Agile Cybersecurity

Featuring Dr. Carol Woody and Will Hayes as Interviewed by Suzanne Miller

---

Welcome to the SEI Podcast Series, a production of Carnegie Mellon University's 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: Hello. My name is Suzanne Miller. I am a principal researcher here in the SEI's Lifecycle Innovation and Automation directorate. I am here with my colleague Will Hayes, a principal engineer who works with me in that directorate, and Dr. Carol Woody, who is a principle researcher over in the SEI CERT Division.

Today, we have a lovely cross-collaboration across the institute to talk about agile and cybersecurity. CERT does cybersecurity. Our team does agile. We are going to talk about them together. Before we do that, I would like both Will and Carol to just tell us a little bit about what is the work that you do here, and what is important about that work from your viewpoint?

Carol Woody: I lead the cybersecurity engineering team. We are really looking at how do we integrate security and survivability into the lifecycle early enough so that we can build the system to be operationally protected.

Suzanne: OK, and Will?

Will Hayes: I work with mostly major embedded weapon systems in the Department of Defense, heavily engaged in Air Force programs, working directly with the program office and helping them to maximize the adoption and the benefit accrued from these new ways of working: the use of DevOps pipelines and ways to accelerate the speed at which we go from an initial concept to no-kidding operational capability. We are trying to dramatically increase the rate of value being delivered.

Suzanne: So, increase the rate of value being delivered while preserving attributes that we really care about is what we are talking about today. So that is the agile and cybersecurity thing. You explained a little bit about agile, but give us a little more about agile and a little more about cybersecurity so people can get an idea of how they have kind of worked together.

Will: Well, so anymore, the term *agile* really doesn't do an adequate job of talking about the challenges being faced, because the notion of seven plus or minus two people sitting on a scrum team working on code that gets pushed out on the commodity hardware—that is not what we are talking about. What we are talking about is incremental realization of capabilities at a system level. The infrastructure required and the good practices that are essential to being able to achieve those goals are really under explained when you talk only about agile. If it is not secure, if it is not high quality, then you are not delivering value. You are not increasing the speed at which needs are being met. You must have the must-haves is one of the ways we talk about it, and so, let's talk about must-haves.

Carol: Well, one of the must-haves is definitely cybersecurity, but essentially how we have had to think about cybersecurity has changed because the role of software has expanded so exponentially in the last 20 years if we want to think about it. It has gone from under 50 percent of functionality to over 90 percent. We are dealing with a much more malleable component that is controlling our systems and technology. We are also dealing with something that cannot be perfect. It has always got defects.

The best we can do is high quality, and that still contains a good number of defects, and our research has shown that a good number of those are cybersecurity vulnerabilities. We are dealing with a product that is going to have vulnerabilities. How do we keep them as low as possible so that we can produce the highest quality and have the lowest number of vulnerabilities in the operational environment?

Suzanne: Will talks about incremental realization of capabilities of the agile-and-lean kind of mantra, and we talk about security and product defects. A lot of those defects don't show up right away. How are we dealing with the incrementalism of agile versus the, *I need to see the product to know if there are operational defects in it?* That seems to be one of the big challenges in talking about these together.

Carol: I think there is another piece of it as well. That is that frequently you are building a lot of pieces or you are including a lot of pieces that were built for other things. It is the reuse that is also tied together. What we care about for cybersecurity is the end product of how all these parts fit together.

What we also care about when you are putting a system together is that when you are going to field it, that you have got the best security in place at that point in time. So, that depends on what you are doing with it, the context it is going into, and that needs to be factored in as part of how you build the system. Right now, most of the development and practices don't automatically include that.

Will: I think we are discovering, and we have known this for a long time, that looking at a product after it is finished and evaluating its value is inadequate. That the investment you make and the care you take needs to match the environment in which you are going to push that product. It is what you do before the product exists that determines to a great degree the level of security, the level of all quality attributes we care about. True, you will never find a memory leak by inspecting a requirements document. So, there are realities of the situation, but the concept of going fast really is predicated on doing it well from the first.

Suzanne: Doing it well in the agile kind of construct doesn't mean doing perfect requirements completely and then doing perfect design completely and perfect implementation completely, right? It means picking, architecting things in a way, so that we can do an incremental piece where we get all of the quality attributes associated with that piece embedded into it.

Carol: But let me add a factor around the decision making and the risk concerns that need to be factored into that. As you are building the code, you have got to be looking at it to make sure that you're not coding in vulnerabilities at the individual component level. As you are putting the pieces together and understanding the personas that are going to be using it, the attacker is one of the personas that needs to be thought about in terms of what you are dealing with. And as you are looking at what you're concerned about, you have to prioritize the threats and the potential impacts that could occur based on the context of where you are fielding this and how you are operating it. That is not traditionally part of, even in the agile development, how they have been building their requirements...

Suzanne: The discussion at the beginning before we implement.

Carol: Well, also getting the right expertise because understanding risk requires the knowledge of how things could occur. Your average stakeholder is not fully aware of how their system can be attacked. So, if you don't have experts there that can bring up those issues early, they are going to be missed until the experts do come in. We have typically not brought the security knowledge in until the tail end, and it's too late.

Will: What it means to do a good job includes the engineering trade space has to encompass these considerations. We no longer can tolerate, *I've done a good job. It works on my machine.* Throw it over the transom into a completely different environment and say, *Well, that's your problem because there is no parity with mine.* That is not a responsible way to build the systems of today.

Carol: Plus, we also have better tools for finding the security problems. Those have to be incorporated as part of what we do in the pipeline, so that when you build software, you run it

through the scrubs and identify the vulnerabilities and fix them then instead of waiting until the end, until somebody runs a penetration test and finds all kinds of gaps.

Suzanne: What are the challenges? We are talking about a concept that we have talked about in other podcasts called shift left, right? From the right-hand side of the lifecycle, shift security involvement to the left to the earlier phases and also the earlier increments.

Carol: I think that is too simplistic. What you are really needing to do is, *Look, I am going to build these increments. So, what's the security I need? For each one of the increments?*

Suzanne: That means I have got to start at the beginning.

Carol: You do have to start at the beginning, but you also don't want to over-build to the ultimate of what you need right at the beginning. Too frequently we see security people bring in a stack of controls and say, *I'll approve you when you put all of these in place*. That doesn't directly link what we need to do for security with what we are trying to do at each point in time. The structuring of how we interface all of these pieces and how we really think about it really needs to change.

Will: I think one of our colleagues, Nanette Brown, has put out a blog post on, *What do we really mean when we talk about shift left?* I think the shorthand that serves us well in getting people's attention may gloss over some of these notions. It is not simply moving something that used to be done in one place to do it in a different place. It is a different implementation. When you try to assure and you try to do these other areas of focus in a different time, in a different level of maturity for the product, they play out in a different way than they used to. The amount of information available to you is different. The infrastructure that you need to build for yourself is quite different. Modeling and simulation and personas and really challenging yourself is an essential part of what it means to truly shift left.

Carol: The knowledge that you need doing each one of those pieces is different because we are looking at having to have the right skills there that can execute the new tools, that can understand the outputs and know what is important and make those prioritizations.

Suzanne: Is the use of a particular type of code scanning tool, for example, the right kind of tool for cyber-physical versus a payroll system? It might work well in one setting but not another. Just because I can check the box that I did this particular code scan doesn't mean that I met the risk needs of this particular setting, or I might have over-met the risk need and overdone it. So, all those things play in.

Will: There is an opportunity that comes from this as well. Our friends at NASA IV&V in the Orion Program, one of the things that their comprehension of what it means to do incremental

release led them to really understand in a different way what it means to add assurance to a program like the one there. It is a very aggressive, very ambitious program, and it caused them to reconsider, reconceptualize what it is that they are working on. They are working on a mission, not a vehicle. So, when we look at the vehicle from the lens of a mission, the areas of focus, the expertise that you need to bring to bear, really fundamentally shifts. And, it is more than a shift left kind of shift.

Carol: Reparsing what's important is what it boils down to.

Suzanne: So how does agile from your viewpoint help with that? How do the principles that come along with agile help you make that shift? Or, another way of saying it is, what are the challenges of that kind of different way of doing business that you may be able to get some relief from thinking about things from an agile perspective?

Carol: I think it is really part of the agile perspective to put information together and to pull people together to be constantly discussing things and working on them as you are moving towards some vision. Typically, security has not been part of that developing the vision. *What does secure look like for this product?* Not, *Oh, you have given me a product, and I am going to test the heck out of it and prove I can break it*. It is a very different way of thinking about it. It is going to require different participation potentially than the people that are currently doing that tail-end brake check so to speak, because they have got to be part of the thinking process, part of the view of how could something be attacked. *Is it a big enough risk now? Or, is that something we need to stage later in terms of prioritizing the issues?* It's not that one group is more important than the other. It is, *Everybody's working together*. I think agile does foster that much more effectively. Also, what we have seen with some of the tracking of defects is done more effectively in the agile environment. Then that can be leveraged to determine potential vulnerabilities that need to be identified...

Suzanne: And, what are some of the risk profiles that are apparent early on? That is one of the things I see is [with] the incremental development, you get a view into not just the number of defects, but the character of defects. Some of those have security implications, some of them don't, but you start to get an early view as to where I may need to spend more effort, time, subject matter expertise to get some of these defects, prevent them from occurring in later increments.

Will: If I can [pull] that thread a little bit, I think one of the fundamental things that we try to encourage our customers to understand is that you are seeking learning opportunities. A larger number, a more frequent experience of learning, *What it is we are building and how this is working*. Having multiple opportunities to look for issues is a benefit, but the fact that we are purposely choosing to make smaller batches of work forces us into understanding the

engineering trade space in a different way. The infrastructure we need to build, to model, or simulate aspects of the system that we are not choosing to build on this increment, forces us to a more formal understanding of really what we mean by these system interfaces, what it means for this to work well, and to really think about *Not just, is there a happy path through this code that leads to the confirmed behavior I'm looking for?* You really need to say, *Are there paths through this code that are adequately walled off?* Because if you're building a stub for a feature that hasn't been built yet, you really need to understand that interface much more clearly than waiting for the integration explosion to work out what the interface really was because you didn't take the time to think about that.

The formalisms that come into play really cause you to do a much better job, I think, when you successfully break things into small batches. The challenge is many people look at that and view it as, *Well, you are just prolonging the amount of time required to do this*. I think the data show us that at the end of the day doing the work correctly the first time as opposed to doing it over and over and over again leads to faster delivery and ultimately a better product. That is something you need to prove to yourself. That is not something that you should take a salesman's word for obviously.

Suzanne: I want to pull on the integration thread because we haven't really mentioned the frequent integration that is typical in agile settings is one of the things that actually enables us to explore some of that threat space and some of that vulnerability space. Because we are not just looking at, *I am building 16 components, and then I am going to do a big bang integration*. We are looking at integrating all along the way and doing it frequently.

Carol: And, the problems can't be verified until you do [actually] do the integration to make sure that the decisions are made. I think too the eyes are on the right kinds of problems at that point. *Does it work well, etcetera?* I think one of the risks that we see is frequently agile is being used as a faster way to drive cost and schedule. Here we are really talking about cost, schedule, and an emphasis on quality, because that is really the driver that is going to get your defects down, improve your vulnerabilities, and the integration pieces as well as the appropriate requirements are important.

Suzanne: From an agile principle's viewpoint, quality is part of what we are looking at embedding into our agile development cycle. Unfortunately, sometimes it is one of the things that gets left out when we are making those drives to cost and schedule. Just like every other process, quality can be the first thing to go. It is not meant to be, but what we are highlighting is if it does go in agile, one of the things though, I think, is you will see that faster. You are going to see the defect profile [go] up very fast, very, very high if you are not paying attention to quality, and that can be a signal that you have got to slow down and pull back a little bit.

Carol: Assuming you have got the right eyes on the problem.

Suzanne: Assuming you have the right eyes on the problem. Absolutely. I agree. Yes, we have got to have the right subject matter experts. *Are there barriers within the security community itself to being part of the development cycle in the way we are talking about, or is that just a myth,* I guess is what I am asking. Or, is there really something that prevents that from occurring?

Carol: I don't think it's so much so that there are barriers as the mindset has been heavily focused on compliance. *I have this list of controls that somebody said I have to implemen*t, not necessarily identifying what are the threats that those controls are defined to address. If that is the way you are looking at the problem space, it is going to be very hard to tie these pieces together. You have to think about what it is I am building differently, and that gets back to your point of, it is a different mindset you have to bring to this construction.

Will: A purely retrospective view of what has been built and whether or not it is adequate, I don't think that's going to suffice going forward. Really the risk management framework is meant to say, *Proactively, what are the controls that need to be in place*? *How do you know that they are serving you well, so that the continued work that you are going to do will continue to lead to the outcomes you want?* Not, Do we look back and make sure the i's were dotted and the t's were crossed as they should have been? That kind of perspective, we can't afford that anymore. The scale of what we need to do and the expansion of the risk space and the number of vulnerabilities we face and how rapidly they change prevent that process from working anymore.

Carol: There is another piece, too, in that the security testing has always been thought about as looking at the system as a whole. So, figuring out how to define that in a way that as you are integrating pieces you can begin to test it, is going to force some changes in terms of how we think about structuring those steps.

Suzanne: We know that one of the shifts in the community has been from DevOps, development and operations integration, to DevSecOps, integrating security into that I think because of exactly what you are saying. If we can't figure out how to automate the security testing as much as we have automated the functional testing, we are going to be at a loss in terms of being able to give confidence to people that all these along the way actually do work together the way they intend.

Carol: Or, you end up with an extra step at the end that slows everything down to catch up with the security pieces, when they can finally look at a total system. That defeats the whole purpose.

Will: It really needs to be viewed as an engineering challenge not a checking, making sure, inventory. The process of building things like software integration labs with frameworks for

assuring good practices, those need to be tested with an engineering mindset, not a conformance to the document that describes the steps you should have followed in doing this. Those are inadequate for the dynamic world we are in today.

Suzanne: So, this is the beginning of a conversation. I am really glad to be a part of it. Where do you see this collaboration on agile and cybersecurity going? What kind of plans do you have at this point for doing work together to bring these concepts into greater focus for people?

Carol: I think we have to draw them into actual projects and identify gaps based on how we see things working versus the realities that we are dealing with. We are hoping to relate it and connect it with maybe some of the work that we were talking about that NASA is doing. But, also, to mine some lessons learned as we are looking through agile projects as they are attempting to try these and see what works.

Suzanne: I know there is at least one place that you and I are working together where there is going to be security risk assessment. There is also going to be agile readiness and fit kind of adoption. I am looking forward to kind of looking at those together to see where some of those mindset shifts may be apparent, either they are happening, or they are not happening. We have a lot of different ways to attack this problem. We are at the beginning of looking at things this way.

Carol: Tied to the group that we are looking at, they are planning to do all of these pieces. So, how do we support their preparation for that in terms of being ready at the right time? We don't have direct answers of that, so we're feeling our way through.

Suzanne: Yes. Which is exciting.

Carol: Oh yes.

Will: I think we are going to see a maturation of the way we think about what our products do for us. Many large-scale programs have traditionally, due to realities they face, focused on the duration of work and the cost of that work. It is as if for some of these programs we have hired someone to fill potholes in our city. We have given them a budget to make sure they don't exceed the dollar amount. We have given them a schedule to make sure they don't work past, but no one goes around to make sure the holes are filled or that the material that is placed in the potholes is adequate to the job. Our focus on what it means to have secure systems, our focus on what it means to do a good job of engineering these things needs to change from that mindset.

Now obviously, the world is not uniformly described in the pessimistic way I have described because there are many very capable, very powerful systems out there that we have come to rely on every single day. But, the scale at which these systems have to operate, the critical

dependence we have on them is making it so that our mindset needs to shift to understand more clearly what it is we are paying for when we buy these systems.

Suzanne: Any closing comments from either of you on this subject?

Carol: We live in interesting times.

Suzanne: We live in very interesting times.

Will: Indeed we do.

Suzanne: I want to thank both of you for joining us today. I think this is an exciting new area for us to be exploring at the SEI. We have some really interesting projects that we will highlight as things go along to help people understand how they might be able to shift their own mindset in these areas. Thank you so much for opening this up, and stay tuned. We will be continuing.

Any of the resources that we talked about today—we didn't really mention very much specifically—but we will make sure that things are in the transcript. There are some things on agile, there are some things on cybersecurity, and the Software Assurance Framework measurement. There is a bunch of stuff that we will add in to help people understand where we're coming from as to why we think this is actually something that we need to be paying attention to. Thanks very much for viewing.

Thanks for joining us. This episode is available where you download podcasts including SoundCloud, Stitcher, TuneIn Radio, Google Podcasts, 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 don't hesitate to email us at info@sei.cmu.edu. Thank you.