from Delivery Conf – and it's not a new message to me, but I was really impressed with how often I heard it. At just about every talk they emphasized small steps, one small step at a time. Just try to get better at one little thing. Just try to overcome—make one problem smaller at a time. And I think that's really good advice, because when people make these big bang transformations, then it kind of leaves everybody in a state of confusion and they're sort of dazed and don't know what to do next. So by doing small experiments that don't take very long, then if they fail, that's good. We learn something and we can move on, but we didn't invest a huge amount of time and money in it. So I was really heartened to hear that from so many expert practitioners.

**Mike Kavis:**
Yeah, the only thing I'd caveat – add to that – so again, I do a lot of operating model work for people, and there's a technology and a people and process component. And everyone just does the technology piece because it's easier. So yeah, they'll do small changes, but they often don't do the changes that really move the needle, right? They do the low-hanging fruit changes, and that’s part of the goal right? If you're not working on the biggest bottleneck, you're just shifting the bottleneck somewhere else, and that's what I see a lot. So yes, you should do small experiments, but you should do the right ones.

**Lisa Crispin:**
Yeah, that's a really good point, and one of the things I think testers bring to the party is we tend to be big picture people. So developers, you know, because of what they're doing, they're focused in one small part of the product at a time. They're writing code for one thing. It's harder for them to back up and say, "Well, how will this impact the rest of the system?" And testers, I feel like we're always thinking about how is the customer going to use this, what's the big picture here, and I think that's one skill that we bring or one perspective that we bring that's really, really helpful. We ask questions about those kinds of things that other people are so deep into what they're doing, they may not think to ask those questions. And the same for on the operability side, we can spot patterns in things like production-log data that maybe nobody else sees right away, because that's one of the things we like doing, is looking for weird patterns and asking the questions about them. So I think that that's a great set of skills and perspectives that we bring.

**Mike Kavis:**
I agree. So the next set of questions are all about – all right, we've talked about DevOps and all this, and then there's cloud, right? What are the impacts of cloud technology on testing and quality? And we have a bunch of topics within that, but let's just start with the testing environments. How does cloud change how we think and use testing environments?

**Lisa Crispin:**
It solves such a huge problem. [Laughter] You know, for decades it was like – I mean, I was on – I joined teams that didn't even have a test environment yet. And then the test environment would be broken, or we couldn't get a build to deploy to it, or the database was messed up in some way, or everybody's fighting over the same test environment. It's like, "I want to put this version of the software on there. No, I want to put this version." And we just had such limited resources, even with virtual machines – you know, my last team, we had three test environments for a team of 30 developers. That seems okay. Well, we had a lot of different features being built and there was a lot of contention, or maybe somebody wants to test a bug fix, so they need to deploy an older version of the product.

And when we moved to the cloud and all of a sudden now we can just spin up a test environment for any story branch that we want to, and have exactly what we want in that environment, and it happens automatically, it's just transformative. It just totally removes that worry. It's like, "Okay, here’s a poll request and here's the previous environment where I can go and test it. I don't even have to think about that." Or it might've – at one point it was hours or perhaps even days of delay, waiting to test some change. So to me that's just removed a huge obstacle for so many teams.

**Mike Kavis:**
What about the whole CI/CD process? So there's a lot of automation in place now including, you know, test automation. How does that change the game? And what does it make better and what does it make worse?

**Lisa Crispin:**
Well, of course you've always had automation, but one of the problems is we want short feedback loops. And when we automate, especially through the user interface level, when we need to have workflow end-to-end type tests through the user interface and all the layers of the application, those are going to be slow. They're driving a browser. And to me it's just amazing that these days we can spin up a container for each test and just run. If we've got 500 tests through the UI level, we can run 500 tests concurrently, each in its own container. And our feedback loop is only as long as the slowest test. That's just amazing. I mean, when you're trying to do continuous delivery and you want to get your pipeline short enough to where you can deploy every day, or if you have a hot fix you need to get to production, you can do that within minutes. The cloud enables us to do this, the ability to have these parallel test feeds and stuff.

Having the short feedback loops makes us so much more productive and so much more responsive. You know, we don't wait days or weeks to find the bugs, at least the regression failures, and it frees up our time. So now we have time to do the exploratory testing to find the really serious unknown unknowns we didn't think about upfront. We have time for manual testing, or I like to call it human-centric testing, where we can cover things like security testing and accessibility testing, which you can't really automate all of that. So by automating all this drudge work, we have time to do the important thing that we need our human brains and eyes and ears and fingers for. And again, that technology is just transforming, and we need to take advantage of it. It's not that hard to get into place. There's lots of ways to learn how to do it and lots of automation that makes it easy. it's kind of a little tester utopia, really, [laughter] because we can focus on what we really need to focus on instead of focusing on the infrastructure around it.

**Mike Kavis:**
Yeah, it's nice to hear you say that. You know, we hear the same thing on the server side. People who are afraid for their jobs, and we keep saying, "No, the mundane, repetitive, low-value stuff is getting automated. Now you can use your brain to solve the bigger problems." And that's basically what I just heard in testing. You know, the manual tests that may not be all that exciting, that's getting done for us and now you can do exploratory and all these other types of testing you mentioned that sound like a lot more fun.

**Lisa Crispin:**
Yeah, and now we know our test environments are never going to look like production, even with today's technology. We probably don't have load balancers in them. And also we don't know exactly how real people are going to use our software. So now we have all this capability to do dark launches or use – release feature toggles to get stuff in production, turn it on just for ourselves and do testing, turn it on for a few customers and watch what they do. We can watch exactly what they do – kind of big brother-ish, I guess, [laughter] but we can see what people do. And if we have a problem, we can revert the change so quickly, or get a fix out so quickly, that we minimize the customer pain, and then we can start doing things like chaos engineering, chaos testing. You know, we can have enough safety in production to do those kind of, you know, crazy – let's make one server fall over and see what happens, or let's delete a database table and see what happens. We can do that safely, in production, and that's a whole range of things that we didn't have the time or the technology for before. So that really extends where we can do our testing and ways to improve quality even more, and just make the customers' days better because they're not running into these bugs all the time.

**Mike Kavis:**
One other question in this area is as we move to more cloud-native applications, these architectures are becoming more distributed, more complex, right? So you used to have – you know, you had a database server, you had a web server, and you had an app server, and that was it. And now you're sitting here with 100 microservices running, and 100 containers, and maybe using four different databases because they're just an API call now, right? So I may be using a graph database along with the regular – a SQL and a no-SQL. So really, really complex stuff now – how does that change the game when you have to test all this stuff?

**Lisa Crispin:**
I think that's where observability comes in, and observability is – because we're instrumenting our code, we instrument every event that happens in our code, and we don't aggregate our logs, so that when something happens in production, we have the tools now to trace exactly what did a user do before they got that error? And because it's so complex, we should keep trying to test obviously, but we know we can't test everything, and so we need to be ready to quickly identify those problems before the customers see them. Or if the customer does experience it, we can be right on it and get a fix out, revert the change, turn the feature flag off, whatever we need to do to quickly take care of that.
And I'm a real fan of Tierney Majors and Cindy Strahan, who've done so much pioneering work – oh, and other people, too, Liz Fong-Jones – with observability. And I think Sarah Wells – I can't remember if it was Sarah Wells, or Abby Banks, or who said this. But observability is basically tool-assisted exploratory testing in production. So we can go and ask questions and find out the information that we need to answer those questions by the telemetry in our code and by all the wonderful tools that we have to help us analyze these huge amounts of data. You know, we have the capability to store the big data, but we need help analyzing it and I think things like machine learning are coming in to help us with some of that analysis. So I think that's the whole new field for testing and for where testers can add value, and I'm super excited to learn more about it.

**Mike Kavis:**
Yeah, it's really cool stuff. The last question – so like you say, you've been in unicorn land for a while. You've been working on this stuff. You're very far along in your journey. What would you tell a colleague of yours who's just starting this journey? You know, the first question I always get is where do I start? So what would your advice be to someone who wants to lead change in an organization to move towards this world. Where should they start?

**Lisa Crispin:**
Ooh, that's a good question. We have so many resources available. Obviously, familiarity with test automation is really important, so even if you're not going to do the automation yourself, you should probably be involved in it. I don't write a lot of code anymore but I'm really good at specifying tests. So if I collaborate with the developer, I can do a great job at test automation. But you mentioned Angie Jones. She's pioneered Test Automation University where you can just go take these amazing free courses to help you in all kinds of aspects of test automation. So that's one place. Ministry of Testing also has a lot of wonderful online material, master classes, all their conference sessions and webinars and things like that. There are all kinds of online courses that you can get.

But I think you have to kind of follow your passion. It's like, what's most interesting to you? Because it's just a vast area. So do you really enjoy security testing? Because that's a huge opportunity there. That's growing all the time. Accessibility testing – I saw an amazing session at the TSQA Conference in North Carolina with – oh, Jenna Charlton did a great session on accessibility, and the tools that we have to do accessibility testing. So you can find – whatever you're most interested in, you can go and find plenty of resources to learn it and get involved in it. And, you know, I would say try a bunch of different things out and see what interests you the most. And of course when you're on a team you need to ask the team, "What's the biggest problem that I can help work on?" and collaborate with your teammates to start doing experiments to make that problem smaller. So a lot of different avenues to approach that.

**Mike Kavis:**
Cool. So that's our show for today, Architecting the Cloud. Thank you, Lisa. I've got some of your websites here people can check out: www.LisaCrispin.com and www.AgileTester.ca but more importantly where can we find you on Twitter?

**Lisa Crispin:**
Well, I'm just Lisa Crispin on all the social media, so @LisaCrispin on Twitter and I'm Lisa Crispin on LinkedIn. So always happy to talk to people and hear other people's experiences. I love to learn from other people, so you know, ping me on Twitter or LinkedIn and we'll have a conversation.

**Mike Kavis:**
Cool. I like to learn from other people, too, and that's exactly why I do podcasts. [Laughter] I just sit back and listen, right?

**Lisa Crispin:**
That's great. Well, I listen to podcasts all the time. I have donkeys, so when I'm out in the barn doing chores, cleaning up after the donkeys, I like to listen to podcasts like this one. I get to learn a lot and get inspired a lot that way. That's how I got interested in things like observability and machine learning, and so it's a wonderful resource.

**Mike Kavis:**
Cool. So you can learn more about Deloitte or read today's show notes. Head over to www.DeloitteCloudPodcast.com. You'll find my podcast and my friend Dave Linthicum's just by searching for Deloitte On Cloud Podcasts on iTunes or wherever you get your podcasts. I'm your host Mike Kavis. If you'd like to contact me directly you can reach me at MKavis@Deloitte.com, or find me on Twitter, @MadGreek65, and thanks for listening. 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.

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.