On Cloud Podcast
March 2025

The Deloitte On Cloud Podcast

Gary Arora, Chief Architect of Cloud and AI Solutions

Title: **Platform Engineering, AI, and culture: Google Cloud's Nathen Harvey helps explore the new DORA State of DevOps report**

Description: In this episode, Gary Arora is joined by Google Cloud's Nathen Harvey and Deloitte's Eddy Krumholz to explore the latest DORA State of DevOps report. They explore the evolution of DevOps, platform engineering, and AI's role in the SDLC. They also discuss the "AI trust" gap in leveraging AI-generated code, the evolving challenges of AI and cloud adoption in a dynamic environment, and why culture remains a key differentiator of DevOps success.

Duration: **00:23:54**

Gary Arora:
Hey, everyone. Welcome back to the On-Cloud Podcast. I am your co-host, Gary Arora, chief architect for cloud and AI solutions here at Deloitte. Today, we are talking about the DORA 2024 report, also known as the State of DevOps report. This is their 10th year investigating high-performing technology teams and organizations, and this year they have all kinds of surprising insights into how to make your teams more effective, the role of AI and GenAI in software engineering, developer experience among many of the topics. So, I am very excited to be joined by Nathen Harvey from Google Cloud, who is not just involved with the Dora report, but actually leads the Dora team. Along with Nathen, I have my friend and colleague Eddy Krumholz, a tech fellow and chief architect in our DevOps practice. Eddy, Nathen, welcome to the show.

Nathen Harvey:
Hey, thanks so much Gary, I am excited to be here.

Eddy Krumholz:
Thank you, Gary, likewise.

Gary Arora:
I've got to confess, there are very few things that I have read cover to cover. It's really hard to focus and apart from my sci-fi novels, which I will keep aside for another episode, the Dora report is something I do read cover to cover every year. I start with the graphs to see if there is anything really surprising, and then I read all 120 pages and I really like how thorough this report is. So, let me start with a core question, over the years, how do you see DevOps as a discipline evolve given the latest advancements in technologies and software engineering, because so much of DevOps used to be an afterthought in the bad old days, but now it's so woven into everything we do. So, what are the most significant shifts you have noticed along the way?

Nathen Harvey:
Well, Gary, for me and this is Nathen here, I think if we go back to 15 years ago when the DevOps movement first started, it was all about culture and collaboration, automation, learning and using technology to drive the business forward. DevOps started as this collaboration between Dev and Ops, but it has quickly grown well beyond just those two departments and it really covers any technology driven organization or technology driven team as you said, it's really woven into everything that we do.

Now, unfortunately, I think that the DevOps label doesn't really fit that pattern of being woven in everywhere and it is one of the reasons, frankly, that Dora is starting to move away from using that term DevOps so that we can look more holistically across the entire organization. That said, of course, there are lots of changes that have happened, but maybe a little bit more closely to your question, Eddy, what do you think?

Eddy Krumholz:
I feel that the evolution of DevOps has been really marked by significant changes in technology and software engineering practices that are transforming from an afterthought to more of a core component of modern software development. In my opinion, there are really a couple of forces around that. One is the transition from rigid to an agile and an iterative approach to development. The second one is the idea of having tailored delivery methodologies. Initially it was standardized one-size-fits-all and now it's more of different ways of delivery. The other is about next generation delivery and automation. Initially this was manual, siloed, very theme oriented, that was leading to inefficiencies and higher risk and today it's the high-performing distributed teams that are more collaborative.

Last is more like a proven track record and focus on reducing risk and cost. So, there are some key advancements in technology and software engineering as well. One is around the idea of platform engineering and developer experience. The platform engineering is seen as a critical component for reducing time to market and improving overall software delivery performance and it involves creating and maintaining the underlying infrastructure and tools that enable developers to build, test, and deploy applications.

On the other side, you have the DevEx side which is more of an emphasis in improving developer experience, streamlining processes, and integrating responsibilities across the SDLC. Others are AI and automation, which we are going to be probably talking about, where AI is driving tools that are increasing the integrated software development life cycle from test all the way to releasing and beyond even for SREs to automate repetitive tasks etc., and the idea of automation. Last but not the least is the implementation of holistic and user-centric approaches, which in a holistic approach, the developer experience includes more providing developers with the necessary tools for success and user-centric development, which is a combination of good documentation and user centric approach to software development, which is very powerful.

In my opinion, what I am saying is that the evolution of DevOps as a discipline has been driven by the need for greater agility, efficiency, and collaboration in software development. So, by adopting agile iterative approaches, tailoring delivery methodologies, leveraging next generation delivery and automation model, and focusing on reducing risk and cost, organizations can achieve significant impact through DevOps practices.

Gary Arora:
Well said Eddy. I am really happy with this level of penetration of Dev OPS into SDLC. The benefits are so clear and there really is no going back, especially now with AI that is already having a massive impact on software development, and that's a trend we expect to continue to grow. Now, in terms of AI adoption, the report found out that at an individual level, 75% of respondents are relying, at least in some part, on AI in one or more of their daily professional responsibilities. So, ranked in the order of writing code at the top, summarizing information at number 2, explaining unfamiliar code, optimizing and documenting code at ranks 3, 4, and 5. Now all of this tracks, these are the obvious use cases, but what surprised me was the trust in AI-generated code. Almost 40% of respondents reported little to no trust in AI-generated code. So, help me reconcile these clear benefits with such persistent trust issues.

Nathen Harvey:
I think there are a couple of things going on. First, you have got new tools. When you start using something new, it's sort of human nature that you have a little bit of distrust for it. I also think 40%, may be that the right amount of people that have little or no trust in it. I think if you think about what are the opposite ends, like the 0% trust, should you have 0% trust in the code that AI generates? If you do, you are probably not using it. Should you have 100% trust? Oh, that feels a little risky as well. There's probably somewhere in the middle that does make sense, kind of that Goldilocks of trust and then what's happening is that the tools are always improving.

The way that we interact with them, we are learning and understanding how to better interact with them. So, that's improving. So, how much should you trust the code that's generated by AI? Well, that's going to change over time. I think an interesting question also though is how you improve that trust and I think one of the ways that you improve trust in anything is through additional usage. As you use something, you begin to trust and understand what you should expect from it more and as you trust it more, you use it more and then you trust it more, and so forth; it creates this virtuous cycle. That cycle is really based on feedback. So, understanding what is it that the AI has generated for me? Do I have a good mechanism to evaluate if this is good or bad and some of that feedback mechanism might be look at the code that's generated. Some of it may come further down the line in things like automated testing and so forth. So, I think that over time, we will improve the level of trust that we have in these tools, but it's always important to maintain some healthy skepticism.

Gary Arora:
I like that healthy skepticism around new tools makes sense, and as you use them more, you get more comfortable. So, this will only improve. Not just that, I really liked was when an individual increases their AI adoption by 25%, their job satisfaction increases by 2.2%, their productivity increases by 2.1% and their flow which is how much focus a person tends to achieve during development task increases by 2.6%. For an attention-deficit era that we are living in, these are substantial benefits, and these are at an individual level. So, when you extend this across tens of developers or even tens of thousands of developers, that's a massive advantage. But AI is all about trade-offs. For the second year in a row, Dora found that AI tooling worsens software delivery performance. So, Eddy, let me ask you this, how do you make sense of these conflicting insights and what should leaders consider when scaling AI across their organizations?

Eddy Krumholz:
In my opinion, the adoption of AI software development has been very promising at the individual level. Levels such as this increases job satisfaction like you mentioned, productivity, and focus. So, when an individual increases AI adoption by 25% and their job satisfaction raises by 2.2%, and productivity by 2.1%, and their ability to achieve flow during the development task by 2.6%, these improvements are significant in our attention deficit era and when scaled across large teams of developers, they can translate into substantial organizational advantages. But despite these benefits, the findings, like you said, for the second year in a row indicates that AI tooling can worsen overall software delivery performance.

So, in my opinion, to reconcile these conflicts leaders should consider a balanced approach when scaling AI across their organizations. Firstly, it is crucial to ensure that AI tools are integrated seamlessly into the existing workflow without disrupting established processes. In the past they used to call this by model way of operating. This involves providing adequate training and support to help developers effectively utilize the tools and mitigate any potential downsides. Secondly, leaders should focus on enhancing the transparency and explainability of AI outputs so that you can build trust and confidence among developers, which is what Nathen was referring to before. Additionally, recommendations are to implement rigorous validation and testing framework that can help maintain the quality and reliability of all AI-generated code. By addressing these considerations and at the same time fostering a culture of continuous improvement, organizations can leverage the individual benefits of AI and minimize its potential impact on overall software delivery performance.

Gary Arora:
Nathen, anything to add here?

Nathen Harvey:
It's really interesting because we are seeing these sort of early stage or early in the process benefits impacting those developers and that is super important. As Eddy just outlined for us, there is a lot more to delivering software than simply writing code, and I think that a lot of our AI efforts across the industry up to now have been focused primarily on how do I write code more efficiently. But writing the code has rarely been the bottleneck for being able to deliver change into your production environment. So, as we start thinking about where else across that software delivery life cycle can we introduce AI, I think this is where we will start to see those trends change and start to see the software delivery performance starting to improve.

So, I think there are a lot of reasons for optimism and, don't get me wrong, just simply making developers' lives better and giving them faster opportunities to test out new things, try new approaches, learn faster, all of that is really good. We just aren't yet seeing the downstream impacts when it comes to software delivery performance, and I think that yet is doing a good bit of lifting in that sentence.

Gary Arora:
Culture remains a top factor of success, according to Dora. The data showed a staggering gap between high and low performing teams. Your high teams had 180 times more deployments per year, 127 times faster lead times, 8 times lower change failure rates, and over 2000 times faster recovery times. That's the difference between leaders and laggards. What are your thoughts on why this difference exists and what can organizations do to bridge this performance gap?

Nathen Harvey:
I think you hit it at the beginning there, culture. Culture is foundational to everything that happens here, and we continue to see this frankly across the decade or more of research that we have done with Dora, culture is always one of those number one indicators into how a team is going to perform. When we say culture, what do we mean? What's a good culture? Well, let me tell you, it's foosball tables. No, sorry, that's not what we mean by culture. What we mean by culture is that we have organizations where there's strong alignment across teams. We have organizations where risks and learning are encouraged, and when something goes wrong, we use that as an opportunity to learn and then improve the system. A failure is not reason for punishment, its instead of reason for growth and learning.

When we start embracing these ideas, it becomes much easier for us to attain those high levels of performance. The other thing that's really important to remember about those levels of performance, in order to become a top performer, you don't necessarily have to seek out 180 times more deployments this year, what you need to seek out is 180 small experiments that will help you improve. Identify the bottlenecks in your system, go and make changes and improvements to those and then look for the next bottleneck. Continue that over and over. Dora really is here to help you and your teams understand how you get better at getting better. This is a journey of continuous improvement.

Eddy Krumholz:
I will add that to bridge this performance gap, organizations should focus on fostering collaborative culture, promoting cross-functional teamwork, and empowering them to have decision making autonomy. Adopting a continuous improvement mindset is also crucial because it encourages learning and experimentation and establishes a feedback loop for ongoing enhancements. So, investing in automation to handle repetitive tasks and leveraging advanced DevOps tools that can streamline workflows and improve efficiency can also help. Finally, providing strong leadership with clear goals and visions along with active support and resources for DevOps initiatives can drive success and help low performing organizations achieve the high performance in leading DevOps practices.

Gary Arora:
So, continuous improvement, collaboration, and really psychological safety for teams to experiment and recover from failures. Thank you for so elegantly and clearly calling out this X Factor. The other finding that got my attention was that organizations moving to cloud without adopting the flexibility that cloud has to offer may be more harmful than remaining in the data center. We are seeing the same with AI. We are just throwing AI or Agentic AI, or any new buzzword because it's trending right now is either backfiring already or bound to backfire soon. So, why do you think organizations fall into this trap, and what's your advice for ensuring a true transformation beyond just changing an application's location?

Eddy Krumholz:
In my opinion, organizations fall into this trap of adopting new technologies like AI or Agentic AI, like you said, which I think is fascinating, but where that is going without fully leveraging their inherent flexibility and capabilities. I think that this can be attributed to several factors. One is trend-driven decisions, whether it's because organizations go through hype and pressure or because they are doing just a surface-level adoption. The surface level adoption is typically where they adopt this technology superficially focusing on immediate benefits of the appearance of modernization rather than a deep strategic integration.

The second one is due to lack of strategic planning. Whether you are having insufficient planning moving to the cloud or integrating AI without comprehensive strategy that can result in missed opportunities to optimize process and workflows, then organizations may not fully consider how these technologies can transform their operations. Also, in that same venue, this idea of legacy mindset that they carry over traditional practices and mindsets into new environments. It is an issue of skills and knowledge gap and the lack of expertise. Organizations may lack necessary skills to effectively implement and manage these technologies.

Lastly, there might be insufficient training to develop staff that can hinder the effective adoption and utilization of new technologies. To fix it, I think what you need is the obvious. You need to develop a clear strategy, you need to leverage and flexibility and capabilities at par, and you need to invest in skills and training. Lastly, you need to have an iterative implementation and feedback through pilot programs and continuous improvement process, as well as focus on the business outcomes. Aligning with business goals and measuring the impact of those goals.

Gary Arora:
Nathen, I want to get your take on this because I agree with Eddy that skills and knowledge gap is going to be a key differentiator and a key theme in the AI era.

Nathen Harvey:
Absolutely, I would love to give a little bit of insight there. Honestly, I love where Eddy ended it, talking about business priorities and aligning to business priorities, Dora has found lots of things over the years, including insights into cloud, but one of the things that's really come out in the last few years is this idea of user centricity and making sure that we are doing things in order to serve the users of our application. In order to do that, we have to know who those users are and what they are trying to accomplish. When you think back on the cloud, it's really interesting. I would imagine that there were zero users of your application or service that came to you and said what I really want is for you to be running on the cloud, I am tired of accessing this application through a data center.

Of course, none of your users ever ask for that. But they did ask for other things like more reliable systems, more secure systems, and that flexibility that you get from the cloud is a way that you can achieve the user requirements and really build a user centric team that's aligned to how do we provide better business value and how do we provide better value for our users. So, I think when you think about cloud or any other technology, you have really got to put the user first. So, I think that's something that's really interesting and important for teams to do and that also comes back to culture. How are we organizing our teams, how much information are we sharing, and what sort of collaboration do we have across the teams? So, I think really putting that user first is so important.

When you do that, it can guide not only your cloud transformation, but also your adoption of new tools like Generative AI and whatever comes next. Then the other thing that was really important there that Eddy touched on is creating this climate for learning and fast feedback because the truth is we can sit down and write a five-year plan and if we stick to that five year plan with no deviation from it for the next five years, we are going to end up in a bad place. We have to have feedback mechanisms and the ability to change across those five years, ten years, whatever that is, that's the most important skill and capability that we can build within our organizations.

Gary Arora:
Couldn't agree more. Putting the user first as a way to guide transformation and adoption. Love it. Finally, I cannot wait for your next edition this year, especially with the pace of innovation expected this year. So, Nathen, what can we expect to see in the 2025 edition of the Dora Report and what are you most excited about to see in that report?

Nathen Harvey:
First and foremost, I can tell you in the 2025 report, here is what you will see, I don't know because we haven't written any of it, but what I can tell you is that it will have a different name. We talked a little bit about the challenges of that DevOps label at the top of show. So, expect it to have a different name, expect is to be the 2025 Dora Report, that's what you should be looking forward to. I also think that Dora is getting better about taking our own advice. One of the things that Dora says is that in order to improve software delivery performance, you should ship smaller changes because then you can do it more frequently.

Well, you don't have to wait for the 2025 report to get new insights from Dora, in fact we have published more things since the report came out. There are smaller articles. We have got articles on how you build trust in AI and how Generative AI affects the value of development work. You can read both of those over at Dora.dev. So, one of the things that you will see is more frequent, although smaller, releases from the Dora team. We want to continue to be side by side with you as we go throughout the year and as you continue on your continuous improvement journey.

Then in terms of things that I am most excited to see about what happens over the next year, honestly, we have also created our sort of supporting Dora Community of Practice. This is a global community of practice for leaders, practitioners, and researchers all in this space trying to take the findings of Dora and put them into practice. You can join it at Dora.community and the thing that I am most excited about is what we will continue to learn from this community and together with this community who are taking these findings, using them as hypotheses for their next experiment, and driving that continuous improvement muscle within their organizations.

Gary Arora:
Alright, that's it for the episode. A big thank you to our guest, Nathen Harvey and Eddy Krumholz for sharing their insights. If you enjoyed today's discussion, be sure to subscribe and leave us a review. You can also check out our past episodes wherever you listen to your favorite podcast. Thank you for listening to the On-Cloud podcast. Until next time, I am Gary Arora.

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.