

SEI Podcasts

Conversations in Software Engineering

Atypical Applications of Agile and DevSecOps Principles

Featuring Lyndsi Hughes and David Sweeney as Interviewed by Suzanne Miller

Welcome to the SEI Podcast Series, a production of the Carnegie Mellon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by the U.S. Department of Defense. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

Suzanne Miller: Welcome to the SEI Podcast Series. My name is [Suzanne Miller](#), and I am a principal researcher in the SEI's Software Solutions Division. Today, I am joined by [Lyndsi Hughes](#) and [David Sweeney](#) to discuss their application of Agile and DevSecOps principles in complex systems, hardware, and other atypical uses. Lyndsi is a senior systems engineer in the [SEI CERT Division](#), and David is also in the CERT Division as an associate software engineer. Welcome to you both.

Lyndsi Hughes: Thanks for having us.

Suzanne: Really glad to be speaking with you. I have worked with Dave before. I have interviewed Lyndsi. So this feels like old home week. It is all good. Lyndsi, we have had you on our show before with [Vanessa Jackson](#) discussing your [framework for DevSecOps adoption](#), and we will link to that.

David, you are new to our podcast series, so you get to go first in telling us

about your background, what brought you to the SEI, and what is cool about the work that you do here?

David Sweeney: Sure. What brought me to the institute was, truthfully, this was my first position out of college. That being said though I have stuck around for a while now because the work we do here is so interesting. I think one of the things that really separates us out from other locations is I am doing new things very regularly versus other friends who I am like, *Hey, what's new? Oh, you know, working on the same project that I have been working on for five years.* And it is like, *Oh, that's fun. I mean good for you, but that's not quite the cup of tea I like.*

Suzanne: David, are you trying to tell me that you are easily bored?

David: No. I'm just always fascinated and interested. We keep it interesting here.

Suzanne: I agree. I have been here for going on 20-some years, and I am easily bored. It has got to be interesting if it is going to keep me here. Lyndsi, some of our audience members may not have heard your [other podcast](#). Can you also tell us a little bit about yourself as well, and what brought you to the SEI, and what is cool about the work that you do here?

Lyndsi: Yes, absolutely. I have been at the SEI for about 12 years now. Like David, this has been my first job straight out of graduate school. I have very much enjoyed my time here. Unlike Dave, who has been focused on a lot of different software projects, I tend to be focused more on the infrastructure side of the house and supporting software developers and analysts in a number of different ways. I have lost my train of thought. I get to work on lots of different infrastructure projects that support various efforts, research, and otherwise here at the institute.

Suzanne: Excellent. Let's get into our topic. Modern software engineering practices of [Agile](#) and [DevSecOps](#) have provided a foundation for producing working software products faster, more reliably than ever before. I think we can all agree on that. Far too often though, these practices really focus on the software product, and they don't tend to address the business mission and capability delivery aspects of products that are non-software. We know from working with multiple large organizations in DoD and in my team, in the Agile Transformation team, that you can't just lift and shift Agile practices from software product development into non-development areas. Walk us through your perspective of the traditional landscape and why ignoring the

business mission and non-software capability delivery aspects is a problem.

David: Sure. You are absolutely correct, Suz, that a lift and shift doesn't exactly work, especially when you look at the traditional development of building software, testing software that doesn't translate necessarily to non-software roles. We see that when you do apply Agile or DevSecOps principles to more business side of things. It is still very helpful when you start taking on the principle of always trying to improve, when you think about how, *Let's actually look at our process and start asking the question of why. So What are our pain points?* These kinds of things are very helpful. Obviously from the business side of things, like within acquisitions as well, a lot of the current pipelines just aren't working. People complain. People are upset. There is pain out there.

Suzanne: OK, and, Lyndsi, do you want to add to that?

Lyndsi: Yes. I think in addition to what Dave said, when you are looking at the traditional landscape, particularly Agile, the [Agile Manifesto](#) was written by software developers for software developers. They had in mind a very specific application, a very specific problem set that they were up against and that they were trying to solve. They came up with a really ingenious solution for that problem, but it is very narrowly focused. We think, like Dave said, there are really good aspects of the Agile methodology and of DevSecOps that can be applied in other areas that are not the development of software. It is just a matter of, as Dave said, understanding your pain points and then understanding those Agile and DevSecOps processes and knowing which of those can help alleviate the pain that your organization is feeling in your specific area.

Suzanne: We were just talking with a group of testers this morning actually at a meeting that are bringing [Agile IV&V \[independent verification and validation\]](#) practices into their world, which is a place that you don't normally think about it. That is something that NASA did a few years back that was revolutionary at the time. We are seeing more and more of these folks. I am going to assert, and I want to see if you agree with it, that the pain point that drives many of these non-software organizations to embrace Agile is twofold. If you are on the hardware side, if you have Agile at the bottom of the V, you keep seeing Agile software being implemented faster than you can integrate it. You get this very sort of cognitive dissonance thing going on of like, *Wait, wait, wait. No, no, no, you are too fast. You are too fast.* People start asking the question, *Well, why can't we be that fast? Why can software be that fast, and we don't?* That is assertion number one. Assertion number two is that on the

business side, the many kinds of regulatory aspects of acquisition and various aspects of mission operation, people are starting to question, *How much of that do we really need to do?* The [DoD Adaptive Acquisition Framework 5000.87](#) is the first framework from OSD [Office of the Secretary of Defense] that actually lays out iterative incremental development as a legal, if you will, possibility. That came out a couple of years ago, I think. We are starting to see people saying, *Well, we can do this. We just have to figure out what is the best way and how to address our particular pain points.* Would you agree with those assertions or amend them in some way?

Lyndsi: I will say that I have definitely lived in the land of your first assertion: that software developers are moving quickly, and sometimes they get upset when the infrastructure team can't support what they are saying they need for some reason. I think part of that problem is when you are looking at the capability to deliver value, that pipeline-platform level of a software development shop, those folks, they want to enable the developers to succeed. They also have in mind all of these other requirements that are forced upon them for business reasons, for regulatory reasons, that makes it sometimes difficult to provide to the developers the exact capabilities they want to support a particular project at a given time. You are sort of straddling in both worlds of trying to maintain a stable, secure, resilient infrastructure and be on the bleeding edge of the latest technology, so that your software developers can work quickly and develop products faster.

Suzanne: At the speed of need as we talk about. David, did you want to comment?

David: Yes. I think, Lyndsi, you covered that first point really well, and I am in agreement. The second assertion I definitely agree with as well especially when we talk about policy. Policy should be enabling you. It shouldn't be this pain point that it has become. We are starting to see a lot of people especially once you start taking on this kind of mindset of, *How can we improve if this policy hurts?* Like this makes it a very painful process. *Is this actually a necessary policy, or do we need to amend it in some way to actually allow us to do work and be more agile, be swift?*

Suzanne: Be more responsive.

David: Push us forward. Yes.

Suzanne: In your recent [blog post](#), you called for this expansion of the traditional software-product-focused DevSecOps pipeline to teams who are

focused on other aspects of capability delivery and mission system and business for the organization. Tell us about the new approach. I think we have kind of already covered, what is your catalyst for widening that aperture. We all want to see the whole system move more quickly than just the software move more quickly.

Lyndsi: I think the answer to this question you are asking, Suz, is an expansion of all the things we have been talking about already. There is definitely a need to be able to provide a platform that has capabilities that the developers want and that the developers need to do their jobs. Taking that iterative mindset out of DevSecOps and applying that to the way that you integrate new capabilities and new features into the infrastructure, I think is a really valuable approach. That is what we have seen in our work with some of our customers, and that is what we have tried to implement, even in-house here within our own lab environment that our team manages. It is really all about making things repeatable so that as new people are onboarded, we are able to teach them how things work in our environment. We are able to have them be productive as quickly as possible. And it is automating as much as is feasible. I think when we are talking about the capability delivery pipeline, there are some things that are much harder to automate than when you are talking about a straightforward software development pipeline. It is a matter of automating things where you can, making them repeatable where you can't and iterating over the needs and the requirements and adapting as required.

David: Yes. I mean, the process becomes very much the same on the business side of the house as well, especially when we are talking about some of the acquisitions where a lot of it is very repeatable. So that is good, out of the gate, in terms of creating a pipeline, but we could also then automate parts of it. As we go through the process, a lot of what you see can be automated and that is awesome in terms of...

Suzanne: One of the things that I have seen is automating the preparation for decision-making. You are not automating the decision-making in a business environment, but you are automating the preparation for that decision. You are making it feasible for the collection of the data that is needed for a particular, say, configuration control board or acquisition board. Instead of people spending lots of time gathering that information through email and such, you have got a repository that is the deposit point for everyone. You have got a schedule where if you don't get your stuff in in time, you get an email or some kind of a notification that, *Hey, we are waiting for you before we can go to the next step*, and things like that. They are pretty

simple things when you really think about it, but they are just not the way we have done it before. It is changing people's mindset away from sort of serial email to, *Hey, we just need to collect all this data. It just needs to be there in time for making this decision. It doesn't matter how we get to it, but it needs to be in a safe, secure place, and then we can go ahead and move forward.* Are you seeing some of that mindset shift, especially in the business areas that you work with?

David: Yes. Traditionally it goes from A to B to C for approvals for certain people, but what I have seen is they're also starting to say like, *Does A really need to pass it to B than to C? Or, Can we really just like hand it and get all three approvals at once?* So kind of like this deserialization of approvals as well. We could look at these different stakeholders and say, hand it all to you at once and say, *Just please approve this,* rather than like, *All right, so this guy's approved it. Now, we got to pass it over to the next guy, the stakeholder.* There are ways to also automate, or not automate, but deserialize the approvals as well.

Suzanne: I like that term because that is a lot of what we see happening. The other thing that I am seeing...I work with a large DoD program that's applying these constructs across their entire mission and business space to varying degrees, but the idea of moving away from these asynchronous things like PowerPoint as a way of communicating, having the communication be something that everyone has access to all the time. As soon as something changes, you have notifications. Instead of having to have a presentation built from that data, you just go look at the data. That has been a game changer for the program I am working on. Is that something that you are seeing as well?

David: Yes, pretty much exactly that. The other thing that we are seeing as well is that the business people start to get hungry for...

Suzanne: Oh, yes.

David: Once they get a taste of it, and they start actually enjoying it, they get very hungry. I would almost say ravenous for it, which is totally fine. I am happy to help with that. In fact, that makes it even more exciting. Once you get that, you see that hunger and desire to continue.

Suzanne: I don't know if you have this. We have a special role called a *toolsmith* partly because of that appetite. Because not everybody in the contracting space, for example, is going to be able to work with the tools that

enable a lot of this to happen. But they have somebody assigned to them that can create dashboards, can change the workflow to meet your needs, and things like that. That toolsmith role has been very critical in enabling them to move forward with some of these ideas and to satisfy that ravenous appetite. I'll put it that way.

David: Sure, especially when it comes to leveraging automation to make the business people's lives just a little bit easier here and there. But it is very surprising how excited they get over it too, because it probably hasn't happened before, and it is exciting. Your life has been made easier. Your job is easier. You move faster.

Suzanne: I will say I haven't had anyone that I know of that has come into the program and said, *Oh, yes, we did this over here. I know how to do this. We have done this somewhere before.* As you say, it is new to many, especially the people in the government space. We are seeing a lot of pathfinders. I have talked a little bit about the program that I work with. Tell me about some of the programs that you are working with and what you are doing with them to help them move away from just software product development as the focus for Agile.

David: Sure. One engagement was with the Navy. I was taking their acquisition, their existing acquisitions process, which was a paper process, and then turning that into one at the start, just a digital process of, *Here is the process as is.* And then start iterating over it. And then see, *All right, so how long is it taking? OK. It took them X long. That's fine. Where are the pain points?* Then let's start seeing where we can leverage, for instance, automation for once the, we'll say, issue or the contract is in a done position, and its period of performance ends, then let's close it and just save them the effort.

There was a lot of automation we got to leverage. That is when we started, especially with the automation part, that is when they got ravenous for more changes, and yes, this is awesome. Let's keep going.

Suzanne: Lyndsi, what about you? What kind of cases have you been dealing with in this area?

Lyndsi: Yes. My area of focus here has been on the development platform side of the house. We have worked with a customer who, they are developing software products for their customer, and they want to make sure that those software products they are producing are of high quality, and that they are reliable. Part of our responsibilities in terms of supporting their efforts to

develop their software have been to bring in some of these ideas of iteration, automation, and testing, testing, testing. We are helping them to develop and build out the infrastructure that they can use to run their CI/CD [continuous integration/continuous development] pipelines to automate tests at every stage of their software development. We are working with software people, so having these software development mindsets is...They are helpful in some ways, but they just don't always apply. We have to get a little bit creative with how we can take some of these DevSecOps principles and use them to help make the platform better for the developers.

Suzanne: I know people that don't understand the pipeline kind of things that we are talking about, when you start building an infrastructure for a pipeline, you are talking about everything from getting [ATO, authority to operate](#), certifications on the software that you want. *Oh, by the way you are never going to get all the software that you want to be aligned.* You also have to get ATOs for the hardware that you want to use. If you are not doing approved cloud stuff, you have to get authorities/certifications for the on-prem hardware. *And oh, by the way, you have got to have all the licenses. And oh, by the way, you have got to have the contracts for the people that are going to maintain that hardware.* This is a very multi-dimensional ecosystem if you have never run into it. Trying to apply Agile when you have got supply-chain issues because you can only get three-fourths of the parts that you need—we ran into that ever since COVID—those are real challenges.

What are some of the problems and challenges in applying Agile and DevSecOps in atypical scenarios that you have run into? Can you walk us through any of those?

David: Sure. I will start us out. From my experience, it has been more of a people problem, or probably not people, but an organizational friction at some point.

Suzanne: Problems in the social system.

David: Yes, not with the system itself of, *Why are we doing this? We have always done it this way*, is one of the classic arguments. Hitting that friction is unfortunate, but most of the time you can talk through with these people by being open-minded and curious as to why they are so against it. If you are being hyper-inquisitive it is going to result in that individual or individuals digging their heels in further.

There are ways to get around this by showing metrics, to show them that, *We*

are doing faster. We are moving faster. That can help swing them. But also, really trying to understand what their pain points with it are. If it's simply the answer of, *We have always done it this way*, that is a pretty easy argument to win. That is not exactly a valid argument by any means.

Suzanne: I don't know about you, but when I hear that statement my first answer is, *And how is that working for you?* In almost all cases, these are people who are too busy to take training, who are too busy to do this or do that. *The way we have always done it* isn't really working for you. Lyndsi, what about you, what are the things you have run into that you want people to be aware of?

Lyndsi: One thing we have seen is an over adoption of an Agile mindset. Where we have seen people constantly jump from problem to problem to problem to problem. They say they are being Agile, but they are totally losing sight of the overall mission of a project.

Even when you are being Agile, there is still a need to maintain...keep your eye on the prize, understand what the mission of the project is. Within that boundary, being Agile to help achieve your team's ultimate goal.

Suzanne: We are not looking for chaos.

Lyndsi: Exactly.

Suzanne: We are looking for adaptability. We are looking for agility in the sense of being resilient. Chaos is not the goal. I recognize what you are saying about over adoption.

Dave: Sometimes it seems like people trade...They put the Agile hat on and are like, *Yes this is great.* Then they immediately take it off and put their firefighting hat on, and are like, *But this is also great, and this is the same.* No, it's not the same.

Suzanne: It's not the same. Those are valid problems. The problem I alluded to as well is, *I'm too busy. I'm too busy to do anything differently.* Again, how is that working for you? Are you getting done what you want to get done? Are you happy being that busy? Some people are. Many of us like to go home and have dinner at night at a reasonable hour. Too busy is too much. They are all problems that I recognize coming into organizations that are trying to do something different. Just be aware of those, so you are not surprised if you see those.

Dave: Yes, exactly.

Suzanne: Dave, you mentioned some benefits, the ravenous appetite. What are some of the other benefits that you have seen that you would want to make sure people know about that are thinking about doing this?

Dave: Other benefits. Not to put words in other's mouths, but morale seemed to be improved quite a bit in terms of they are not getting yelled at as much about how slow they are. That is always a good feeling. Also, it wasn't as painful as starting to walk around the office and say, *Hey, who has got this? Where is this at?* So there was a better communication as well. That was a great benefit. The transparency as well in terms of everyone to look and see in the process in the pipeline. What I am getting at there is improved transparency and improved communication is a wonderful thing.

Lyndsi: Yes. I'll add to what Dave said and to that list is improved collaboration. When people can look at a ticket in an issue tracker and say, *Oh, OK, so, so-and-so did this and then so-and-so did that, and here is what is left that needs to be completed before we can say this project is done*, is so helpful in, number one, assessing how much longer it is going to take before that work is completed. And number two, knowing who to talk to about what might be their blockers. Is there anything you can do to help them get it done faster? It is just that collaboration is just streamlined so much when you start implementing these practices.

Suzanne: Yes, and I have seen similar benefits. The other benefit [is] I have seen some explicit reduction in meeting time. Where there used to be three meetings a week to deal with a particular set of issues, and now that is down to two meetings a week. Now, that doesn't sound like much, but those three meetings had over 50 people in each meeting. If that meeting is an hour and a half, I have saved 75 hours every week of people's time. That builds up. I am betting that you saw some of that in the areas that you are in as well. There are a lot of things that are more qualitative but there are also some quantitative benefits that we can look at some of these scenarios and find.

We like to emphasize transition in these podcast series. If I am an audience member out there looking to expand Agile and DevSecOps—let's say I'm a contracts person—beyond traditional software, what steps should I be taking, and what resources are available to me that you can talk about?

David: I am going to be a little cheeky. Check out the [SEI blogs](#). We have a lot

of really good information on this but in particular, as people are adopting this, you might find yourself hitting challenges. So a couple [of] years ago, [Joe Yankel](#) and [Hasan Yasar](#) had a [blog over challenges to implementing DevSecOps and how to potentially overcome those](#). I am just going to throw that one out there as a very useful article. We do have articles as well on some of the more challenges that you face. Outside of that when we talk about how to start, I mean you could read like the [Agile Manifesto](#) to get an idea. There were the other existing DevSecOps or even Agile principles and see how people have done it and start asking yourself, *Where can I apply this? How can I apply this?* If answer is, no, *Why can't I?* And start asking the *why* question as much as possible. But make sure to also ask for like *why* questions that result in actual responses and not yes, no.

Suzanne: Always good advice. Lyndsi.

Lyndsi: Yes, I think Dave hit the nail on the head. I would just add that another place to start looking if you want to implement some of these things, is looking into what tools are available. There is already software out there that can help with some of these communication challenges that you are going to run into and tools that will help improve your team's collaboration. Look at some of those. Look at the features that they offer. Again, get creative and think about the problem you are trying to solve at your organization and how some of these tools might help you overcome it.

Suzanne: OK. I am going to add one that you guys may not be as aware of if you are not in the acquisition community, but Naval Postgraduate School does sponsor an [Acquisition Research Symposium](#) every year. For the last several years, they have had a track on Agile. Part of the idea is Agile in acquisition settings. They make their proceedings public to the extent that the authors allow. That is another resource particular to the DoD acquisition community that may provide some ideas for you. OK.

What is next for the both of you? Both of you said you like working on different things. What is the next different thing you are going to be working on that I can bring you back to talk about? Lyndsi, let's start with you.

Lyndsi: Sure. David and I are actually working together on two new things. The first is we have started taking a look at the idea of [secure by design](#), [secure by default](#) and how that is being applied to software. We are looking at different ways that we think we might be able to apply it outside of the software realm. Sort of within the same theme of what we are talking about today, but a slightly different angle, is how do we apply this idea of secure by

default when we are standing up an infrastructure platform? The second thing we are working on is, in a lot of these projects that we have talked about today, we have noticed that there is a prevalence of a willingness to reject a lot of intensive planning in favor of constantly jumping from problem to problem and putting out fires all over an organization. We are thinking about why that is so prevalent and if we can come up with some interesting solutions to help organizations snap out of that sort of mindset.

Suzanne: I can talk to you guys about that offline because we have run into that same problem. I won't say we have total solutions, but we certainly have seen some approaches that have worked decently. David, are you working on anything where you are not working with Lyndsi?

David: Those are the two current things that we are both working on. I am not working on anything additional because we have our hands quite full with those two.

Suzanne: That's fair. All right, well, I want to thank both of you for spending the time talking with us today. As you know, this is one of my topics of interest and so, having others to talk to about, *What are you learning? What am I learning?* I love that. Thank you so much for sharing. I want to make sure our audience knows, as always, we will include links in the transcript to resources we have mentioned during the podcast. And a reminder to our audience that these podcasts are available all kinds of places, SoundCloud, Spotify, Apple, Google. I am sure there are others. And, of course, our favorite, the [SEI's YouTube channel](#). If you like what you see and hear today, give us a thumbs up because that is always helpful on YouTube. I want to thank all of you again for joining us in the audience, and thank you again, Lyndsi and David, for talking with me today.

Thanks for joining us. This episode is available where you download podcasts, including [SoundCloud](#), [TuneIn Radio](#), and [Apple Podcasts](#). It is also available on the SEI website at sei.cmu.edu/podcasts and the [SEI's YouTube channel](#). This copyrighted work is made available through the Software Engineering Institute, a federally funded research and development center sponsored by the U.S. Department of Defense. For more information about the SEI and this work, please visit www.sei.cmu.edu. As always, if you have any questions, please do not hesitate to email us at info@sei.cmu.edu. Thank you.