Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> *Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems*
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 1**

Shane McGraw: Hello, and welcome to today's SEI Webcast, Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems. My name is Shane McGraw, Outreach Team Lead here at the SEI, and I'd like to thank you for attending. We want to make our discussion as interactive as possible today, so we will address questions throughout today's talk, and you can submit those questions in our YouTube chat area, and your panel will get to as many as they can. Our panel moderator today is Mr. Hasan Yasar. Hasan is the Technical Director of the Continuous Deployment Capability effort here at the Software Engineering Institute. He's also an adjunct faculty member at Carnegie Mellon University. Now I'd like to turn it over to Hasan Yasar. Hasan, good afternoon! All yours!

Hasan Yasar: Thank you so much, Shane. I am really excited today because we are kind of like at end of our four years of journey and building up of DevOps standards, and I'm so excited because there's a tremendous effort on this development as we stand. And every Friday morning, 7:00 a.m. Eastern time, it was kind of my go-to time, everybody knew in my family that it's a IEEE time. Now it's the time to celebrate, and we have a panel of co-authors with us today, and Bob Aiello, and Ruth Lennon, and Paul Bruce, and Altaz and Lynn Carter and Sarah Baker. So, we had a huge discussion, a great topic in talking about DevOps standards. What that means for us, what is the content of DevOps, why do you need the DevOps, why do you need DevOps standards? What is the overall topic? I know there's a lot of content in it. I don't think we can cover up only in one hour, but please, I encourage you to join our channel, trough YouTube Channel, ask your questions, we'll be monitoring, and try to answer your question as well. So, as I said, it's a long journey, and we're going to discuss more. And let's talk to talk about that. So, I'm going to start with Bob first. What does DevOps mean for you, Bob? Why you started, what DevOps in your context?

Bob Aiello: So, to me, DevOps is a set of principles and practices that help enable communication and collaboration. So, the teams understand each other's perspective, and we work more effectively together. So, developers understand where the operations team is coming from, the operations team understands where developers are coming from, and it's wider than that. It includes all of the key stakeholders. And a lot of times we focus a lot on the just the deployment engineering or cloud engineering, but I really think the DevOps culture is the most important factor if you can achieve success. So, really communication and collaboration is key for me.

Hasan Yasar: Anybody wants to chime in? Because otherwise I have my opinion, maybe I can say that, because I know that DevOps is not the culture, but also we have other things as well. Maybe look at the lifecycle, we are looking for including other stakeholders. Other stakeholders like the QM as a

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> *Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems*
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 2**

perspective, and acquisition perspective, right? Paul, you want to chime in for the acquisition perspective, or the QM perspective, Paul?

Paul Bruce: Sure, so even just backing up, right? I work in a lot of various different communities and groups that say #DevOps. And it means different things for everybody. And that's kind of a problem when you go to operationalize what that means. The name was picked a long time ago, right? And since that time, right, I've been realizing that it can't just be scoped to Devs and Ops, right? It really does include-- that's exclusionary. The other thing is like, look, if DevOps can't be applied to high compliance, or enterprises, what does that say about the way the people have synthesized DevOps? There's something fundamentally wrong with that way, right, if it can't also apply in really important circumstances. So, for me, it's not just squishy culture. That's important, the people, process and technology. But you got to ask yourself like, you know, "Is it just tools? Is it just about process? Is it about people?" And it's about a lot of those things, but it's certainly not just about developers and op-ers, right?

Hasan Yasar: Altaz? Can I hand it over to you?

Altaz Valani: Yeah, thank you, Hasan. One of the things that we often debated in our working group was making sure that there's a clear understanding around the business value of DevOps. It really has become a key part of the processes of a business today. And in so doing, what we've tried to achieve here is to create something that's auditable, and that includes both business and technical stakeholders. Many times, we find that the discussion turns just on the technical side, but we found ourselves naturally moving towards including others that are outside of the technical domain, and I'm sure we'll get more into that later on as well.

Hasan Yasar: Lynn, you want to chime in?

Lynn Carter: Yes, absolutely. I agree with Altaz about the business aspect, but there's another aspect to the business that is very technical. Nobody builds computer systems by themselves now.

Hasan Yasar: Hm.

Lynn Carter: Computer systems are built by an ecosystem of organizations who collaborate to produce the products. And all of those ecosystem individuals are changing rather quickly. And so, the culture that Bob is talking about is one of how do we create a calm, planned, organized way of working to ensure that we're delivering that value, delivering that security that DevOps demands in so many applications? And if that isn't a culture issue, I don't know what is?

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 3**

Hasan Yasar: You could claim then that there is a ecosystem that you mentioned, and you're going to dive into the more ecosystem lifecycle perspective. And Sarah has something to say. Sarah?

Sarah Baker: Hi, I'm a long-time Operations person. And DevOps means to me was something that I was always trying to find the name for. And when they came up with the name DevOps, it was like a lightbulb went on. I also spent a lot of time after some of the concepts started coming out reading about some of the places it came from, which interestingly put me into a lot of industrial engineering documentation. And we did a-- I did a lot of reading that industrial engineering learned this stuff like 30 years prior. And all of what I see today is incorporating that into software engineering. And it's great because there's a whole host of material. And we can learn and do the same things. It's very similar, even though people thought it wasn't.

Hasan Yasar: So, by hearing that lifecycle and ecosystems, that remind as well, software is leading the world, right? So, every system that we do, software is part of it. So, let's go to the next topic and I might open up why DevOps matter for us? I know we may be hearing a lot of vocabularies like DevResOps or DevSecOps, I know I'm fond of DevSecOps, I like it a lot, I know it's offensive to Bob, but I like it. But there's a lot of terminology on that topic. So, what do you think about why DevOps matter for us?

Bob Aiello: Well, one of the interesting things here is that even though we're saying DevOps has a lot to do with culture and a healthy culture, and the flip side of it is identifying dysfunction, we couldn't write a standard about culture, right? We could talk about communication and collaboration, we could talk about including the right stakeholders, but we couldn't really talk about culture, per se, in a standard. So, we had to look at what factors operationalize culture. And in practice, one of the most important things is Deployment Engineering. It's Release Engineering. It's making sure you can baseline your systems. You can detect if systems have been tampered with and effectively self-healed. So, we dug into what are the factors that really operationalize DevOps and make it an effective practice? So, the standard really is focused on organizations that have regulatory requirements, whether it be financial services, banking, medical, pharmaceutical, engineering, automotive, aerospace and defense. Organizations that have regulatory requirements who want to use these best practices, and still be able to pass an audit. So, a standard was needed, not so much to define DevOps, but to define how to do these practices in real-world scenarios that most of us find ourselves in on a daily basis.

Hasan Yasar: Altaz?

Altaz Valani: I think one of the things as well in terms of the standard was the response to what we saw happening in the industry. The delivery cycle itself now is broader than just the SDLC, and it actually

includes other areas like compliance and legal and other lifecycles within the business as well. So, the goal here is how do we go in there, and how do we bring these teams together to help manage risk on a continuous basis? And if we don't come up with some standardized best practice way of doing this, then it's really difficult to try and coordinate our efforts. So, I just wanted to put that out there. This is why it's really important and why DevOps matters in terms of the context.

Hasan Yasar: Okay, Ruth, you want to chime in next?

Ruth Lennon: I've just been waiting for this!

Hasan Yasar: You're excited!

Ruth Lennon: I've been listening to everybody speak. And I think we're missing one of the bigger things. When was the last time you developed a piece of code in isolation? That you were the only person involved in the team? And you're not! And we haven't been that way in for-- in many years, let's say. And as well as that, it's about the hardware that sits underneath So, I love to spend my time in the data center when I can, which is not nearly often enough. And, like Sarah, I find that halfway, has a bit of software, a bit of hardware, is great for me! Because a lot of what I do is infrastructure as code. So, DevOps starts to get us to think about the wider perspective. We're seeing systems thinking. You can't develop your small piece. You have to think about how that piece is part of a larger system overall. And that's important!

Hasan Yasar: Isn't that the reaction for how DevOps really started, because of the infrastructure team member, they were a little upset about getting so much changes? Right?

Ruth Lennon: Mm hm.

Hasan Yasar: Something. Let's fix that operational problem that we are hearing.

Ruth Lennon: Yeah.

Hasan Yasar: Paul, you have some thoughts here, Paul?

Paul Bruce: Yeah, yeah. So, like, all right, so when people talk about culture, that they can't change culture or whatnot, I just ask, "What define-- what do you mean by culture?" right? And if we scope to that to like what we do in the standard-- we spend a lot of time, like every single time we wanted to pen something, we had to be very careful not to be prescriptive, right, versus a normative situation. We can't

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 5**

tell you how exactly to do things, but in any healthy culture, aren't there guidelines and principles? Like, for instance, people might know this one, "Thou shalt not kill," or how about the Categorical Imperative, right? If the rule can't apply to you, but only apply to me, maybe we need to think about that a little bit, right? There's many people and many needs, many perspectives. How about, "Clean up after yourself," how many people have been in a coworking space? Or in an office where people don't clean up after themselves? How rude! Right? That's disposal. That's proper disposal in the cloud, and it's not just rude, it's dangerous! Right? You are not exempt from this. And just because you're writing code, and you're not the expert at something else doesn't mean that you get to completely ignore the downstream effects, or even upstream sometimes effects, of what you do. So, my point is, whatever your culture is, right, it has outcomes, too. However functional or dysfunctional you think your home culture is, your work culture is, whatever it is, right? That "culture," that squishy culture term, right? It has outcomes. And so, in our standard, because we can't tell you exactly how to do the thing, this tool to use or that tool to use, we had to focus on, "What are the outcomes of a working process? And how is that different in DevOps?" right? So, what we're saying here, right, is that it's-- you can go figure out how to do those outcomes, but if these outcomes aren't in the final result of having a functioning QA process or QM process, risk management process-- if you don't have those outcomes then something's seriously like getting dropped on the floor. There's going to be a significant gap, and you're risking something really bad for that. So, culture might be squishy to some people, but it always does have outcomes. The question is, you know, the outcome's in the standard, right? How are you going to go about doing that?

Hasan Yasar: I hear you're saying there is kind of like a sans engineering principle behind the DevOps. So, we're going to come back into the details of each of the components. Lynn, you want to give _____ 00:14:39.

Lynn Carter: Yes, one question was asked about the simultaneous engineering or concurrent engineering. And I think this is a critical concept from several perspectives. First of all, to get the velocity that we're talking about, you can't do things in a traditional waterfall. These activities have to be taking place simultaneously. And yet, we need to coordinate those things because it still needs to be a flow design all the way from requirements all the way through to the user experience. And so, how do you do that if things aren't happening concurrently. Now, let's flip to the other side. Where do you find employees, especially new employees, that know how to do this? Especially when I'm not familiar with very many universities, colleges or even training institutions that teach anything other than how you are going to write the entire thing by yourself, okay? Where's the teaching how to collaborate, how to coordinate, cross-functional teams? How many people actually practice those things in the school? So, guess who's doing it? The organization who's producing the code has to do that training. That's very expensive and not many employers are particularly good at doing education. So, DevOps not only is about those people,

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> *Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems*
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 6**

it's about the universities, it's about our training organizations, it's about all of these things. And if they don't all play together well, I mean, that's the big ecosystem! Because how can you be successful in business if you can't get the employees you need to do the work?

Hasan Yasar: All right, thank you so much, Lynn. And before jumping to the, "Why you need DevOps," I see Ruth wants to say a couple words as well. Ruth?

Ruth Lennon: Yeah, I'm agreeing with Lynn in that, you know, it's more than just that, and we need to think from the ground-up, but unless we've got those structures there from management to support the lower-level staff when they need to do training, or when they want to try out a new technique or technology-- or whatever it might be-- if that fundamental support is not there, then we won't move forward. And that's one of the nice things I think we put into this standard is that we are requiring the management to show evidence of support for these processes. It has to be put into your schedule. It has to be given time.

Bob Aiello: So, Ruth and Lynn are both spot on, but one of the things I want to point out is that as a practitioner, I often find that I'm working almost as a marriage counselor. And I have a degree in Industrial Psychology, so it's not that far off. But often, I'm finding myself counseling the development team, the operations team, the security team, the end-users. I'm going around, I'm cutting deals between different stakeholders, and I'm a technical guy! I want to be writing code! But a lot of times getting the people to work together more effectively is the most important aspect of DevOps, and that's how we fly. So, it really is kind of a little bit of the out-of-the-box thinking, but Paul, just back to your point, what I was suggesting earlier is not that we can't transform culture, of course we can! But it isn't something we can legislate in a standard. So, we had to walk a very fine line between what we can put into a standard, and what we actually need to make DevOps effective And I think we did that.

Hasan Yasar: That's actually the next question is, "Why do you need a standard for DevOps?" I mean, there was a kind of comment in the social media, now DevOps became real because we have a standard. <laughter> Right?

Sarah Baker: There's _____ 00:18:11.

Bob Aiello: Okay, but we all know, those of us who are in the trenches, know that a lot of banks, a lot of financial services firms, big companies, they're like a deer in the headlights when you talk to them about Agile and DevOps! So, we tackled the tough task of how do you do these practices in highly regulated environments? So, I've spent a good chunk of my time of the last year helping medical professional being

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> *Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems*
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 7**

about to respond to the pandemic. How do you do these practices when you have HIPAA requirements? And the fact is, you can. So, what we focused on in the standard is being able to accelerate velocity, helping teams be incredibly effective, but still maintain all regulatory requirements and effectively pass the audit as they need to in order to stay in business.

Hasan Yasar: Sarah, you want to chime on this?

Sarah Baker: I hear Bob talking about-- excuse me-- I hear Bob talking about all the soft skills,, and that's all great stuff. And as a techy I want to get down to the brass tacks. And I've heard people say, "DevOps, isn't it just deployment? CICD and all the tools and stuff in the cloud?" And yes, there is that, too, and all of that has to be enabled by an organization that supports that. And so, if you're looking just narrowly as the lion software engineer, yeah, you could-- it can be that. But in order for you to get the support to make that happen between you, between you and ops, between you and security, yes, you have to have tools, but then go across all of those different pipelines, and so that you're all coordinated. And so, yeah, those tools are really important, but you have to have the coordination and agreements in place to make that happen.

Hasan Yasar: Altaz?

Altaz Valani: I think it's important to call out as well that we regularly in the working group when we were developing this would ask ourselves, "What does DevOps do that makes things different?" And I think one of the big things that emerged several times in our discussions was the fact that we really want to focus on business value. Just a real tangible thing that could happen. For example, with the financial services company. They have to have a certain amount of capital reserves. This is a business decision. In DevOps, if we can show that we are providing greater resiliency, if we're able to respond quickly, these metrics that we collect at the DevOps level don't just stay there, but they are directly tied now to business value and what we can, for example, to help manage those capital reserves. This is the kind of thing that's happening today. When we talk about the value of DevOps, it's a part of a digital delivery model today. And this is kind of bringing all these things together, not just thinking about it from a perspective of, "That's just technical," but looking at it more holistically from the business value standpoint.

Bob Aiello: We answered the question, "How does DevOps change the game?" And we all kept saying, "We want to be disruptive. We really want to make an impact on the industry. How do we help organizations develop incredibly reliable and secure systems?" That was really our focus.

Hasan Yasar: It was one of the discussion that always remembering myself, "If I'm an acquirer or a supplier or the architect, what I should do in my job in a DevOps way, what that means to me?" because we're not really creating a new persona, a new role, right? We are creating a new way of thinking, a new way tasking. And that's what we tried to put into the standards. "Here's the outcome. This is our goal, and here's what needs to be done to achieve that goal," and employee perspective, or the infrastructure management perspective, it took me a while, Bob, to change my persona to think about that. What DevOps means to me in a way that-- to keep in memory, "What DevOps means to what I'm going to do tomorrow and next week to change my job in a DevOps way?" Right? So, there was a questions, maybe, Bob, you can answer that question, it says, "Is there a standard available to the non-profit organization for free?"

Bob Aiello: I think we'd have to direct that question to the IEEE, but I'm happy to-- if folks want to reach out to me on LinkedIn, I'm happy to make an introduction and we can ascertain if that's the case. Swiboc [ph?] will certainly be-- we are giving a lot of input on Swiboc, so Swiboc is certainly available for free. I'm glad to follow-up on that. I don't know the answer off the top of my head.

Hasan Yasar: Okay, thanks, Bob. So, I'll hand over to you, Paul, for your feedback on that, maybe the upside of it.

Paul Bruce: Yeah, so I think this is kind of coming back to what Sarah was saying about how like, "Yes, there's a tools component, but there's a process and people I walked over and drew this thing over here. Because it's so-- it's in my mind. It's a three-body problem. Every conversation I have is if we're going to change one of these things, it's going to have an effect on all three of these things, right? You can draw your own bubbles, you can have your own view, and actually the journey matters, right? What's more important as you go through, let's say, your "DevOps journey," or whatever that means, is to be asking, "What does this mean for me?" right? Now, a lot of people don't have the luxury of attending all the webinars that you do, all the things that you go to, all those fancy conferences you used to spend a lot of money on going to, right? We don't all have that luxury. We've got a lot of work to do. So, when it comes to, "Okay, well, I'm in a regulated industry, and I need sort of something somebody else thought through," right? A bunch of people put time and energy, right, many years at this point has gone into the thinking around this, right? And the years before that of all the experience, right? The point is-- and I'm not selling the standard, I like open source just the same as anybody else-- but it was like surprising to me just now to be like, "It's pretty cheap. I mean, you can get your hands on it real cheap." So, that's not a blocker. What the blocker is is actually taking time to say, "Hey, this isn't going to force your culture," right? Like just buying a standard and reading it, and maybe even trying it out in a team, right, is not going to radically force all of your culture to do this thing. But it provides sort of the-- almost like the Christmas tree

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 9**

box-- the Christmas ornaments box, of things that have already been considered and there's a place for things. Right? And which of the-- a lot of the stuff, turns out, you're already doing, right? A lot of the QM stuff, surprisingly, on a day-to-day level, a lot of the devs that I'm talking to are doing QM and QA activities. They don't even know it, right? Because traditionally it's, "Well, that's somebody else's job," right? So, when we have that proc-- a line to the process, right, when we have a standard that's like, "Here's some processes. Here's what's the outcomes. Here's some of the things that go on in there," you'd be like, "Wow! That's me, too!" Right? "I do this. And I do this. And I do this." And that is a representation of how it can actually encourage the culture to continue on in that good path. Right? And it's not just because it's a standards. Because we're now rowing in the same direction. There's an alignment to what that means for you.

Bob Aiello: Paul, you pointed out something that's so important there. The task of doing the assessment, of having the conversations about folks. Asking them what's going well, and what could be improved? The standard provides a valuable comparison chart for us to work from, but it's really the effort, the discussions we have when we help a team transform, that's the healing factor. And people come to solutions on their own; we're just the catalyst that makes it happen. So, the standard's very valuable, but it's the effort to help the team do the transformation where they magic occurs.

Hasan Yasar: That's actually-- I like that that Paul's analogy. We have a bubble. We're not living in bubbles anymore. All bubbles are connected each other. So, we are touching all our work, but there's no way we can start from everything from scratch by our self. It's not writing everything from an end line and everything is by mseylf. It's impossible. We are developing very complex systems, we're all connected to each others, right? So, Ruth?

Ruth Lennon: Yeah, I just want to follow on what Paul was saying. If we're going to do this for highly regulated industry, or even if it's not. Even if you're just trying to get into this space, you don't know what you don't know until after the fact. Hindsight is 20/20. But you know what? Let's start with hindsight first, go to the standard reader, it sets out those steps. And as Paul says, "Yeah, it turns out you're doing some of this stuff. And you just need top uta. Structure around it. And sometimes that's documentation, and you could say, "Oh, god! I hate creating documentation, and there's too much of it, and it takes too much time and I don't want to do it!" And you're right, I agree. But what we're really doing is we're saying, "That's automated for you When you run that unit test or that regression test, that is the documentation, the test itself, the inputs to the test, the expected outputs and the real outputs. You're saving those in a repository somewhere, that's your documentation. There's nothing more to do with that-- well, except for view it-- but there's nothing that has to be written up separately. A lot of this can be done from what you're already doing. You just need to know that that is effective. And you need to store the information. Don't just throw

Carnegie Mellon University
Software Engineering Institute

SEI Webcast

Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems

by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon

Page 10

that stuff away. "Oh, I ran tests on that." "Did you? When? Exactly what version of that software did you use? How do we manage all of this? What kind of repositories are you using? What is your general configuration management strategies?" And I know we're going to come back around to this again. But these all give you a lot of the tasks and processes. And although this standard is written for highly secure, highly reliable regulated systems, you can jump onto this if you're a small company. You don't need a thousand employees. There are easy starting points. And I know that everybody's sick at me saying this, but it's true! There's enough in it for you to say, "I'll pick this bit. We'll move this step forward. We level up and we level up as we go through." It's written into the documentation, which is quite good.

Hasan Yasar: I will hand over to Lynn, and after that, Sarah. So, Lynn.

Lynn Carter: Uhm--

Sarah Baker: I worked on DevOps principles, both in a very small company of under 50 people. And now in company of over 6,000. It's applicable to both, and it can be regulated or non-regulated. It is simple applicable to what software does in almost every situation you can imagine. Okay, there might be different applications in different situations, but that's part of what's in the standard. These are the things that are the most important to focus on when you're doing, whether you're small or big or regulated or not.

Lynn Carter: When it comes to the DevOps principles mission customer focus, left shift and so on and so forth, it's important to recognize that these are not just individual pillars, even though the graphic shows it that way. These are integrated, okay? Everybody has to know how these things fit together, so that we can collaborate together. And not just within our firm, across the ecosystem as well. The challenge here is that we learn those lessons as we do it. Show me one book on project management that teaches people to read lessons learned when they get the key steps in their projects. Lessons learned are not a write-only discipline. But folks, that's what we do. Most of us write things and never turn around and read them. And so, this is another culture change that's going to have to occur if we really want the velocity that DevOps promises.

Bob Aiello: Yeah, the principles are really the whole nine yards. I mean, forget about the rest of this stuff if you have to, but if you learn the principles then you're really teaching people not just to have a meal and not to just have a fish on the table, but how to fish. So, I focus a lot on teaching people the principles. Yes, they're really good practices, such as baselining that are an absolutely must have. But if I teach you the principles, then you're good to go, and then you can practice yourself. It's sort of train the trainer. So, Lynn, principles are really the most important aspect of what we produced.

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 11**

Hasan Yasar: We got to discuss more on the principles and also the lifecycle perspective. There's a question on the audience. The question is, "DevOps is tightly coupled with the Agile software development approach whereas we don't have any standard for Agile. How does just the DevOps standard address this constraint?" Ruth, you want to answer that? And then Bob after? Ruth?

Ruth Lennon: Yep, I don't think that it's so tightly tied to Agile. I think this will work with a lot of different techniques. We did look at that from the beginning and we understand that there are a lot of large companies out there. For example, where I am, there are a couple of companies that are still working off of old mainframe systems. And you're thinking that they're not able to move from those systems, because they're there so long they have to move piecemeal, and they need to be able to implement a DevOps approach to how they move away from the legacy system. So, they don't necessarily have Agile here. Can they still implement the standard? Yes. Can they take the standard in pieces to get to that point? Yes, they should. They shouldn't just go, "Right! We're going to change tomorrow to a new standard!" You wouldn't do that with anything else. You wouldn't say, "Yes, let's drop this entire piece of software and move to a new piece of software," bang, like that! And it's gone! So, why would do so with a whole new approach either. Let's work out, as Bob had already said, you do a review of what you've got, then you do a review of the needs and then align them, set out your plan. It's a terraform plan, it's easy.

Bob Aiello: Yeah, so one of the things that I said through everybody throughout the whole four years, Agile needs DevOps to succeed. There's no doubt that Ag-- the iterative nature of Agile requires automated build, package and deploy and every other practice that we talked about in configuration management and what the cool kids at school are calling DevOps this week. But the reverse is not true. I can do DevOps in an iterative waterfall environment and I work with lots of companies that are using iterative waterfall, and there's no reason, Lean, there's different methodologies that are out there, and DevOps can be right sized to each of those methodologies. Now there are efforts underway to give guidance on how to use Agile in highly regulated environments. And I don't want to go-- I don't want to sidetrack our webinar to go into them, but again, if folks want to be in touch with me, I'm glad to let them on some of the developments that are occurring. Again, we're not trying to define Agile. We're not trying to define DevOps, we're defining how to do these practices in highly regulated environments, mostly where they're already introduced to industry standards and frameworks.

Hasan Yasar: Just kind of one other thought here as well. We were thinking about a system level approach. We have a system thinking, so in the system thinking concept, yeah, maybe using Agile practices, fine, and some of organizations are already using a different practices probably, but think about the systems, because we're all connected to each other. See, in our standard, we always follow the-- what the test case should do based on the roles that I'm doing it? What is required to achieve that

Carnegie Mellon University
Software Engineering Institute

SEI Webcast

Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems

by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon

Page 12

test? What needs to be done? What's my expectations and outcome and what is the test that we have to go for it? So, while we are talking about the standards, there's another questions about the scalabilities, I know, Sarah, that you had the experience that you mentioned about the six to six-thousand people as an example, the question says, "I have a standard, and it is unclear about how just to start on a small scale and help me understand how to start small and grow from there." Sarah, you have an experience. Do you want to give some example based on that experience.

Sarah Baker: My first recommendation is to simply bring the vocabulary of DevOps into your organizations. Use the terms as you talk with people. You don't have to hand them the standard, you just start using the vocabulary. And then some of the principles. One of the things that is in the standard that is very obvious, that doesn't show on the principles page, is creating visibility. Creating that transparency. Just bringing the information that you already have aligned with the standard into visibility, to the other stakeholders that you perceive. Making that list of who those stakeholders are. I spent time actually interviewing almost every manager in that small company to understand what made them tick, and what they cared about. And then I brought that vocabulary with me, and I visualized the things that they cared about on the wall. I put up a TV screen, and I, you know, "Is our system working?" "Yes!" "Where are we in our processes?" "Yes!" I put that up on a wall and a TV screen. And by providing that, just walking by visibility, and having the vocabulary to talk about it, provided a platform on which to start the collaboration that is part and parcel. Working with your leaders, talking to them and seeing-- showing them how that is going to impact what you're going to do. Develop some strategies, get that buy-in. So, it's not just getting tools, it's developing a strategy that includes them into it. So, when you build a strategy about the next data center you have, write it up and go walk over to your security guy before you even have thought two seconds after that. And make sure they have it right in the beginning. That's part of putting that into practice.

Hasan Yasar: So, before diving to the more questions, there was a couple of comment on the chat as well. And Paul, you want to chime on that comment?

Bob Aiello: Yeah, absolutely. So, the first comment was from William was about things like representative stakeholders, and particularly concurrent engineering. So, that's a really good, I mean, go off and research and kind of take a look at the concurrent engineering practices. I think you're absolutely right, right? I just received and appreciate that comment, because there is a lot to do with doing able to co-locate, btu also, especially in our COVID world, right, we're all in some way having to co-locate in the tech space, right? So, now, we just have to do it even better. And I think this kind of runs into some traditional command and control structure dysfunctions. Maybe lack of understanding about how to effectively run organizations when you don't have the luxury of looking right behind somebody to see if

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon** **Page 13**

they're playing Solitaire or not. I'm dating myself now. But you know what I mean? Like the point is, we've gotten into this like hardcore. Not just because of DevOps, but also because of the last year or so, where the notion of, "Where do you work and how do you work effectively, no matter where you are?" really applies there. And the second comment was about-- this was from Sabrina, thank you, Sabrina. "Applying standards is dynamic and needs engineering judgment and general <clears throat>-- general application of common sense to identify how the standard requirements can be met. Objective evidence-based." There is literally the word "evidence" in evidence-based in our standard in multiple different places for exactly that reason. We're on the same page with you. But at the end of the day, if you can't' prove that you're actually following a process, then what's the point of even talking about it?

Hasan Yasar: And you want to add something on it, Lynn?

Lynn Carter: Well, what's really important about the concurrent engineering and the things that Paul talked about is some serious regularization, especially for those people who implement on a cloud, you don't own the cloud. You don't own that environment. So, that's a supplier to you. And for you to be successful, you better have a really good working relationship with the cloud provider. And this goes for the tools' providers, the framework providers. So, as much as you like this idea of having everybody co-located, I'm sorry. Except for Carnegie Mellon, who gets Google to move into Research Lab right next door, most organizations can't get Apple and Google and Amazon and all the rest to move into your building with you. And so, you better start developing the ability to do a cross-functional collaborative virtual team, because that's the only way you're going to compete.

Hasan Yasar: Altaz, I'm sure you have something to say about the business and risk.

Altaz Valani: Absolutely. I just want to make sure that we come back to what's the fundamental objective of DevOps. We want to make sure that we're providing business value at the end of the day. Now Agile is important. Agile ensures that you're aligned with whatever the expectations of the business are in terms of requirements, but let's take this one step further. Now that we're actually delivering things out in the marketplace, what does that mean for the business concerns around compliance and risk and security? Right? It's not just continuous integration, continuous delivery, but it's plus-plus, continuous compliance, continuous security. You've got all these other things now that start to enter into the picture, and building those things step by step as others have attested to is the right way to do this. You can't follow this with a Big Bang approach. But even in attempting to do this, we had a healthy discussion, several times, in our working group. The goal is not to go in and just blow everything up and start from scratch, but it's an incremental improvement over what you've already got today. So, for example, if you've got CICD in place, how do you now take that one step forward and look at how you can inject, for example,

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**   **Page 14**

compliance in there. And if you do compliance then how do you go in there and take it one step further and look at security? But you gradually build this up, and in so doing, what you're finding is the value that you offer back to the business is continually increasing. And so, we don't end up with the situation where we've got technical teams that are focused on automated pipelines and delivery, but not really thinking about how this affects the context of overall business delivery and business value.

Hasan Yasar: Thank you, Altaz. There's a lot of questions come from the YouTube channel. I do like the engagement. I could we can answer all these. If we cannot answer all your questions, definitely we'll get back to you with an answer and they will follow-up the chat room. So, let's continue. I know a couple things we discuss about the business. Actually, the definitely of DevOps within a standard is really a business mission, business-oriented system thinking. So, there was one question, I will take that, and after we'll go to go to the next one. And for the standard for visibility perspective, questions about why we putting up textual writeup, we don't have any graphics. We tried to get the graphics, but we were kind of in that situation, ""Get it done. Get it out of the door quick as possible." And also, there is a lifecycle related activity as well. So, we kind of get DevOps standard from 12207 [ph?] and build up a lifecycle elements in the DevOps perspect-- because the question as we all ask our self, it's a system thinking. It's a system, there's a components on it. Like even when you are buying any tools, you cannot really get the tools right away from internet, you have to acquisition process, as an example. Or if you're hiring an engineer, you have to talk with HR to get the right engineer, if there is an HR pieces as well in the world that we are looking for. If you are looking for any type of platform perspective, there's elements for infrastructure pieces as well. So, kind of more about the lifecycle thinking we implemented, so I hope we can get more graphic representation in the next version definitely, but right now we're just getting what needs to be done, what the works are for a specific requirements. So, there's a questions from Jill, "And is containerization] a prerequisite for DevOps or just an enabler?" Ruth, I know you like operational stuff, you like infrastructure. You want to take that question?

Ruth Lennon: I'd love to. Yeah, it's not a necessary component. It's an enabler, certainly, but there's other ways of dealing with it. We can do DevOps with AMIs from AWS or pick your preferred care provider. You don't need to go there. You don't need a specific tool. And I find it interesting that people are picking specific tools or technologies and saying, "Does it work for this? Does it work for this?" It's written in such a way that it will work rather broadly. So, yes, it will work for containerization. And we don't have to think about things like Docker and CooperNet. Go way back to the early star Linux containers. It still works for them! It works the way across. And in a similar question, there was a question about, "Does it work for MLOps?" MLOps, DevOps, WebOps, every one of those Ops-y type things, or WebDev-- or WebKitchenSink, as Bob had described it. It covers them all in that we're talking about a process. We're not talking about the individual products or tools that are needed to get there. It's again, the idea of

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and
> Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz
> Valani and Ruth Lennon**   **Page 15**

changing your thinking. Changing your mindset to think about the higher-level vision or where we're going. Get Altaz' idea of, "Let's focus on, "What is the mission? Where are we going with this idea? And how are we going to measure how we get there?" The tools you use? I don't need to know how to split a car engine in order to be able to drive the car. I might know how to, but I don't need to know how to. I can still get there. So, let's not focus on the tools and say, "Yes, we're focusing on the process, and yes, it can be done."

Hasan Yasar: That's a good reminder for me. You're always asking our self, "Why I need to do that? What am I expecting? Why I have to solve this problem?" which is kind of outcome section, and, "What task needs to be done?" and for to achieve the why, which is about the what? And then how it's basically-- let's get the containerization as an example that we can achieve through technology. I know you want to say something, Bob?

Bob Aiello: Yeah, I think Octavio [ph?] has volunteered to sed us some diagrams to include in the next revision of the standard. So, we are always-- we are planning your next revision, and I know that we would be thrilled to have folks help us with the graphics. As all of you know, I was born blind, and I don't have a good sense of visual stuff, even though I can see now. So, I would love to get some help with those graphics. And Octavio, you could consider yourself part of the team. We're looking forward to your contributions.

<overlapping conversation>

Hasan Yasar: <inaudible 00:46:09>

Lynn Carter: If I could jump in on what you said. If I could jump in. The best way to get a free copy of the standard is to join the working group, and help us make the standard, okay? That's the cheapest way to get the standard. And so, folks, if you think this is important, don't just simply look for free handouts, come help make it work!

Ruth Lennon: Mm hm, yeah.

Bob Aiello: Your rate that you're going to get paid will end up being about ten cents an hour.

Ruth Lennon: Well, we--

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon** **Page 16**

Bob Aiello: But you're going to have a great time! <laughter> And we are-- we have so much fun together. These sessions are just amazing. We have so much fun together. Yes, we are certainly-- I'm also organizing the Configuration Management standard, which is 828. Get involved and you'll find this is one of the most rewarding things that you can be involved with.

Hasan Yasar: So, a lot of comment, questions getting us. So, I would like to cover up the rest of the content as much as possible. I know we would like to continue with another webinar, most likely, because a lot of things to discuss. So, let's talk about the lifecycle process. So, what does really overall the topics that we covered in the standards with lifecycle. Bob, do you want to chime in with lifecycle process, like agreement, organizational, technical and technical process management and technical process?

Bob Aiello: Yeah, so one of the things that we decided to do early on is to follow a standard called 12207, which has a comprehensive list of lifecycle standards. And we essentially went through each of those topics and explained how DevOps changes the game. So, our standard closely aligns to ISO-12207 and ISO-15288. So, that include agreement processes, such as acquisition supplied, include organization project enabling processes, technical management processes and technical processes. So, one of those is configuration management. And you know, there are many different processes that make up the lifecycle. And we went through each one and said, "This is how DevOps changes the game." So, I know everybody on the team has got their favorite processes to talk about. Mine is CM, because that's where I've really focused a lot of my work. So, I think we all have a lot to contribute in terms of talking about these different processes.

Hasan Yasar: So, there is a couple question again for the chat room. If you're going to be part of our working group as Lynn said and Bob said, please reach out to either any of us, we'll help you either through LinkedIn, or send email to SEI will help you coordinate that, so we can get you into the team. Okay, Bob, myself, or anybody can help you. So, let's continue for the agreement process, because that's kind of like a next step. Agreement, kind of how it started, right, in lifecycle perspective. So, we have an agreement, we have our suppliers. So, even though, you know, how you regulate and why, we have a good side of the house. Right? We would like either acquiring the software or acquiring a vendor, or acquiring a contractor. Because there's going to be-- we're not building it from our self. We need somebody to help us. So, let's talk about agreement process in a minute. Ruth, do you want to chime in for that topic a little bit?

Ruth Lennon: Yeah, if you don't mind. The acquisition process and agreement process have a lot in common with each other. And I think sometimes that's something that's missed. We'd love to be able to say, "Yeah, we just buy that product off the shelf," but you don't. You spend time researching that product

Carnegie Mellon University
Software Engineering Institute

SEI Webcast

Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems

by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon

Page 17

and seeing if it's fit for purpose. You also decide is it something you want to develop in-house or not. And then from that, you're going to compare them up, and you're going to talk to the stakeholders. These are things that you already do. Now do you do it informally, or do you do it formally? And write down the decisions that are made and why you came to those decisions. Those are quite important. When we describe this inside the standard, some of the things that we've added are that you provide evidence of capabilities in security areas, and then we list out a bunch of the security areas. If you're using a piece of open-source software or if it's commercial product, you're going to have to evaluate that code and say, "Well, is it open? Is there security risks? Are there ports left open? Are there default accounts that are left on the system, and if so, how do you deal with that as you go through?" So, you might decide that you're in such an industry that you need to write it and close it off yourself. Or you might decide that the best possible way to do this is to purchase it. But you don't purchase it without doing due diligence.

Hasan Yasar: Sarah?

Sarah Baker: After you-- and Ruth was talking a lot about agreements and supply and acquisition having to do with maybe external entities, but as organizations grow, that is happening internally as well. So, consider it both an internal activity as well as an external activity, let's say third party. Internally your multiple teams that you're interacting with, and as the organization grows, that becomes harder and harder. And so, creating that same sort of perspective and approach to just agreements with other teams is going to be assisting you in terms of having the agreements in place about what you will operate, what you will supply, when you will supply it, and who do you interact with so that you can stay in coordination over time.

Hasan Yasar: Altaz?

Altaz Valani: Let's put kind of boots on ground here and talk about some of the things that are happening with the businesses today, right? It's a lot of people are talking about going digital. This is a real thing that the business is interested in one of the responses for trying to go digital is in an era today where we've got a largely remote workforce, you know, perimeter-based defenses, for example, have gaps in them. So, we're going to move let's say, to a zero trust kind of model. You know, so these are tangible things, and when it comes to acquisition, if we just jump on acquisition with just a singular focus on, for example, productivity, or speed to market, we miss out on a whole slew of other things related to compliance and security and risk and a bunch of these other factors. So, from an acquisition's perspective, what DevOps does is it provides the right balance. Yes, you're trying to enable the business, but you're also at the same time trying to ensure that all of these needs are balanced so that when you achieve the end objective and you've got something in deployment, you're not caught with something

Carnegie Mellon University
Software Engineering Institute

SEI Webcast

Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems

by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon

Page 18

where you're having to now respond instead of having to think through this beforehand. And that's really what DevOps is all about. It is, in fact, trying to get people to think more proactively with a wider lens than we have historically.

Hasan Yasar: Great. So, there is another questions on the audio side. I'm just going to continue our slide deck, but I need to answer that question. So, why not DevSecOps? We're talking DevOps. When you started DevOps, about four years ago, and by definition of the standards, we are saying security should be part of it, because we are building reliable secure system as-- while we are doing DevOps. So, will we have a continued security section on the next version of the DevOps standards? And we're looking for to collaborate as well. So, let's continue for organizational project enabling process, which is the next process in lifecycle perspective. I know a lot of thing to talk about, let's cover up as much as we can in nine minutes. So, in organization project enabling process, which is a lifecycle management, infrastructure portfolio and HR and quality knowledge. So, Bob, you want to start that discussion?

Bob Aiello: Yeah, so they're all really important, but knowledge management is just the Holy Grail. If you teach people to share what they know, if you make teaching each other and sharing knowledge a normative culture, if you really promote knowledge management, then your DevOps culture will fly and teams will be really effective. Too often, silos keeps secret their specific details on how they do things, and that causes all sorts of dysfunctional behavior. My most effective weapon when I'm doing this in the trenches, I get people to share their screen and teach each other. And once I establish that as a culture, the team flies. So, I think there's a lot of good factors there, a lot of good processes, but to me, knowledge management is top of the list.

Hasan Yasar: I know we spent a lot of time in the knowledge management pieces, which is our learning pieces with Altaz. Altaz, remember that journey we spent together? <laughter>

Altaz Valani: Absolutely. Absolutely. So, we've really got a portfolio f many different ways that we can achieve knowledge management today, right? We've gone from having things that are hybrid situations, and then we've got-- gone from there to having offline courses to really at this point, when we think about DevOps you want to try and get to the point where in situ, if you are a developer that needs assistance with either risk or compliance or security or something, you've got something that is very actionable that you can now go in and work with that particular requirement, and it's now, from a DevOps standpoint,, you're looking at the interactions really being translations across all of these different domains, which we haven't really done efficiently in the past.

Hasan Yasar: Paul, for the QM, so you spend a lot of time in QM/QA.

Paul Bruce: Yeah. So, yeah, so I worked a lot with the-- a couple other people, Kathleen Ramas [ph?], who isn't here today, but also worked in ISO. I love the comment about that there's an ISO for how to do other things in the YouTube chat. The QM, quality management process, quality assurance process, verification, validation, right? These are different things. Most people are so used to thinking about, "Oh, it's just QA," and especially thinking about, "QA is this other department or center of who cares, that we can do without, especially in DevOps, right?" No! No! First off, it's not a silo. <laughs> And second off, even if you go up a level and you say, "These are important things to have in place," let's pause on who exactly does what for a second, these are important things to have in place, right? The management is not a manager with a beeper on the belt, and the fanny pack from 1990, sort of project management. We're talking about making sure that there is a layer of thinking about exactly how the process is being developed and iterated on, right? Not just what's in place, and let me look over your shoulder, right? So, anybody on this call who had any myth about that kind of stuff, hope I broke that for you. But the broader thing is, I've also worked with a couple of these other ones, the human resources. Lynn did a lot of work on making sure a human resources process, right, should really be in place, because we're-- in the DevOps space, at least in Boston, we're having a hard time trying to figure out how to hire the right people, and what are the right people? And how to get qualified people in place, right? Do we just use recruiters and that gets tricky, and what does an open req look like? Is it just all about what technologies you use? What about all that other stuff that we said it's important for knowledge management to be shared and transferred. Effective communication, collaboration. Right? These things are not in the classic view of how to get bodies to do some things for you. And so, on a DevOps mindset, human resources and knowledge management and education are critical components.

Hasan Yasar: That's a great segue to the HR perspective and other things, which is a very important for the organizational structure about the HR. Ruth, you want to chime in on the HR a couple minutes?

Ruth Lennon: Yeah, it kind of follows on from what Paul was saying, but also what Lynn and I have been talking about for years now is that you can train people up on the individual tools, but they're only good as that tool. And that's not where we're going. And a lot of the universities don't really think about the process and why we want to train them up, and how we want to train them up. And remember, this is a continuous learning process. If I get you as a first-year student today, in four years when you come out with your degree, there'll be a whole new set of tools by the time you get into that industry! So, it's not that that we're trying to teach. We're not trying to get people to be as much aware about those tools as the process. And then you find the tool that works for you. Now when it comes to the HR themselves, we also want to think about how these job ads are written. And we've all seen the funny job ads having ten years in experience in a tool that only exists for the last four years. It's about collaboration with HR, and making them realize what you're expecting as part of your team. And describe it in terms that are more

Carnegie Mellon University
Software Engineering Institute

> **SEI Webcast**
>
> ***Announcing IEEE 2675 DevOps Standard to Build Reliable and Secure Systems***
>
> **by Hasan Yasar, Bob Aiello, Paul Bruce, Lynn Carter, Sarah Baker, Altaz Valani and Ruth Lennon**
> **Page 20**

appropriate to a DevOps process. That way, somebody who may not have experience with that tool, but they have the right mindset, and they're willing to learn, they're the people you want. It's rescaling. It's showing that what you've got is talent, and how do you apply that talent? I'll keep it at that, because I know it's short. But I that's really important to get right.

Hasan Yasar: Actually, I was expecting that actually. It's a great discussion so far. I know we have another cover-up to two major topics about a technical management and technical process pieces. And I would like to ask everyone that's available at this meeting, can we do another webinar, deep dive into the technical management, technical processes, the kind of meat of the standards. So, we can schedule another deep dive talk and go into details and answering that and take the same question, because there's about the CICD perspective. There's a verification/validation perspect-- there's a disposal perspective, so we can cover up more in details. So, just to wrap up to today, I know we covered up the more what DevOps standards it is, and what the common principles are, what the common processes are. So, lots of things to talk about it. So, we have about one minute to go. Any last word that you would like to say to the audience. I'm just going to call out one person at a time. One word. Bob, you want to say one word, Bob?

Bob Aiello: Yeah, I think there's no longer an excuse for systems being penetrated. Systems should be secure, reliable. We've thrown down the gauntlet, we can tell you how to do that. No more excuses.

Hasan Yasar: Sarah, one word?

Sarah Baker: Operations and you know, what started with Dev and Ops has moved across the entire value chain of your product. It's everywhere, guys, and it's in everything. Keep thinking.

Hasan Yasar: Ruth?

Ruth Lennon: Communications are key. All of this only works if you communicate with all of your stakeholders. So, think about who your stakeholders and clients actually are.

Hasan Yasar: And Lynn?

Lynn Carter: Education is important, but folks, it's practice that really gets us what we need. People have to be able to do this stuff, and not just do it, but do it when they're under stress and when they're under pressure. And that only happens when people practice. So, let's practice before we start to perform.

Hasan Yasar: Paul?

Paul Bruce: Growth mindset.

Hasan Yasar: And Altaz?

Paul Bruce: As the teacher said to my eight-year-old in school. And this is-- growth mindset is all over underneath this standard. So, every time you hear in it "As much as possible automated" or, "This should be automated," right? That's your CICD question, but it's also, "Why wouldn't we be doing that?" Growth mindset, people. We could. We could automate that, right.

Hasan Yasar: Last word from Altaz.

Altaz Valani: Focus on business enablement and risk management.

Hasan Yasar: So, thank you for all. I'll hand it over to Shane. I know we passed about a minute. Thank you for all the participation. Shane, the floor is yours.

Shane McGraw: So, a great discussion today. Thank you to each one of our panelists, we appreciate you guys sharing your expertise. And lastly, we'd like to thank each of you for attending today. On exiting, please hit the like button below if you found value, and share the archive, which will be found at the same URL. You can subscribe to our YouTube Channel by clicking on the SEI Seal in the lower right corner of your video window. Lastly, join us for our next live stream, which will be May 11th, and our topic will be, "Service Level Agreements" with Matthew Butkovic and Alan Levine. Information to register is on our website now and will be emailed out as well. Any questions from today's event please send to info@SEI.CMU.edu. Have a great day everyone. Thank you.