CLOUD REALITIES

CR061

Product engineering in the worlds biggest startup with Steve Westgarth, Haleon

Capgemini

CR061

Product engineering in the worlds biggest startup with Steve Westgarth, Haleon

Disclaimer: Please be aware that this transcript from the Cloud Realities podcast has been automatically generated, so errors may occur.

[00:00:00] You've, you've all been super organized and be like, you know, this is what you kind of need to do. And I'm like, yeah, yeah. I've podcasted before. It's all good. It'll be fine. And I've turned up and like, like disaster zone, like a literal disaster, but it's all good. I'm here. And that's all.

Welcome to Cloud Realities, a conversation show exploring the practical and exciting alternate realities that can be unleashed through cloud driven transformation. I'm David Chapman. I'm Sjoukje Zaal, and I'm Rob Kernahan.

In this week, we're going to explore product engineering. Now product engineering as a subject and moving to product based organizations is something organizations often think is really only the space of born in the cloud companies.

He's a digital companies, but actually [00:01:00] there are a lot of complex organizations that are implementing these structures. We're going to examine that today and some of the challenges and opportunities that it can create. But before we get to that,

I wandered into the office the other day, just as Rob was finishing his new, um, his new, uh, implemented daily standup, I think he was about to go into the third hour of his daily standup and, uh, I, I wandered over to him and I'm like, well, what's going on? He's like, well, Dave, I'm really enthused by Agile Frameworks at the moment.

And I thought I'm going to implement a daily stand up wherein I can really dig into what's going on. So give me a minute and I'll tell you about what I'm confused about this week. So I said, okay, I'll just go and sit over there. Anyway, a couple of hours later. I've got a chance to say, Rob, what are you confused about this week?

So, David, love it, but you know I'm not the best at paying attention for long periods of time, so I don't think that would work with my, uh, with my [00:02:00] attention span. I'm not suggesting you were consistent within that four hours, Rob. Probably kept falling asleep halfway through it as well and waking up again.

Yeah, most likely, let's be honest. Let's face it. But this week, David, I am mainly confused About good intentions gone bad and then how sometimes it auto corrects itself and sometimes it doesn't. I think I think we've all fallen over on auto correct at times. I know I have, but the one I'm going to call out is the cookie law.

Born in, born in the right ethos, but badly implemented. So, uh, data, privacy cookies are tracking us all over the internet. What do they do? Well, privacy law rises and says you have to have the right to control your cookies. So what happens? This really clunky implementation comes in. All the websites have an accept cookie button.

You can manage your choices, but there's 2000 items to go through, which basically means that as we've talked about it before, people like convenience and everybody just have clicks, accept cookies, [00:03:00] which means that the law with great intention ended up, um, not really achieving what they want. And, you know, you're all out there, you all know, you've just got, I can't be bothered accept cookies.

Right. Yeah. And then what happened was the tech industry has looked at that problem space and has caught up. Yeah. And if you look at what the chrome engine's doing, you look what apple and android phones are doing, they're starting to implement new privacy systems where you set your preferences at the core, which is basically don't let anybody track me.

And then the platform itself takes control of that. So the accept cookie button in the next six months will be redundant because the phone and the browser has completely deleted it. But I'm still confused about the it took a long time. For the tech companies to catch up with

something that maybe the legislators should have thought about and said it's within the platform's right to control this.

Not each individual website because it cost a huge amount of money to implement and it ended up being a farce anyway. And actually, it was easier just to amend the platform. So confused about why that thinking wasn't done up front because [00:04:00] I remember when the law came in a lot of people said why don't you just control this straight from a browser choice and it's just default set and that's the end of it so isn't this just the the classic case of you know well meaning legislators exactly and and governmental bodies you know even with the best one in the world maybe not quite understanding the thing they're trying to legislate against so the they're probably understanding it from a point of view of Your rights, whether it be your consumer rights, your human rights, whatever it might be, what concerns me about this is actually, it's probably less about the example you cite though, you know, I'm ultra guilty of accepting all cookies that ever pop up in my, in my browser.

It is then what I'm signing up to. In years to come given, given the rise of the robots and Yeah. And your data's already out there and consumed and the That's right. Have been made. And the big C companies already know who you are and have done it. So I think we've passed the Rubicon as well. So we're, we're trying to retrospectively install laws.

But the data's, yeah. [00:05:00] The, the, the horse has bolted, you know, it, it, whatever analogy you like. The time for something like this was at the very beginning of when the big corpse were forming. Do you think it's completely unfixable now? We're so far past the Rubicon that actually the game's over. I think, uh, you need to change your name, uh, move house and, uh, think about a new persona online.

If you want to just go, yeah, it's, it's, it's all been collected and processed and gone. I think, I think you've resolved that problem, Rob, with a nice, easy fix for everyone. So thanks for your time. Consumer advice, again. Exactly. Consumer advice. That's what they come here for. So, Rob, thanks for that. Now, I'm delighted to introduce our guest for today.

He's going to help us work our way through the thorny subjects of product engineering. So, I'm delighted to say that joining us is Steve Westgarth, Head of Product Engineering at Haleon. Steve, very nice to see you. Thanks for making the time to join us. Do you want to just say hello and introduce yourself?[00:06:00]

So first of all, really good to be here. And my name is Steve Westgarth. I'm the global head of engineering and architecture at Haleon. And, um, I sit within a product organization within the organization where we look at how we can build products that drive better everyday consumer healthcare with humanity across the organization.

Let's start, Steve, by talking about Haleon themselves. Very interesting company. Lots of things have happened over the last couple of years. Why don't you just give, why don't you set the scene for us and just give us some background? Sure. So Haleon is a consumer healthcare company. A couple of years ago, consumer healthcare was spun out of GSK.

We were GSK consumer healthcare and GSK had committed to, to float a, their consumer healthcare arm. As a brand new limited company. So in July, 2022, we separated from GSK somewhere in the region of 20, 000 staff operating out of 130 odd countries, um, selling products that you'll know every day. So things like Sensodyne, Voltarol, [00:07:00] um, Centrum, all of those consumer healthcare brands that you're used to buying from your pharmacist are owned by Haleon.

But the brand as a whole is, is brand new. Two years ago, we didn't exist. So an 11 billion

company. Born out of, you're a huge pharma company that you will have heard of, but, but as a company ourselves, we're still establishing ourselves within the market. Yeah, amazingly exciting opportunity. And what does that mean for it?

So given you described it interestingly there, has that allowed you to go in with sort of modern it right from the outset? Or are you still dealing with some legacy that's coming across from your parent company? It's a really great question. So I think there are so many opportunities that we've had. I mean, so first of all, the separation was a huge challenge.

So imagine a world where you have a multibillion dollar company, you've got all of your system set up, all of your HR, your payroll, your ERP, all of those core systems. Had to be separated and set up again from scratch in the new company. So, um, that required us to, um, have new vendor [00:08:00] negotiations required us to, um, go in, um, and literally create and, you know, everything from, from, from day one as, as a new installation, new teams to support all those applications, new vendor relationships to support all those applications.

And I mean, the undertaking really can't be understated in terms of how, how big a job that was for our, for our tech teams, our tech organizations. As a company though, we, we approached that in a way which actually got us to a point where on day one, um, of, of launch of Haleon, you know, compared to many other comparators you would look at that have separated in, in, in recent years, we, we had total separation of, of everything.

Every system was, was our own and we were running our own, our own payroll, but a lot of that was replicated from what we had in, um, in GSK, you know, it was based upon a model that we knew worked and understood. We had made some decisions along the way. Things like you're needing to be cloud first and the relationship we have with with Microsoft and you're wanting to move our [00:09:00] workloads from physical data centers into the cloud was really important to us.

So we did make some big decisions to allow us to do that. But even in that world, you were you were moving along where you didn't necessarily have the opportunity to Create all of our applications in a cloud native way. We might be using the cloud as effectively as a data center, but a lot of applications have been put into the cloud on VM technology or using, um, you're not, not necessarily cloud native technology.

And that I think is a slightly longer term opportunity for us as we kind of rationalize and see how we can better leverage the cloud to support, to support what we're doing as a company. I guess you find yourself there in an organizational state or application landscape state that's actually pretty commonplace to be right where you've got stuff running on the cloud, but it might not be completely cloudified at that point, but you've actually got like an opportunity point forward to create a modernization agenda.

You do exactly that. And I think that that's also why engineering is so important at Haleon. So historically, GSK had [00:10:00] outsourced lots of its engineering work to third party vendors, and that had worked very effectively for the organization. You're aware where some of our, um, some of our core technology, how it was being built and how it was being supported and your, uh, some of the decisions that have been made along the way, um, to, to launch from those systems and to look at how we can, um, uh, Build out in a state where you haven't got to think end to end every time somebody comes along with a new proposition.

So I joined the organization as global head of engineering in January, 2022. I was about maybe six months before separation and my remit there has been to, to really establish application engineering as a discipline within the organization and to help the organization to understand how we can, how we can build out a modern product aligned.

Um, engineering practice where we can move to a world where, um, we can build APIs once, build them at scale, and then reuse them many times across different areas of the estate. So we've spent a lot of time really maturing our [00:11:00] API strategy and looking at how we can unlock data within the organization.

But unlock it in a way that can be reused by many, many different applications where we're in a world where, um, and I guess this is applicable to a lot of companies, our core systems, you'll hold lots of our data and it's kind of locked away. It's very difficult to integrate with. There's lots of vendor lock in your systems and a lot of what we've been trying to do is.

To look at our estate and see how we can unlock that and allow us to, to be able to, um, you know, to allow modern applications, new applications that we've built to be able to get access to the customer data. That's really going to drive our growth as we kind of move forward. I think it's a really interesting scenario because you have.

One of the big things that we talk about on the, on the show a lot is that when you're pivoting an organization from legacy structures, which very common things that you're talking about, like data locked away, big vertical applications, outsourcing contracts and reskilling. So when you're pivoting [00:12:00] the organization and just doing transformation without having said detached and created a new startup along the way, you're dealing with this huge cultural change.

So it seems to me, I wonder if, I wonder how you'd reflect on, on that versus the journey that you're on, where you, you've inherited a lot of the technical aspect of the legacy problems, but presumably you've got a chance to sort of create new culture right from the outset. We do. Um, I mean, culture is so important as well.

I mean, we're in a world where I would say across the digital and tech organization, 70% of staff within digital and tech on net new to Haleon 30 percent obviously coming from, from GSK. So it's, it's quite interesting because GSK as a company is a pharma company, and that comes with different regulatory requirements to, to what a consumer healthcare company might have.

And that opens up a new opportunity for us as a CPG. to, uh, to make different decisions about some of our governance processes, some of the, [00:13:00] um, the things that we're, we're operating, how we're, how we're facilitating your, uh, our operations, our software and technology build. So because we're in a world where we can leverage that opportunity, we have a lot of new people who have a lot of new ideas and want to mature the estate.

And what we're now finding is. Yeah, as an organization where unpicking your how things have always been done around here and looking at how we can improve some of those processes, how we can become more efficient as an organization, how we can, um, your really drive business value as an organization by making things less complex.

And I think it's something that comes up quite regularly. that it is sometimes quite difficult to get stuff done around here in tech. And, and that's something that I think the organization is taking incredibly seriously. You know, we have some, some significant programs and I think that's so important because as a CPG, our margin is, is a lot, is very different to the margin of a pharma company.

So, you know, while we, we have very high volume of sales, um, you obviously have your huge multimillion dollar [00:14:00] company. Um, those sales are much more marginal. In terms of where that can be translated to the profit line. So, and obviously you have different expectations, um, you're on the stock market in terms of your, what your profitability will

look like and what the various ratios are going to be that sits with that.

So I think there's, there's definitely been a lot of thought given to how we can drive efficiency with the organization and how we can look to make better bang for our buck, if you like. With the investments that we're making. So, so we're really intelligently looking at how we can consolidate technology, how we can, um, your use things in multiple areas of the organization, how we can use fewer technologies in our tech stack.

So we've got your, your less things to support. So, so there's this rationalization exercise, you know, it's actually, I think, a really healthy thing for the organization to do. And then from a software engineering point of view, it also, it makes sense because, you know, we're in a world where I've inherited.

hundreds of repos and there are, you know, [00:15:00] literally tens of different languages have been used here to build some of these applications. And from a maintenance point of view, if I've got to support 30 or 40 different coding languages, it's very difficult to do that. You know, how can you possibly have the skills to cover that entire breadth within your organization?

So, so we need to be more strategic about your what, what tools, technologies, languages we're using to build some of these applications. We While at the same time, not stifling innovation and still giving our engineers the freedom of expression to leverage the latest, greatest tech and also being aware that technology is constantly moving and we need to keep pace with where with where technology is at.

And that's quite a quite a difficult line to tread. And I think you set up, if I understand it correctly, I think you set up like product domains that align to Big sections of the business, but then you're managing product engineering as a tribe across the organization. Is that broadly in the right sort of place?

Yeah, that's, that's definitely the right ballpark. I mean, so I, um, I talk a lot about domain [00:16:00] driven design and there are obviously different ways to skin this cat and different organizations will do it in different ways. But if you think back to Eric Evans book, which looks at how we can talk in the same language as our business, we have this concept of core domains.

So core domains are the reason that we exist. They're how we're driving revenue in the organization. Um, core domains are the things that we're going to innovate in. You're the things that are really going to differentiate Haleon. So, so those for me are candidates to, um, you'll make sure we've got really good in house knowledge.

Um, that is your, uh, actively working on on some of those, those product line streams aligned to our business strategy. You know, the business is very interested in how we talk about some of those technologies and what we're doing and the outcomes that we're delivering. I then talk about, um, generic sub domain.

So a generic sub domain is something which is a solved problem. So, um, think about Content management, for example, [00:17:00] now Haleon should never, I would argue, be in a position where we're going to build a content management system. That would be a ridiculous thing to do. There are so many content management systems out there.

You would buy off the shelf, really important that it integrates well with our ecosystem. So we're really interested in Mac architecture. We're talking about something which is composable. API first, API driven, probably with some sort of eventing solution. So we can start, you're pushing content to services and doing all of those sorts of things, but still

something which we're going to buy off the shelf.

It's a solve problem. You're a PaaS service that we're going to integrate. And then you've got this idea of, of, um, supporting subdomains. So this is your, where we know there's some integration that's needed. It's probably a solve problem. Lots of people have done it before, but it's not our, not our USP.

It's not where we're, not where we're innovating. So, um, a good example of that might be. Payments, for example. So everybody takes payments. There's a lot of different payment providers out there. You need to integrate back into your state, back into your ERP and into all of those things. But again, it's something that potentially is a really good space for our partner ecosystem to operate in.

So we [00:18:00] start to say, well, because we're in an API ecosystem, that that's your kind of where we're building. We want to be creating, maybe a way to describe this would be almost kind of your internal software as a service products. So yeah. Cool. Um, and those products are effectively capabilities. So, so think about capabilities as things like payments or analytics or order management or transaction management, those bounded contexts, they might be leveraging past services, generic subdomains that we've bought off the shelf.

They might be integrated into the estate. But the concept here is, is that when an engineer or a product line team comes along, they should be able to pick and choose these composable services they want to use, and then reuse those within their applications. Think about if you take a different context other than Haleon, think about something like.

Payments. If you were going to integrate PayPal, so I would go to developer. paypal. com and very quickly I could integrate PayPal into my site because they've given me some sample code. There's some really great documentation and you simply, you'll include that into your react app and suddenly half an hour later there, I've got payments can have been taken.

[00:19:00] I see that as a very similar for our capabilities internally. So we have this model where you can pick and choose the capabilities you want. And we've standardized around backstage dry. Oh, to give really good internal visibility and discover ability of our of our APIs and services with some really good documentation that engineers can use.

And the idea is people can pick and choose all of these applications. We've written some internal SDK is that are able to be used and you can leverage that as part of your application. And although you're it. We're only two years into this journey, but I think you're so far that approaches is really starting to pay dividends because that's, that's where we're really starting to drive the reuse.

So if you take the order management capability, for example, we've done one recipe and you know, we don't need to integrate, you know, we don't need to integrate again for the other use cases that we've got. Cool. I'm interested in I think two levels of how that whole ecosystem is performing. I think the first from a point of view of the consuming business.

So how is it feeling to the, to the consuming business? Do they feel like they are now feeling intimate with that [00:20:00] process and feeling like they've got like a new innovation? Speed that they've been looking for. And then just to get it all on the table. The second element is for one of the developers within that ecosystem.

How do you feel like like it's functioning? And are you functioning yet that the sort of tick speed that you want to be running at? Definitely. So I think from the from our businesses point of view, It definitely has had an impact on how quickly we can achieve some of the things that we've been trying to achieve.

And although this, this, you know, right now, um, isn't in, in every area of our business, you know, it's still a, it's still a journey. We're kind of going on to, I think where, where, where we've seen your success, I think, you know, it's definitely seen, Some, some radical improvement in, in, in speed to market. And, you know, you've come along and you've been able to do something within your couple of months that's actually achieve what we want to achieve.

So, you know, it's, it really has had a massive impact, um, where we've been able to, to, to leverage these sorts of techniques with, with our business, which I think is really having a tangible difference. And when you, you've done this massive transformation, you've. You've greatly [00:21:00] improved agility, very progressive engineering thought process linking with the business, um, creating very high performing teams.

Do you ever go back to your GSK colleagues where you came from and you split from? And I know there are a more regulated area, but do you compare notes on the successes you had and share strategy and see if there's something that can benefit both. Is that a conversation you've sustained because you've done an awful lot to transform the way it delivers to the business very successfully.

So I was just wondering if there's a still a compare notes process that goes on. So we are now a separate company. So I joined Haley on six months before separation, so I haven't really got deep rooted contacts back within GSK. So although there has definitely been conversations and obviously your GSK is still you're a good friend of Haley on and will remain so.

Um, I wouldn't say that we're, we're actively comparing notes on approach. Um, what, what I would say is though, that, [00:22:00] um, as a, as, as a tech leader, I, um, get involved with, with a lot of peers and colleagues and other organizations and do evangelize a lot about this approach that we're going to take, and even the fact that I'm doing this podcast with you today, I talk a lot about some of the benefits that this has driven and the community it kind of brings.

And I do talk a lot about, you know, where, Many of us are in non competing industries and, you know, a lot of the problems we're trying to solve here are common across lots of different types of organizations. And I think there's, there's two things that I think I've, I've learned is, is I've kind of, you went through my career and kind of your experiment with some of these things.

The first is the importance of having long lived squads, regardless of how you're funding something, but people that know your organization. You're working with you. You're for the long term is critical. And I don't care whether those people are internal people or whether they come from a partner organization.

However, you're choosing to resource and to scale up your engineering team. That doesn't really matter. But the [00:23:00] importance is, is continuity because you're actually by building that you built so much knowledge inside your teams and actually where, where you lose velocity and where you lose time is, is when you're between programs and between projects, when something spins down and something else spins up, you know, that, that knowledge, which is lost, it's really intangible.

And I don't think people really appreciate how significant that is to an engineering organization. So long live teams. Is the first thing that I would say, um, is super important and actually within that, you know, what I've also found very effective is, you know, you're taking work to the team. So even if I've got multiple initiatives that might be funded in all sorts of different ways, doesn't matter whether those those initiatives are kind of planning.

You know, milestones over a period of time or kind of whatever that might be. You know, it doesn't matter the methodology you're adopting for the project, but driving the work into the team that has the knowledge, you know, rather than trying to, to orientate a new team around a problem is, is incredibly important.

I think that's, um, you know, that that's something that organizations must get. [00:24:00] Products, which is they're aligned to the business. They can have that intimacy. They need to not only survive, but thrive, but it allows that longevity of relationship to be established. As opposed to say a more mutualized service where the same requests can ping about the organization through multiple layers, get lost and get confused.

So it is, uh, I mean, I'm a massive advocate of the product based organization, but exactly to your point, it allows the type of ethos that you're talking about to be realized. More easily, but it requires the funding lines to be affected. It requires a lot of change within an org to make sure that the product can thrive.

So, so I think there are a couple of different archetypes that we have. So, so, um, archetype one is the true product organization. So you're against my bounded context of domains, your way. You have a team that is iterating against a product backlog, and that's a long lived thing. thing that we're trying to live with for the team, this website, this e commerce destination, whatever it might be that we're iterating and developing on that's aligned [00:25:00] to a value stream in the business.

So, so that's, that's one archetype. The other archetype of engineering we've got is what I would term continuous delivery. And this is where you've got ad hoc demand of things in our business where somebody has come along and And said, Hey, I need to make a change to this. Hey, I've got this request to automate that, you know, Hey, I need, I just need to do this because it's going to enable something for what I'm kind of doing.

Well, when we have that sort of demand, we have this, this concept of continuous delivery teams. It's still working a product line way. They still have a backlog. that's built out for them. But, but that demand is very much skill set based. So I have some, you know, some JavaScript engineers, I've got some platform engineers, I've got some, you know, some, some skill sets to the, you know, maybe API engines, whatever it might be.

And. And you're, you're driving that demand to, to a team that has the right skills to do it. And I think that that's another architect that works quite well. And in both of those scenarios, actually, I mean the funding, yes, it's important. Um, but, but you know, many [00:26:00] people, you know, companies will work with, with programs and initiatives.

Well, I work with programs and initiatives in terms of where they work, right. It's very much, you know, an agency model if you like, you know, it's no different to me setting up a business that was a software development house and selling my services that way to, to, to multiple different businesses. And by thinking differently like that and thinking around how you can, without having to change all of your internal processes and change how you're funding, um, all of your projects and programs and initiatives, you can, you can actually kind of meet your finance teams halfway and realize that to be agile, you don't, you don't necessarily need to have, you know, rewritten the way we do everything around here.

You can adapt and make that work, which in its nature is being agile, right? So there's a lot said, you used the big A word. In that answer there. And I felt like maybe is to bring today's conversation to a little bit of a conclusion. Barely feels like we scratched the surface. I just want to talk to you about agile and it's a well worn path.

So I'm all about driving autonomy to the teams. I mean, this is a big agile, [00:27:00] agile principally. How do you make your autonomous teams that are able to make independent decisions now in command and control management structures where you've got people who need updates and all those things. It's actually quite difficult to let go of control and give the team autonomy.

But one of the things we do, we, we, every, every six weeks, um, each team comes in with their, their leadership. So the product owner, the engineering manager, the principal engineer, and we have a health check and they have, you know, they're required to prepare some content, tell us, you know, what's going well in the team, what our achievements are, what team changes have got, what our challenges are.

What management intervention and what management support that they need. And, and you, you actually find that because you've given that accountability to the engineers, certainly in my experience, they embrace that and they go, this is brilliant because actually we've got total control of our own world.

We can change what we want in our world. And that's all good. The things we can't change. The things you outside of our control, I've now got a mechanism by which people are genuinely interested in helping my team to get better. So, so to, to, to bring that kind of around, I mean, there are [00:28:00] so many topics in the space that I can talk about, but I think, um, organizations need to stop talking about frameworks.

I think that's, that's, that's not where we're going. I think we need to be talking truly about what drives true business agility. And the thing which does that is your product line teams, um, your, your Teams that are the same individual. So you aren't spinning up and spinning down teams. You're according to work, um, and moving to a world where teams are able to, to work with as much autonomy as you possibly can, and as few dependencies as you possibly can, and linking that back to the domain driven design stuff, those bounded contexts, that's a really great way of trying to, to move to a world where you haven't got as many dependencies between teams that enables them to, to really kind of push forward.

Sjoukje, what you've been looking at this week. So each week I do some research on related ideas and transformation and tech. And [00:29:00] this week I thought we should take a look at the five key characteristics of high performing teams. So high performing teams consistently meet their goals, work cohesively as a unit, and are engaged in their work.

So what makes a high performing team so successful? First, trust is essential for teams to reach their full potential. Clear communication. Clearly define roles. Prevent conflicts and boost productivity. Engage leaders, provide direction without micromanaging. And lastly, collective goals ensure that individual contributions align with the team's overall success.

So a question, Are these the most important elements of high performing teams? So that's a very interesting list. I think there's two very key things for me that. We need to think about first is purpose. And maybe that goes to the goal point. People with a purpose are better at creating something, especially when it's a purpose they're passionate about, and that's not increased shareholder value.[00:30:00]

That's create a new experience or something special or help. The citizen out of the consumer or whatever. I think the second point is they all come together to make happiness and happy people naturally more productive. So I think purpose and happiness and leadership focusing on creating purpose that isn't some moronic business concept.

And also the happiness that goes with it. We've made a podcast out of Mariah's business

concept. But that, that sort of just focus on the numbers bit doesn't motivate the average person to go and do something great. So you got that bit and then you've got the general happiness of the environment. And we talk about psychological safety in there and trust is a big part about that.

But without those underlying facets, all the things that you talk about, I think maybe don't work as well. So I think there's some. foundations for success that I would maybe layer in around those to say, if you don't have those, you don't get the other things that you talk about. I was going to, I was going to raise psychological safety.

I think it's like an [00:31:00] imperative in those sorts of environments, because one of the things that high performing teams do bet way better than low performing teams is the level of challenge that's involved. And you're challenging each other the whole time to improve what you're doing, whether that's a digital product or whether that's, you know, kind of any.

Type of work, frankly, just doesn't really matter, but you, you can't have edgy challenge without, without a level of psychological safety in the team. It feels like to me, the other thing I observe, which is a bit around the boundaries of the, of the list that you had, which is. As you read it out, people might have been listening and, you know, I had half an ear on all.

It was all quite obvious. It's like, well, of course you should have a clear reason to do this. And of course you should measure things. And of course you do this. But it's funny how many teams don't do it. And you got to, that's the big question for me, which is like, It's it's worth taking a step back and looking at even how kind of your team functions at the moment or whatever it is and [00:32:00] just go actually haven't got the basics of clean lines in place here before you even get to the more complex stuff, you know, I think Steve when you were closing.

Your point is don't focus on frameworks, but focus on driving autonomy and empowerment so that people can do the right thing well. And I think that that motivates people a lot when they're given the freedom to be able to build what they know they need to build and do. Everybody comes to work to do the right thing.

Yeah. So give them the environment that allows them to do what they need to do. And on that psychological safety point. So just going back to the thing I was told about with health checks. There's a really good example of this a few weeks ago where we went into a health check and we have a team scorecard where I can do various metrics and kind of things that kind of goes out once a quarter.

And in this particular thing, there's a team that looks after enterprise search for us, and they're measured really on. It's the number of users of enterprise search. One of the things we're trying to drive more users of search. So I made an independent decision, you're aware, you know, we'd had a [00:33:00] stagnant three quarters of, you're not increased usage and actually in some cases decline in these of enterprise search.

And I downgraded it from, from a green to an umber rag writing. And we went into this, this team health check. And the principal engineer came in and he was like, yeah, we just want to have a conversation about, you know, the, the. The scorecard, you know, the team was really disappointed that, uh, that we were downgraded from a green to an amber.

We didn't understand why it happened. And I jumped off the back of that. I was like, you know what? Yeah, absolutely. I'm so sorry. I should have went into the team and had a conversation with you before we made that decision to explain to you why we were going from green to amber and to, you know, into, and to help you kind of go through that.

But it made me realize that actually, you know, as, as a leader, you're leading agile teams. You know, you, you make your, all these autonomous decisions and kind of things, But, but that can have a massive impact on, on, on the psychological safety when the team, you know, if I hadn't had that health check opportunity around kind of going into the team and, you know, and the team has the opportunity and are encouraged to challenge me, I thought it was a really healthy thing for the team to call me out on it because they were absolutely spot on.

You know, [00:34:00] I should have absolutely had that conversation first. But, you know, how many times do you have this thing where leadership's made a decision and something's happened, the team doesn't understand what it is, and because you've got that disconnect, you know, that definitely damages the psychological safety of the teams.

So I think this, this comes to, you know, open and good communication lines with the teams, understanding what the team challenges is, and the importance of leadership, of having personal relations with people in the team, so that they're aware that they can, you know, that they can have a robust conversation with you, and then call you out if something that you've done.

Your isn't the right thing to have done. Yeah, absolutely. Absolutely. Look, a great conversation this afternoon, lots of room in that to explore more deeply, but it's been a brilliant insight into the challenges of creating agile, both big and small, a product engineering teams in complex organizations. So Steve, thank you very much indeed for your time this afternoon.

Thank you very much for inviting me. Our pleasure. So we end every episode of this podcast by asking our guests what they're excited about doing next. And that might be, I've [00:35:00] got a great restaurant or I'm doing something exciting at the weekend, or it could be something in your professional life. So Steve, what are you excited about doing next?

Uh, so I'm actually, I've come to lead today, so I'm normally based in Chester and it actually has been interesting kind of as we started kind of getting set for this podcast, making all of my tech work in a different location, but I've come up because I'm going to a very good friend's 50th birthday, um, who has a surprise party tonight.

So, um, so I'm, I'm off out for that, which is, it's going to be great. Um, and then heading back to Chester, um, you're on, on Saturday at some point. So really excited for that. That sounds great. A couple of things. First of all, this will go out well after your friend's surprise birthday party. So you haven't just spoiled it, we're not releasing it now.

No risk there on spoiling that. Which is good for everybody, I think. But also, yeah, the arrangements of having to record this over a 5G phone has been podcasting on a knife edge. I think I would refer to it as dangerous podcasting. We should have had danger music going on in the background as we were [00:36:00] doing it like a tightrope walk.

So thanks for bearing with us, Steve. It's been a, it's been a pleasure talking to you. So a huge thanks to our guests this week, Steve, thank you so much for being on the show. Thanks to our echoing producer Marcel, our sound and editing wizards, Ben and Louis, and of course, to all of our listeners.

We're on LinkedIn and X, Dave Chapman, Rob Kernahan, and Sjoukje Zaal. Feel free to follow or connect with us and please get in touch if you have any comments or ideas for the show. And of course, if you haven't already done that, rate and subscribe to our podcast.

See you in another reality next week [00:37:00] week.

About Capgemini

Capgemini is a global business and technology transformation partner, helping organizations to accelerate their dual transition to a digital and sustainable world, while creating tangible impact for enterprises and society. It is a responsible and diverse group of 340,000 team members in more than 50 countries. With its strong over 55-year heritage, Capgemini is trusted by its clients to unlock the value of technology to address the entire breadth of their business needs. It delivers end-to-end services and solutions leveraging strengths from strategy and design to engineering, all fueled by its market leading capabilities in AI, cloud and data, combined with its deep industry expertise and partner ecosystem. The Group reported 2023 global revenues of €22.5 billion.

Get the future you want | www.capgemini.com

Capgemini