



## Architecting the Cloud, part of the On Cloud Podcast

**Mike Kavis, Managing Director, Deloitte Consulting LLP**

**Title:** Building teams that deliver at the speed of cloud

**Description:** How do you deliver results at the speed of cloud, and how do you ensure that the results you get are the ones you expected? You build better work teams that rely on collaboration and efficient flow to deliver better products, faster. In this episode, Mike Kavis and *Team Topologies* co-author Manuel Pais discuss the characteristics of high-performing teams and how organizations can leverage teamwork to achieve better outcomes. Manuel's advice for building better teams and increasing flow? Align teams and architecture; create smaller, more autonomous team structures; and reduce the cognitive load on team members. Manuel also cautions organizations to manage change effectively and to continuously search for bottlenecks and relentlessly work to remove them.

**Duration:** 00:28:51

**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](https://www.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, and welcome back to the Architecting the Cloud Podcast, where we get real about cloud technology. We discuss all the hot topics around cloud computing, but most importantly with the people in the field that do the work every day. I'm your host Mike Kavis, chief cloud architect over at Deloitte. Today I am joined by Manuel Pais. Manuel is a co-author of the book, *Team Topologies*, doing really well out in the field, one of Gene Kim's publishing company's books. They all seem to do pretty good, because they get good writers like yourself. And with that, I'm just going to let you introduce yourself because writing books is only a part-time job for you—you have a full-time job—and tell us a little bit about your background and how you got to—what motivated you to write that book.

**Manuel Pais:**

Sure. Well, first of all thanks for having me on the podcast. Yeah, writing the book was—*[laughter]* it was sort of a part-time—it turned out quite well since it resonated with a lot of people and companies. My background—both of our backgrounds, mine and Matthew Skelton's, are in computer science, so we've been developers. I've been also in testing, team lead, and I did a lot of work around DevOps. I used to be the lead editor for InfoQ. So, yeah, the book came out mostly from consulting work that we did together between 2015 and 2019. And, so, that's kind of my overall background, a little bit of everything.

**Mike Kavis:**

Yeah, what's interesting is a lot of the folks who had deep technology experience when DevOps and cloud started becoming a thing are all focusing more on transformation now, right? You look at John Willis and Andrew Clay Shafer, and all of us were deep in the weeds doing it, and I think just part of the DevOps thing—it's all about removing blockers, and the early blocker was CI/CD, and then it was testing, then it was shift security left. Now it's people, culture, and team structures. So, I find that interesting.

**Manuel Pais:**

Definitely.

**Mike Kavis:**

So, my first question is I work—part of the things I do is work on operating-model transformations, and we were talking before the recording. Like a year or two ago, it was a much harder sell. People were focusing more on the tech. And now it's incoming requests for help us organize, help us build cloud platform teams, help us transform IT is almost overwhelming. First, are you seeing the same? And second, why do you think that is?

**Manuel Pais:**

Yeah, I'm definitely seeing the same. I mean, we have a lot of requests for help just kind of starting these sort of transformations, more than we can handle, which is a good problem to have, like you said. And I think it relates to what you were saying before. That we've—I think the combination where we see there's new technology, there's cloud-native and platforms like Kubernetes. When you leverage that in the right way—so in the book we talk about platform-as-a-product internally and making sure that really works in the sense of accelerating the internal teams.

So, the technology is there, right, to make that work, and you have really high-profile examples and even some banks, and in many industries where I think the barriers for actually having this sort of transformation, and bringing together technology, as well as better practices with Agile, DevOps, et cetera, as well as now what *Team Topologies* has brought forward, which is more—deeper thinking around organizational evolution, how the kinds of teams and interactions, and how we evolve those to actually become high-performing organizations. So, all those things together, as well as the shift to product—from project to product. So, there's the book by Mik Kersten, also from IT Revolution.

So, I think all those different areas sort of are coming together at the same time, and obviously you have organizations that are starting more from the cloud perspective or the technology perspective. Others are starting from another point of view. But what I'm seeing is in most organizations, they have now this awareness that there are all these sort of movements, if you want to call it like that, where we can actually achieve a lot more with the same organization if we just start thinking in a much more evolutionary approach, where we take the teams and technology and practices all together.

**Mike Kavis:**

Yeah, I agree, and I think another component of that is the need for speed, right?

**Manuel Pais:**

Yeah, definitely.

**Mike Kavis:**

Almost everything we're doing fits into the world of digital now, and consumers of digital are used to rapid change. And, you have banks and other large companies who weren't used to that before now have to become digital companies, so I think the combination of what you said, and the combination that we need to move really, really fast—which is one of the concepts in your book. You talk about optimizing flow, right, or fast flow. So, I had your co-author Matthew on here right before you guys released the book last year, so I don't want to rehash that, but for people who aren't familiar with the book, give a quick synopsis of the purpose of the book. But what I want to get to is, now we're a year later, and you've been consulting with clients, implementing your practices from the book. What are your lessons learned, and what are the things that you preached in the book that are resonating? What are the things that you would tweak now?

**Manuel Pais:**

Yeah, so very briefly, the book helps bring into perspective several aspects of organizational evolution and sort of human factors in terms of trust boundaries in teams and across groups of teams. How do we promote more autonomy in teams so that they can go faster because of that need for speed that you were mentioning? And, so, the book really provides a set of patterns and practices around that, because, until then, a lot of organizations, even the ones that were more kind of switched on, were still mostly relying on looking at the Spotify model, looking at DevOps. But none of those addresses some of the issues around Conway's Law, the fact that we need to have a good alignment between teams and the architecture of the systems, the trust boundaries I

mentioned where it's not the same to if you try to have large teams, or large groups of hundreds of people, trying to develop this huge system. It's going to be very difficult because the trust just breaks down.

So, effectively, we brought all those ideas together. We also talk about things like cognitive load that really—I would say that's the number one thing that resonated with people from all angles, from individual contributors, to the CTO and CEOs, that we all understand that there's a limit to the human brain capacity and to any team's capacity. And, so, I think that's—in terms of bringing the vocabulary to talk about what are the current challenges in terms of how we are organized, and how we work together, and how can we move forward, is the key that the book brought about differently from what existed before.

And, so, we see a lot more organizations looking at *Team Topologies*. Obviously the larger organizations take a bit more time. But it's also interesting, because we see different starting points. So, I think that's something we didn't necessarily anticipate when we were writing the book, that you have—for example, Agile coaches are looking more at the kind of—at the team perspective, at, “How do we improve the performance of our team, how do we clarify interactions with other teams?” And then you might have other people in kind of leadership roles, or the CTO who's looking at, “How do we align with things like Conway's Law? How do we make sure that teams can actually—that we can actually produce and deliver our products and services in a much faster way with more independent teams?” So, it's been quite interesting.

**Mike Kavis:**

Yeah, one of the things you talk about in the book is a team API. So, describe what that is, and then my question will be how easy or hard is that to get companies to adopt that concept?

**Manuel Pais:**

Yeah, the team API has worked quite well. So, we use this analogy with the application programming interface for systems, for technology systems, for the teams, right? So, it's effectively the interface of the team, which makes—helps already reduce kind of the noise in terms of communications between teams. And obviously with the current remote context because of the pandemic, or because organizations are moving towards more hybrid models, it becomes even more important. How do we set up these interfaces for teams so that we know, and we don't have the overhead of finding out, how do I get in touch with this other team, or even who can help me with this problem that I have, or help me find a solution for a problem? And, so, the team API really helps with that. It's very easy to get started and you don't even need—it's kind of a team decision, right? We want to make it very clear for others that depend on us, perhaps on how do we work, what is our roadmap, how to get in touch with us, and, so, on. So, it's very straightforward. It's one of the things that has been very successful and that we see a wide adoption. Yeah, that's definitely one of the things.

**Mike Kavis:**

So, in that concept, all the different teams in the delivery chain would put in place the same team API concept, so regardless of what group I've got to talk to, it's a similar process. Is that the concept there?

**Manuel Pais:**

Yeah. I mean, it's a team decision, so they make it—they expose, let's say, their interface to other teams. It starts—it's a kind of outwards-facing approach where we're thinking, “What do other people need to know about us, or people outside our team?” When you start having most teams adopting a team API as well—so we actually have a template on [GitHub.com/Team\\_Topologies](https://github.com/Team-Topologies), but whatever kind of artifact you use, the important thing is that you have the information there that helps other teams communicate with us and understand what we do, understand how they possibly depend on us. That's kind of the first step, actually, to minimize cognitive load, at least on the communication side. We make it easier for others to get across to us and understand us, and then we look at other ways to minimize cognitive load as well.

**Mike Kavis:**

So, when designing the operating model—do you guys use the term operating model? Or I guess team topologies is probably the term you use.

**Manuel Pais:**

We talk about team topologies, but there have been people like Charles Bass, for example, who talk about this is the next, or this is a better fit for the digital operating model, which I think really means, at the end of the day, that having the tools—to me the book is still, at the end of the day, a set of tools and patterns and practices to reason about our operating model, to reason about our organizational structures and how do we evolve them, which is interesting because the reverse kind of question—what has been more challenging? And one of the things—we still see several organizations that they like *Team Topologies*, they want to adopt it, but they still assume that there is a right topology or ideal team topology, when there really isn't because the problems you're trying to solve and the challenges are—as we are very aware with the pandemic—they change very quickly, and, so, we need to change and be able to evolve the organization quickly as well.

So, there's no kind of definitive model, or sort of Spotify model, that we say, “This is the right approach for our organization.” It's a starting point, and then we need to think about how do we evolve using the interactions between teams, using sort of—we started to see this role of kind of, if you like, socio-technical architect or sort of cross-team enabler, which might be a specific role, or it might be a team, or it might be just sort of different people coming together to have the technical view, and the organizational view, and the team's view, to make those things come together and be aligned, so we can evolve, and go faster, and have faster flow.

**Mike Kavis:**

Yeah, let's talk about that role, because one of the projects I was involved in, one of the guiding principles we laid down was, “optimize for flow,” right? And that drove the concept for the client. “Well, we need to build the team or the structure where there are people who just focus on that.” It sounds like similar to what you just said. So, when companies, whether it's a full-time or a part-time—when they assign someone the task of keeping an eye on how flow is happening and continuing to optimize it, what are the skillsets for that type of role? What type of people do we look for that do that type of work?

**Manuel Pais:**

Yeah, that's a good question, and obviously it has to be someone with a mixed profile of technical knowledge. So, if you have a senior architect or enterprise architect who is also switched on to the people and to the team aspects, and who understands what is the impact of not having the right communication paths or not having team APIs. What is that impact on how we are able to develop our products and our systems faster with better results, then they might have all the technical expertise but without that socio-technical component of understanding the impact of the team setup and communication, then it's not going to help very much.

So, it might be this kind of senior architects that have this also strong focus on teams and the relationship between technology and teams. It might be—we've seen with one client where, actually, they have a very strong community of Agile coaches, where maybe they'll not have such deep expertise technically, but they are sort of good sensors, right? They have these sensors where we can see across multiple teams and we can see common challenges, whether that's interactions or dependencies that are causing us to slow down. And, so, they can at least surface those common problems and get the maybe, higher management, or other roles to actually, "Okay, let's act on that and see what do we need to do differently? Do we need better platform services, or do we need some enabling teams, perhaps, whatever it might be?"

And in some cases—although, to be honest, I haven't seen this in practice yet, but I would assume it could be also sort of a—not exactly a role, but if you see sort of a group of people where you see technical people coming together with maybe management, or human resources even, where we're making decisions together on—especially if you have larger programs of work. If you have that sort of approach, then maybe when we're starting that work, we want these different perspectives to come together and say, "Okay, how are we going to actually get the right teams in place, support the teams in terms of their communication needs, so that we can evolve and build this—maybe a larger system in a way that is flexible enough for us to adapt to whatever surprises come up when we're developing the system?" Right? And move away from the, "Well, we're going to define everything in the beginning, do a project schedule, and then just help everything goes fine." And instead that we're already from the beginning acknowledging things are going to change. There will be surprises, things we didn't know. How do we make sure we help the teams be able to communicate?

**Mike Kavis:**

So, two follow up questions. One is how do you measure flow? And then the second one after that is how do you design for change? Because a lot of times people will put in an operating model, and it's a big reorg and then they're very reluctant to do another one. So, how do we design so—maybe is the reorg not that big? Is it small, incremental changes and you keep going? Or—so first question, how do we measure flow? How do we measure that our design is actually optimizing or driving faster flow?

**Manuel Pais:**

Yeah. So, for measuring flow, obviously now we have the *Accelerate* book, and we have the four key metrics that have become—I really see that as a major step forward for the industry, compared to what I saw when I started about 20 years ago, where there were recurring discussions of what metrics, what should we be looking at. So, now we have this body of evidence where these four metrics—lead time, deployment frequency, change failure rate, and mean time to repair—are really correlated with high performing. But there's also another metric that I tend to recommend, which is quite simple, which is flow efficiency, which is you take the lead time, how long it takes from the moment we have an idea or we have sort of approval to make some change, until it's actually delivered to customers. And from that overall lead time how much was spent in some sort of wait state where we're waiting for another team to do part of the work, or we're waiting for some approval?

And, so, flow efficiency is basically the percentage of the overall elapsed time that we have actually been actively working on the change. And surprisingly or not, for most organizations that number is very low. If even 10 percent or 15 percent of the elapsed time was actual active work time, it's actually a decent number. And, so, that just means there's a huge opportunity for improving there. So, that can be quite a good approach, where you're looking at flow efficiency as a kind of high-level metric, but then when you dig into, "Okay, where are the wait times," and you start doing things like value stream mapping, you start seeing what are dependencies between different teams in this value stream, and then you start to improve from there.

**Mike Kavis:**

Yeah, flow efficiency is a good one. I've seen a value stream where I think 98 percent of the process was non-value-add, because it was mostly manual, most review gates, mostly checklists, and it was just time between—it wasn't that people weren't doing the right thing. It was just so much time between the steps that no work was getting done.

**Manuel Pais:**

Yeah, it's never—or not often—it's not like bad intentions, but it's again a mix of reasons. There are historical reasons why we started doing these sort of controls, and there are other reasons which sometimes are related to the way we're organized, right? So, if we have, for example, compliance as kind of a separate line in the error key compared to application development, then typically there will be more of those kind of perhaps gates or approvals because the actual organization structure seems to indicate that we need this separation, right? And, therefore, we go back to why we're thinking more about team topologies and interactions and thinking about moving from blocking dependencies to non-blocking dependencies.

So, I don't want to pick on compliance, because it's also quite a complicated domain, and there's a lot involved in there, but that's actually one of the challenges we see still for many organizations, especially the larger organizations, is, how do we move away from compliance as a sort of blocking dependency to non-blocking? And, so, I think we're going to see a lot of evolution in that space. People like John Willis also are talking about that quite frequently. How do we move to more kind of self-service compliance, if you like? How can we have a sort of platform self-service approach to those requirements which are obviously very important, but can we implement that in different ways that don't block the flow?

**Mike Kavis:**

Yep. And, so, the second part of that question—we were talking about this role or this group that looks at fast flow, and we talked about the need to constantly change. How do we implement something like that when, a lot of times, the first change is a big organizational reorg and then there's no appetite left to do another one anytime soon? So, how do we approach more of an iterative approach to implementing change?

**Manuel Pais:**

Yeah, that's also my experience. What I've seen is those big reorganizations, when we're trying to—again with good intentions we're trying to align to some new approaches like Agile, DevOps, or SRE, or we're adopting the Spotify model. So, the intent is good, but, because we're starting with the trend or the approach, instead of starting with more kind of fundamental objectives, or principles for the organization, which, like we were saying in the beginning, fast flow, or the need for speed, has kind of become fundamental for almost every organization. And, so, instead of focusing on big reorganizations where we're changing roles, perhaps, and we're changing teams and groupings of teams in kind of a—again with the expectation that there is some sort of one-step or change where we'll get to a much better place after this change, that's not very realistic.

And like you said, many people start losing the motivation to even go through these transformations, because they have not seen the value that was expected from the previous transformation, and now they're already going through a new one. And it just kind of, burns a little bit the motivation of people who actually want to improve but they don't see how this big reorg is helping, because what they see is the impact on, we're changing again the teams, we're changing again the roles. So, my opinion is, again, to have much more of a continuous evolution, have changes that are smaller in terms of organizational structures as well, have changes driven by the interaction between teams where we identify, “Well, there's a blocking dependency here. There's a problem with how we depend on this other team to create infrastructure for us or to control our cloud costs for us.” And, so, how can we move away from that and kind of give teams that sort of autonomy that we know is going to help achieve faster flow?

**Mike Kavis:**

Yeah, absolutely. That's the whole team API concept, right? How do we better interface between groups, right? So, fascinating conversation. I wish we could go another hour on this because it's one of my favorite topics. But the one thing I wanted you to point out to people is what's unique about you guys' book is there's a lot of artifacts that go along with it. You have stencils. You also have little mini-chapters, or little mini e-books. There's a lot of good stuff. So, tell us where we can find all this, and I highly recommend listeners to dig into this stuff. It's some great stuff. I actually leaned on some of this stuff in my book slightly, because I was talking a little more broadly than this, but I actually leaned on some of those concepts so I appreciate that.

**Manuel Pais:**

Yeah. So, there are a lot of materials, both that we've created to kind of help organizations just apply these ideas in practice, as well as community materials and applying the ideas of *Team Topologies* to other domains like data science, for example. So, the easiest sort of entry point is our website, TeamTopologies.com, and from there you'll find key concepts, you'll find community materials. You'll find links to our GitHub repositories, so that's GitHub.com/TeamTopologies, so we have templates for team APIs, for tracking team dependencies and a lot of other things. We actually, this week, or the week of the recording, we just published some new infographics on how to get started, which have been very well-received.

We're also—we've also started a Team Topologies Academy, so we've done a lot of live training, and now we've set up online, self-paced training, because we've realized, for some organizations they want to scale across a larger number of people quickly. And also there are just some time zones that it's very difficult to help with training. And, so, we're quite excited about that and to see where that goes, so we're going to add more, learning paths and new modules. And we're also trying to scale our kind of—the consulting approach with partners that can help with support to more clients. And we're also quite excited—we're starting—it's sort of still early stages, but we are trying to have better ways to assess cognitive load and evaluate cognitive load over time.

Like I said, in the beginning that's one of the key ideas that almost everyone resonates with and says, “Yes, this helps us have better understanding of problems in teams and talk about those problems in a more concrete way.” And, so, we're looking with someone who has a strong background in psychology, as well as understanding the software delivery side, and, so, we think we'll be able to come up with something that is going to be quite useful.

**Mike Kavis:**

Well, I'm looking forward to that. And you do a lot of podcasts and speaking. By any chance is there one place to go, or is the best bet just to Google you and find all your content?

**Manuel Pais:**

In terms of talks, on YouTube I have a playlist, so, *Team Topologies with Manuel Pais*, and Matthew also has a similar one. But yeah, in terms of podcasts, it's a little bit spread around. [Laughter] Just, reaching out to different communities, and that—like I said, it's been a very pleasant surprise that people with different roles have an interest in the book and the ideas, because different concepts can be applied depending on where you are in the organization.

**Mike Kavis:**

Well, great. Great chatting with you today. Congratulations on the success of the book.

**Manuel Pais:**

Thank you.

**Mike Kavis:**

I think it's becoming the bible of this topic, so congratulations on that. So, that's it for today's episode of Architecting the Cloud. To learn more about Deloitte or read today's show notes, head over to [www.DeloitteCloudPodcast.com](http://www.DeloitteCloudPodcast.com). There you will find more podcasts by myself and my colleague David Linthicum just by searching for Deloitte On Cloud Podcast on iTunes or wherever you get your podcasts. I'm your host Mike Kavis. You can always reach me, [MKavis@Deloitte.com](mailto:MKavis@Deloitte.com), or find me on Twitter: @MadGreek65. Thanks for listening and 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](http://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](http://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](http://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](http://www.deloitte.com/about) to learn more about our global network of member firms. Copyright © 2021 Deloitte Development LLC. All rights reserved.