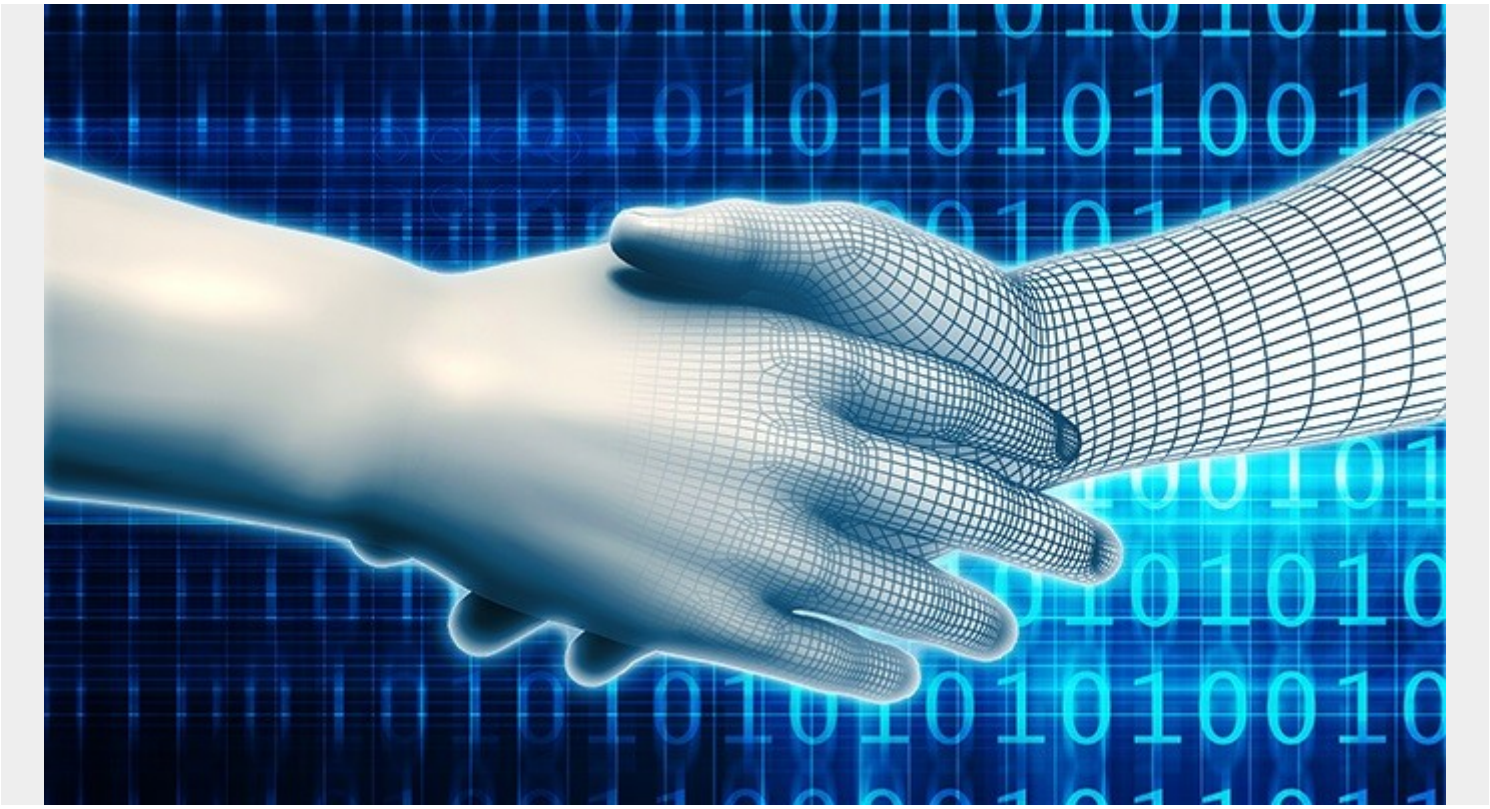
WHEN CULTURES CLASH - REMOVING FRICTION BETWEEN DEV AND OPS



Eliminating as many bottlenecks as possible in the Software Delivery Lifecycle is essential to competing effectively in the Digital Age. Dev teams must continuously deliver new features and services their customers demand, while Ops teams need to predict and solve problems as quickly as possible, ensuring optimal performance, availability and security. Achieving this state of equilibrium is next to impossible in siloed organizations with staid cultures—a breeding ground for friction and mistrust. In this episode of BMC AMI Z Talk, BMC product management leaders April Hickel and Sam Knutson discuss how modernizing tools, processes and culture, including adopting mainframe-inclusive DevOps, can help organizations continuously deliver high quality apps and services to market more quickly. Below is a condensed transcript of our conversation.

April: Sam, you've had a lot of experience working over the last six years, helping large enterprises adapt Agile and DevOps on the mainframe. Can you share with us what the early days were like?

Sam: When I started at Compuware six years ago, enterprises generally undervalued the mainframe. There was a lot of discussion of re-platforming, re-hosting. There were still people who thought rip and replace might be the best way to modernize the mainframe. And, there was some good reason for this. You saw a lot of frustration where organizations were internally silent, so that mainframe application development had been left unchanged since the 1970s or 1980s. It was purely waterfall. Very, very heavy project management.

And, those groups were kind of the masters of know, while these same organizations, who work in a very competitive landscape, were confronting opponents who were being successful with some of their digital transformation. They wanted to do this themselves. They saw the opportunity in some of these global industries that are really driven by mainframes, but they couldn't emulate what was being done by some of the small start-ups. And, the start-ups really had paved the path with some very worthy techniques, using Agile development, DevOps methods to build a continuous delivery of new capabilities that customers loved, and to really develop product management to an art.

And, what we saw on the mainframe was a very apathetic landscape, where there wasn't a feeling that customers had great choices. And, they didn't see the mainframe as modern. And, of course, they thought this very wrong because the mainframe was, in fact, a platform that has evolved – from a hardware and operating system perspective – very, very continuously. IBM has been a good steward of that ecosystem. But, what hadn't changed was the way that develop – that customers built their applications – the way they treated their developers.

And, in particular, their mainframe developers were often the side organization that worked in different ways from the developers who worked on systems of engagement – their web front-ends. And, those developers in the very same company, often, had adopted the tools, and techniques that did allow them to go faster. So, they were building software in small iteration.

So, it wasn't that the companies didn't know how to do this; it's that they hadn't applied those techniques to their mainframe. They left it in its own silo. And, there was just that there was a lot of apathy, and so people tended to either do nothing, or do the wrong thing – investing in solutions that really didn't provide any savings, or competitive advantage. And, they kind of saw their mainframe as a cash-cow in the basement that just did the record-keeping, and they couldn't do any better. And, the only option was to replace it. And, it was not a happy time.

April: Given that context, I think it's really interesting how you looked at the struggles that customers were having, and you helped pioneer the idea of mainstreaming the mainframe. Can you talk a little bit about the changes that have happened?

Sam: So, that approach that we did bring to the market is one that we internally did a lot of those techniques ourselves. And, as we learned what we needed to do to change the way that we were building software for our customers, and we started to talk to customers who had the same challenges, we realized that there was a broader concept that everyone needed to get behind. And, that was this idea of mainstreaming the mainframe; making it different only in syntax, so that all of the goodness that had evolved in the development ecosystem, outside of the mainframe, could be brought to the mainframe.

And, that you could on-board next generation developers who typically, when they looked at the very old archaic processes that surrounded mainframe development, and the tools that were given to them, made them just kind of run the other way, screaming. You know, they didn't want to work on a green screen. They didn't want to fill out three-part paper forms on changes. It was those things that were repellent to them; not the actual syntax.

April: So, can you share what you've seen in customers who have started down the Agile and DevOps adoption path?

Sam: Absolutely. And, we've seen customers who've gotten in the boat with us on mainstreaming the mainframe, and they've really realized that they have to go away from these processes that they've had, and embrace Agile and DevOps. And, my mission has been to enable Agile and DevOps

on the ZOS platform. And, I can tell you that, when I was talking to customers originally, you know, say, again, going back to 2016, when we were going through these changes, the conversations would often be, "Why should I do that?

Why should I use Agile? Why would I want to do DevOps?"

And, when they saw that there were outcomes that could be derived from this, and they saw those outcomes both from Compuware, and from other customers. So, Compuware, by adopting Agile development, was able to deliver roughly twice as much code per developer. We were able to deliver in quarterly releases, so that our customers could adopt these new capabilities that they needed much sooner, and get value from them. These smaller iterations were much easier for them to manage. And, we were able to significantly reduce trapped bugs – bugs that never escaped the lab.

Because, we wanted to find them, but, you know, developers will never write perfect code, but by adopting shift-left techniques, but putting up all of the quality automation into a CD pipeline, you're able to help the developers deliver quality production. And, we've definitely seen that in our customers' efforts, as well. And, some of them – I'll give you a good example of a customer in the UK that adopted automated testing. And, they were able to significantly reduce bugs by more than 80 percent; cut their delivery time by a third; they've been able to bring new graduates to work in their mainframe environment.

Over 150 developers that they've on-boarded, and they figure that they saved about three months, per developer, in terms of on-boarding time, and have saved over $21 million in a three-year period. And, by the way, that's not my number. That's a number that was derived by Forrester Research, who did a study with the customer on the total economic impact of automated testing. So, they've seen outcomes that matter to them by applying these DevOps techniques, including automated testing, to the mainframe

April: So, as you think about the evolution of development to DevOps, there's a link between development and delivering that code, and operations. Can you talk a little bit about how those organizations, or teams within customers, are working together?

To listen to the rest of this episode, visit SoundCloud or Apple Podcasts