SEI Webcast

A Cybersecurity Engineering Strategy for DevSecOps

by Carol Woody

Shane McGraw: Hello and welcome to today's SEI webcast, the cybersecurity engineering strategy for DevSecOps. My name is Shane McGraw, outreach team lead here at the Software Engineering Institute, and I would like to thank you for attending. We want to make our discussion as interactive as possible today, so we will take questions throughout today's talk, and you can submit those questions in the YouTube chat area, and we will get to as many as we can.

Our featured speaker today is Dr. Carol Woody, and Carol's a principal researcher in the CERT division here at the SEI. Her focus is on building capabilities and competencies, measuring, managing and sustaining cybersecurity for highly complex software-intensive systems and supply chains. She co-authored the book "Cybersecurity Engineering: A Practical Approach for Systems in Software Assurance" published by the Pearson Education as part of the SEI series, "Software Engineering."

Now I'd like to turn it over to Dr. Carol Woody. Carol, good afternoon. All yours.

Carol Woody: Thank you. Today I want to share with you the journey we have been taking, researching the challenges of cybersecurity and DevSecOps. This has been a multi-year process, first to understand the value of DevSecOps and what it's bringing to an organization, and then to understand the risks, the risk concerns, and finally to define where cybersecurity and cybersecurity engineering needs to be applied for success.

I want to start by describing what we see in cybersecurity and the concerns and then go into some of the challenges of the DevSecOps pipeline. Next slide, please. There we are. I think we've caught up. We had some confusion here to begin with. So this is the agenda I want to cover for you, describing the cybersecurity challenging environment that we see right now, and then focus on two major areas that we have identified in our research, that are critical to addressing cybersecurity risk. One of them, the DevSecOps pipeline, and the other, the supply chain. And then with that understanding, look at, how do we plan for these? What do we need to do?

I plan to address questions at the end of each section. Shane's going to help me and I hope you will share your thoughts and concerns as we move ahead. Next slide. Let me move to the next slide.

What we find is very interesting and the reason I've listed all of these different changes together, is that we find that few organizations are just focusing on one activity. Instead they are shifting from hardware to software. They're also shifting from Waterfall to Agile at scale. In addition, they're shifting from owned, organizationally-controlled infrastructure to shared infrastructure, as well as compliance is shifting from a verification that's done periodically to leveraging all of these other aspects and moving into monitoring continuously. And then we also have systems that were developed from requirements in architecture and now they are built primarily with the simple parts that come from a lot of different third party sources. As well as moving from the development cycle that was tailored specifically to each individual system under development,

into an environment that now is basically a software factory, where pieces and parts are built in a certain order, assembled, monitored and the system is under continuous development.

Different parts of the organization has latched on to each one of these individual changes and they're pushing them into the other areas without sufficient preparation to make sure that the changes are really effectively monitored and managed. The changes impact engineering, acquisition, compliance development, operations, and program management, as well as other organizational-specific areas. And in many cases, there's no direct collaboration, historically, among these different groups, and so these levels of change are forcing organizational interaction and coordination that hasn't had to happen before. Unfortunately, the result of this area gaps in cybersecurity, because processes and procedures need to catch up with the realities of the cyber risk that all of these different changes are introducing. In many cases, the recognition of these risks is being lost under the pressure of just getting a job done.

So let's look a little more at these risks and why they pose a concern. Next slide. Software is everywhere, and one aspect about software is that frequently, it's not just a few lines of code. Everything you buy is essentially a software platform, that operates with many parts and pieces that communicate to many other parts and pieces. But the key concern and takeaway here relative to cyber risk is that each of these has software defects. Even the best of code has a certain level of defects, and this has been tracked historically over many years of research. What our current research is showing us is that a certain percentage of these defects have to be thought of as potential vulnerabilities, so we are continually increasing the level of vulnerability that we're dealing with in all of these products. Next slide.

Each piece of software is becoming actually not built for purpose, but a blend of new and existing code, aimed at meeting a certain set of requirements. As this code is integrated, it carries along code that's no longer needed. It carries along functionality that may not be part of what's actually being used, but it's there because it was just part of a previous product and it's just carried along. This creates limits in the cybersecurity aspects, and potentially carries additional risks with it, in terms of how this code is handled. Reuse is rampant, but few organizations have really integrated how they effectively manage reuse within their development life cycle, and how all these pieces need to work together. In addition, all of this code that carries these defects are responsibilities of different owners within the supply chain. So we see this tight coupling of how we're developing and the supply chain. Next slide.

It's even more extensive if you look at this example in that there are prefabricated units that are frequently just incorporated into many different designs and versions. That would mean that an organization has limited direct control over what comes in with those particular fabricated units. They're really just like tinker toys to plug and play. This automobile design actually integrates then hundreds of different suppliers, each putting pieces that will then be part of the automobile, and this design is not unique. It's similar to what's happening in almost all platforms now. Each supplier has their own processes and practices, from managing their development as well as the cybersecurity of the pieces they're providing, and the integrator will then have varying degrees of control over what each supplier delivers, depending on the acquisition strategy that's

in place for each one of these components. The potential for gaps is huge, and this is repeated for every product. Next slide.

As we mentioned, there are vulnerabilities with each piece of software, and much of the code that's using those vulnerabilities then needs to be updated at some point in time, when the vulnerabilities are identified and fixes are available. We're seeing, in the National Vulnerability Database that the level of vulnerabilities continues to rise. The integrator must pick and choose what updates they apply, once the suppliers have created the fixes, because the integrators are then going to be dealing with potential incompatibilities that will change what the operational running of that particular integration is. So it's a multi-step process, and an environment that's rich for problems for integration and delivery, as well as a target-rich environment for attackers. Next slide.

The vision we have of all this uncoordinated change is potentially operational chaos. It's 100 percent reactive. Many organizations struggle with this growing risk of cybersecurity challenges. They have to throw a lot of resources at their operational execution to just maintain a steady state. That eats into time and money that would be spent in potentially better ways. So the more that we can reduce what comes into this operational environment, the better off we'll be in the long run. But that requires a lot of discipline and planning up front.
Next slide.

That requires us actually thinking about how these weaknesses and vulnerabilities are introduced, so that we're reducing the impact when it gets to the mission execution in the operational environment. We need to recognize that each step along the way can introduce problems, that testing which would focus on verification of requirements is not going to be sufficient for this kind of support and strength. There are a wide range of methods, practices, guidance and vendor tools that can be available to potentially address some of these vulnerabilities, but they have to be integrated as part of your normal way of doing business and that also adds cost and time to the development. And there's really no guarantee of the end results. Planning is going to be a key part of this to balance between all of these major concerns. Next slide.

So potentially, an integrated response among development, security and operations, using something like DevSecOps, which is looking to coordinate how all these pieces fit together, can help provide improvement, so that we end up with a more consistent and better-managed result in operations. But this is going to require a better job of managing the activities of selection, acquisition and implementation, that are currently scattered all across the organization.

Now that you've gotten a sense for all the pieces and where we are, let's think about what we need to do with DevSecOps and the supply chain. But before I do that, let me check with Shane. Do we have questions that I need to address?

Shane McGraw: We do have one question, Carol, and also I just wanted to give a shout out and thank you to our audience for engaging in the chat, letting us know where we're from.

We've got people from Illinois. I saw Minneapolis, Virginia, California, New Jersey, Pittsburgh, Canada. So we got a great audience. Thank you for participating there and keep getting your questions in there. One question, audience question, Carol, came in: why are organizations trying to change so many aspects of a system in software development at one time?

Carol Woody: Every organization is under budget pressure to improve cost and schedule. Each vendor needs to deliver their goods as quickly as possible to the marketplace to capture as much as possible. And so each one of these changes is essentially providing an element of savings in cost and schedule. But what we have to make sure is that it's appropriately balanced with effective cybersecurity management, so that we end up with an operational product that we can effectively run, and that we're not opening ourselves up to major attacks, like our little picture in our operational environment here. It's a very challenging balancing act that organizations need to have good awareness for.

Shane McGraw: Okay, that's it for now, Carol, so we'll turn it back to you.

Carol Woody: Next slide. The Sec in DevSecOps implies that security is included. But the pipeline actually has to be designed and supported to meet appropriate requirements to make this a reality. Defining those requirements is not insignificant. And then implementing them requires that level of coordination that I've been talking about earlier. Next slide.

I'd like you to have a sufficient understanding, and not all organizations do, of what DevSecOps really is. Many talk about it as though it's a single product, but in reality, it is an organization's commitment to automated existing processes and practices for development, security and operations, in a consistent way that can consistently produce output of a certain quality level, and with certain capabilities and emerging requirements. This mandates collaboration among the many parts of the organization to work together with the supply chain, since suppliers provide many of the infrastructure components, tooling, and in many cases, parts of the product as I've described earlier. So it's very complex, it's very interactive and it's not automatically automated. That requires really, clarity in terms of defining what steps are done when and how they need to work together. Next slide.

The pipeline itself is a system, and it needs to be engineered like a system. There's a process flow for development steps that functions within gateways of security on top of an infrastructure. And all of these are subject to change, and all of these need to be integrated. Each of these areas is assigned to different parts of the organization, so coordination is imperative. Automation will not replace coordination. We're seeing in many cases, one group will implement a pipeline, automate a lot of pieces, but if the group that's supposed to be the recipient of the automated information is not part of the process, things will not happen as they need to. The pipeline can collect lots of data about cybersecurity, but if no one is monitoring and managing that information to effectively address cybersecurity, then the results will not be as expected. Next step. Next slide, rather.

In the building of the pipeline, we have identified that there are four different maturity levels. Pipelines do not spring out of the box fully implemented. There is increased functionality and coordination that comes from implementation and then continuous monitoring and improvement over time. These four levels start from essentially basic execution, moving into then basic automation of things that we know are repeatable, and then managed execution, where you can essentially look at the output of the product and know consistently that you're getting what you expect. And then finally, proactive execution where you can really plan and execute and transition and change effectively, based on what you're monitoring and managing.

The level at which cybersecurity is embedded into these steps needs to increase with each level, and how it works needs to be monitored and managed continuously, since the pipeline is an integrated system, and each system has to change over time. Next slide.

This is a view that starts to look at the complexity of the DevSecOps environment. Essentially, there are business requirements that will drive what each organization needs in its unique respect. And then the product and the infrastructure, which in some cases can be thought of as separate pipelines, need to work closely together. There are, again, different parts of the organization that are affected by these. Planning for how the various pieces of the acquisition and development life cycle will integrate is going to be critical to reaping the benefits of the pipeline. Otherwise, there are going to be hiccups, aggravations, and additional risk that no one will benefit from. Next slide.

The infrastructure will be composed of many elements from shared services, open source, third party products. This inherited risk must be considered in defining the integration of various elements into the pipeline. Too frequently, we're seeing organizations only focusing on the new code that they're developing, and not really looking at how all of this reuse is going to be affecting the risk levels of what they're producing. Some organizations actually architect the product outside the pipeline and feed the detailed requirements for software into the pipeline for development. Some actually deliver software out of the pipeline and the real implementation is after integration with specialized hardware and specialized testing for compliance, that is all done outside of the pipeline. So the pipeline itself can be different parts of the life cycle, depending on what the organization needs in terms of their final delivery. And each one of these approaches puts different cybersecurity requirements on the pipeline. Whatever the role of the pipeline, it still requires this coordination that I keep mentioning, between acquisition, engineering, development, infrastructure, and security, to deliver effective cybersecurity. We find in some cases, organizations are not focused on that within the pipeline. They're looking at the compliance afterwards and basically trying to verify the issues post-development. That just increases the costs, slows things down, and there's too much pressure on budgets to deliver things that way. So we've got to start looking at how all these pieces tie together. Next slide.

To support this integration, we have developed a platform independent model. This captures a baseline set of requirements, processes and capabilities that each pipeline should be addressing to deliver effective cybersecurity. As I mentioned though, pipelines vary, so an organization would then take this independent model and compare it to what they need to make

sure that they have not implemented potential gaps. The next few slides I will go into will give you a sense of what's included in this platform independent model. Next slide.

The team has been working on this material for about a year, and the current model contains system requirements, capabilities, operational processes and structures, roles, a glossary, maturity levels related to these elements that I've talked about, and a bibliography. It's still very much of a work in process. We've done some initial piloting of it, but I would not begin to say it's complete, but it's a good starting point. Next slide.

The model focuses on seven different areas that need to be coordinated for pipeline cybersecurity. And requirements have been developed for each one of these areas. These various requirements have then been mapped to the four maturity levels that I showed you earlier. So this provides somewhat of a road map for how to move the pipeline from each one of these levels, to advance with continuous improvement on cybersecurity. Next slide.

Ten capabilities are currently incorporated into the model, and you'll see those listed under that list of strategic taxonomy. These capabilities are then mapped to the seven areas of system requirements. Next slide.

I've pulled an example here so that you can look at just one of them, and I can explain to you what you're seeing on these numbers on the side. If we look at software assurance as one of these capabilities. Then we can look across the system requirements that are linked to software assurance, and we can see that six of them map to governance. There's one general requirement. Six of them map to the architecture and design. Fifteen of them match to the development, two to the testing, one to delivery, and nine of them to systems infrastructure. So all the components of the pipeline itself are integrated together to deliver elements of software assurance, and we need to take this cross-cutting look, to understand how these are delivered within each individual specific pipeline. This gives us a way to have a baseline to compare a specific pipeline too, to say, are we getting what we expect to have? Next slide.

We've also modeled in the pipeline the activities of a DevSecOps pipeline, and how they connect to the various requirements. In order for the pipeline to deliver effective cybersecurity, essentially it must be designed for cybersecurity, operated for cybersecurity, and monitored for cybersecurity. And I would point out to you that the number of activities in the monitoring piece is quite extensive. This is not an area where tools can be pulled out of the box to help you. These are the unique decisions of what data needs to be organized, and who it needs to get to, that will drive who can see what, and how the pipeline itself can be monitored and managed. Next slide.

Planning is going to be critical to this. As you can see, it does not happen by chance. There are many requirements that must be met and monitored to ensure the performance and the implementation is up to what the pipeline must be designed and operated to deliver. In this updated picture, you will see that there are many points of monitoring with these new figures that have been added, which actually increases the integration and participation on the pipeline.

The monitoring could be done internally to an organization, or externally. And some of this could be automated as well. So there are lots of ways that the pipeline can be streamlined and improved, as you move through the multiple maturity levels.

Let me stop here. Shane, have we got any questions to toss in?

Shane McGraw: We do, yeah, lots of great questions coming in, Carol, so thanks for taking a pause here for a second. Rick wanted to know, what is the importance of an audit throughout the SDLC to finding and maintaining the correct balance you just spoke of a little earlier?

Carol Woody: Well, I feel this belongs into part of the monitoring. Essentially, the organizational group that is assigned to do some of the monitoring is auditing. But typically, they show up once a year to check things out and make sure everything's working. Well, with a pipeline, that's not the way it operates. There's got to be participation that's continuous. So in some sense, how auditing defines its role needs to be tuned to how the pipeline needs to operate. And this gets back to some of this organizational change of how different groups, because of the decisions and choices they've made around adopting the pipeline, are driving change into other parts in the organization.

Shane McGraw: Great. Next, Carol, we got is another question: is DevSecOps the idea that developers take more ownership of the handling of security requirements, or is it more of the injection of security personnel into the software development life cycle?

Carol Woody: I've seen it both ways. In some sense, it depends on how much is actually automated, to make sure that change is-- and verification is part of the pipeline. I would suggest that the most cost-effective way to have developers reduce vulnerabilities is to train them in good, secure coding, so that they don't put them in there in the first place. Then your second level of defense becomes running the tools to see if they missed something. That's much more cost-effective in the long run approach to managing this. But not all organizations are prepared to step in and do that, so there'll be varying degrees of who's assigned what, depending on how your organizational involvement is structured.

Shane McGraw: Okay, two more if we can squeeze in this section, Carol.

Carol Woody: Sure.

Shane McGraw: Then we can move on. The next one is: considering certification alongside accreditation within a DevSecOps pipeline, and with the continued changes for security versus stability software safety, can the two be aligned in cATO? It's a small c, then ATO, cATO.

Carol Woody: Yeah, your continuous ATO effort, yes. We're familiar with that and the challenges of it. It depends on the resources you have involved. It depends on the requirements that you put together for your pipeline and how well you are positioned to have the right knowledge and expertise working on the pipeline. If you have developers that don't really

know what they're doing and you're relying on tools to find everything, then your monitoring and managing is going to be much more complicated. Part of what needs to be handled as well is how our vulnerabilities are prioritized in terms of the find and fix cycle. This gets into aspects of technical debt, and we see many organizations that produce things very quickly, get them out fast and sloppy, and there's a lot of technical debt, and they have to go back and fix it later. That's, again, how the organization chooses to address these pieces. There's nothing magic about the pipeline that's going to fix that. That comes down to people working together and coordinating to do the best job they can, and to use the tools most effectively.

Shane McGraw: Great. Quick housekeeping item. There's some people asking about the slides. I did add a link to the slides. They are available now on our website for a link to PDF copies. I will repost that link so you don't have to scroll back through the chat too far, but they are available now. Last one for this section, Carol, is from Rachel, asking: what is a good approach to having developers become more accepting of development changes like DevSecOps? I find it very difficult for developers to accept change to their processes.

Carol Woody: Some of it boils down to how their job expectations are established. There's a link between how you define their role and how they are managed with what they produce and how they deliver. It also boils down to defining the tools and capabilities and their role, relative to the developer. The tools aren't there to stifle development creativity. The tools are not there to limit what a developer can or can't do. And sometimes, that's how they're used. It is not effective management, if that's the result. The tools really should be there to help the developer, to show them how not to screw up, which is where giving them training and helping them understand the cybersecurity risks they're injecting, if they don't build the code right, is usually the path of least resistance. You can also structure the monitoring so that the developers can monitor their own work. We've seen some organizations that very effectively structure work cycles around letting the developers see how much code they're producing, how they're working, and letting them manage their own delivery schedules, and then living up to what they've said they will do. It boils down to giving them a little leverage in terms of controlling how they're operating, as opposed to just reacting. Again, all of that's how the organization chooses to approach DevSecOps and its implementation.

Shane McGraw: And our colleague, David Shepard, had a good comment in the chat, saying that DevSecOps should make your developers' lives better, and if you make that front and center, we'll get on board. So that's it for this section, Carol. We can move on with your presentation.

Carol Woody: Okay, next slide. As I noted earlier, third party software has been a growing, almost exponential component, and this is frequently replacing hardware. So we're getting more and more lines of code, more and more defects, more and more issues in terms of things that are coming from the supply chain. This establishes a dependency for cybersecurity on the supplier, since all of the software defects and our research has shown that 5 percent of these will materialize as vulnerabilities. If these vulnerabilities need to be managed by a third party,

then that relationship has to be part of what you consider when you're looking at your cybersecurity. So managing the supply chain risk then becomes critical. Next step. Next slide.

There are actually several types of supply chains, and the acquirer has varying levels of control over the supplier's product, depending on the type of acquisition used. This will vary from minute control of every detail, all the way up to just accepting as is, if you're just downloading software from an existing library. That may mean the organization has to augment the code itself, with some additional support, before they actually use it. That changes the ownership, and where you're getting changes and how you're controlling it, from potentially supplier control to inside the organization. Someone needs to have management of these pieces and components to make these decisions, and to determine how to integrate with the supplier. You also want to have one connection from the acquirer to the supplier, to ensure consistency, and to assure that there's sufficient visibility and control of the output. Next slide.

Most organizations use all of these different kinds of acquisition strategies within a single program. And again, what they can do and how they can change and what level of control they have varies. That control varies not only cost and schedule, but cybersecurity. And depending on how the acquirer maps cybersecurity requirements into the selection, acquisition and determination of implementation, will carry with it potentially cybersecurity risk. Next slide.

The reality is that supply chain risk is a growing component with attackers. They're compromising more and more of the supplier's product, which then is transferred into the acquirer's environment. In some cases, this becomes an easier entrée, because acquirers are doing a more diligent job of managing their environment, and in some cases, some of these suppliers, or sub-suppliers, because a main supplier may use a lot of other smaller suppliers or small businesses, and they don't have the level of control and rigor in place, organizationally. The visibility of each one of these new attacks, when it surfaces, then provides opportunity for copycats to do the same thing with other vendors, so that we're seeing patterns of these emerging. Given the volume of suppliers, this can be somewhat of an infinite stream of different ways in which different organizations can be attacked. And you might not be a direct target. You may be a target of convenience, because you happen to be part of the supply chain. That makes the risk level even more uncertain. It's not necessarily sufficient just to focus on what you consider to be critical assets, and what you consider to be high value relationships, because some of these smaller ones with other pieces coming in can be just as impactful. Next slide.

There are many participants in the supply chain management, and effective cybersecurity is going to require coordination among the mission infrastructure, acquisition, development, and compliance perspectives. All of these different groups have their hand in terms of how some of these pieces are managed and organized. What we find in many cases, they don't talk to each other, they don't coordinate. And so in some cases, there are major gaps of how things are done with parts of the organization accepting what the supplier delivers, and other parts trying to monitor and manage them. That creates gaps, confusion, and ultimately, cybersecurity risk. Each of these areas has varying processes, relative to how selection approval and

implementation and monitoring is done. And so that consistency, again, can provide problems. I can't emphasize enough the challenges of coordination among all these elements. And in some cases, the different groups think they're coordinating. They're using language that is relative to their way of operating, and the other groups that they're working with don't understand it the same way. So vocabulary is getting in the way of some of this coordination. When you talk about risk, risk means different things to different parts of the organization and different levels, depending on your perspective. And so we're seeing that a lack of consistency and understanding of what is risk, how is the risk shared, what needs to be done, and when does risk get to be too much that needs to be managed und moved up to other levels of the organization is not consistently implemented and understood. Next slide.

This breakdown of activities and practices and controls within an organization was done to provide separation of duties, to make sure that you have the right authority levels looking at it. But the role of cybersecurity has not really been well integrated among all of these components. But in reality, they all own a piece of the cybersecurity management of supply chain risk. There are activities and elements of cybersecurity that belong to each one of these groups. In many cases, it's not recognized and not understood. So you have to look at acquisition and the development risk, certification risk, mission risk, infrastructure risk, and all of these need to clearly work together, with clear reporting lines. What happens if something is out of sync? Who should be involved? What should trigger a concern about supplier risk? Finding out who's responsible for that and defining that needs to be planned. It doesn't happen automatically. Next slide.

Here's the picture that we have put together of the varying parts and pieces, and each one of those arrows represents a level of coordination that needs to happen. And if you're doing continuous monitoring for your security people, then there are automated feeds of data and information from each one of these pieces that needs to be part of the process. If you only focus on the pipeline, that's not sufficient for supply chain risk management, because your acquisition elements are-- and your implementation and monitoring and coordinating aspects happen elsewhere in the organization. So we can look at one level with the cybersecurity in the DevSecOps pipeline, but then when we look at the supply chain itself, we have to expand our view. And in essence, there's a range of cybersecurity responsibility that goes through each one of these levels. Next slide.

Cybersecurity needs to integrate then with the general processes and practices that each one of these areas implements. They have their roles, responsibilities, rules and guidelines that they have to follow for the various aspects of acquisition, relative to legally handling suppliers and dealing with them, as well as internally in the organization controlling the elements that come in, that are acquired, and how they are monitored and managed, licensing issues and all of those various aspects.

Each supplier really must have a point of contact, so that we have a coordination function. Otherwise, we get confusion with multiple contact points that don't see-- or that only see part of the problem, or only see part of the information, and don't really have a good understanding of

how that supplier is actually impacting the organization, to do effective risk management. The pipeline needs to be part of this coordination, but it's not the total focus. But we do want to ensure smooth integration with the infrastructure, development and delivery of the organization's product through the pipeline. So all of these components need to look at each other and coordinate. This is really coordination at scale, and it doesn't happen by accident. It has to be planned and monitored. Next slide.

We have identified in our research six key areas that have a role in this coordination. For each of these, we are in the process of assembling goals and practices to describe the needed cybersecurity support, what needs to be there in terms of the coordination, and how that can be critical to success. So we're looking at building what needs to be done within each one of these areas for cybersecurity, and then how they need to cross over into the other aspects for addressing that shared information and coordination.

We will essentially be turning this into an assessment that we can eventually share with organizations, to help them identify gaps in their specific implementations, to drive improvement. Hopefully this will be available early next year. We actually have drafts of some of these areas, and we're fleshing out the others to make it a more complete picture. If you would be interested in being part of the review of this process, we would welcome your input, and you've got my contact information within the slides. Feel free to reach out, and we'd be glad to have your support. Shane, let me stop here for questions, if there's something else that's come through?

Shane McGraw: We do, We have one from an earlier section, Carol, so I'll back up and grab that one first. It was: what guidance/standard is being considered with respect to the software assurance capability in the PIM?

Carol Woody: We've pulled in, as part of the requirements, a full range of standards. They're identified in the bibliography that we put together. Basically, we're looking at your standard engineering. We're looking at a lot of the NIST standards and guidance that are critical to aspects of development, as well as, in our case, because we're a research center for the Department of Defense, we've also looked at a lot of the DoD guidance, to make sure that that's integrated in.

Shane McGraw: And is the PIM something that can be shared?

Carol Woody: We're definitely looking at how to do that, and so if that's something that you're interested in, let me know, and we can see about how to do that. As I mentioned earlier, the model itself is under-- it still under development, but potentially we could fork off a version of that to get some input from outsiders, which would always be useful in terms of improvement.

Shane McGraw: Okay, and then one question from this section before we send it up-- or send it back to you for a wrap up. How were the six key life cycle areas determined?

Carol Woody: We basically looked at assembling the experience that our organization had had in terms of what was affecting supply chain risk. We worked extensively in infrastructure analysis, critical infrastructure, for DHS and have an assessment that focuses on the operational aspects there. So we were mining our experience there. We also have many years of actual working with programs, in terms of supply chain risk, and how to address the challenges that we're seeing. So we were basically assembling our experience, putting that together and saying how could we improve and correct these? What will help us there? And so that's where those areas came up with, or from our experience.

Shane McGraw: Great. We're all caught up, Carol. We'll let you wrap up and then we'll see if we get anything more coming in.

Carol Woody: Okay, next slide. As I've mentioned several times, planning is key. And one of the key plans that organizations can assemble is a cybersecurity strategy. So I'd like to have you think about what needs to be done for that. Next slide.

Essentially, in advance of pushing change into an organization, if you think about the planning aspects of cybersecurity, as part of how that change is rolled out, then you end up with the potential of a much stronger experience. We have assembled a range of questions around cybersecurity strategy that can be used to get started on that, to help organizations begin to understand what is it they need to think about? Several papers and articles are available, that we have assembled this information. Obviously, I don't have time to go into it here, but they really are driving thinking about, what is it you need to accomplish? And then, how are you going to do that with the way you run your organization, to put together how all the pieces and parts need to work together as a straw man? And then, that becomes your baseline implementation, and then you-- this is a living document-- then you work to improve it over time, as you learn more. Next slide.

What I'd also like you to think about is, when we talk about these defects, remember that there are design weaknesses, coding weaknesses and implementation weaknesses, and there are different ways to identify them at different levels across the organization. The more emphasis you put on finding and fixing earlier, then we've had experience and documented results that show the lower cost you have in terms of managing and maintaining the product early on. Organizations that really look at it, emphasize early life cycle intervention, do a better job with cybersecurity. And that's where I would urge you to increase your involvement and focus, but you need to plan for that. These are parts of the organization that have not typically had cybersecurity responsibility assigned to then, except for the requirements that were actually fed into a system build. And too frequently, those requirements basically say, produce something that has good cybersecurity. Well, what does good mean for that particular implementation? Those need to be defined, and then how are we going to know if we're producing what we need? Those need to be defined. And then those become ways that you can begin to start to look for, how are we improving and what's working? Next slide.

Here's another look at that continuous improvement that you can implement. There are many steps that can be integrated and can be done at different parts within the organization. It involves training the people to do them. It involves adding those to their work list, in terms of what gets accomplished. And it also involves coordinating the various pieces across the process. You can't just do threat modeling at the beginning, never integrate it as part of your requirements, and expect that the output of the product is going to effectively address those threats. In order to do that, you're going to need misuse, abuse cases, and all the pieces and testing for it and verification, all the way along the line. The integration of those across is key. Next slide. Any other questions, Shane, before I drop into final thoughts?

Shane McGraw: We are all caught up, Carol, so final thoughts.

Carol Woody: Okay, good. Next slide. Essentially, just hitting on the key points here, you need to establish a plan for what is sufficient for the system and the cybersecurity that you're engineering. This is taking an engineering approach to thinking about how you build your pipeline, how you manage your supply chain, how you put the pieces together effectively, so that it can be consistent throughout the whole organization. You have to look beyond just each individual activity, and look at how all the parts and pieces fit together. I think there's lots of guidance and lots of ways to do that, but it involves the organization integrating planning and continuous improvement as part of what they're doing, instead of just bringing in developers and tools and dumping them in together, and saying, "Make good things happen." You'll get some results, but you certainly won't get what you're looking for. And so planning is key. You can get the results you need, but it requires thinking in advance and getting organized. And we're certainly glad to help you, and hopefully, what I've shared with you will be of value. Next slide.

If you're interested in more information, I mentioned a lot of papers. Those are available on our website, but I also have a book that I co-authored, and we have a certificate that can give you a lot of online training, to be helpful. Next slide.

And here's my contact information and pointers to some of that web information I was talking about. And I think at this point, I've covered the points I wanted to share with you. Hopefully I've given you some food for thought and I'm happy to answer any additional questions.

Shane McGraw: Great. So the queue's open, folks, if you have any questions for Carol. We've got about three minutes left, so I'll just do my wrap, and if we get a question, Carol, I'll read it off to you. So first of all, we want to thank you, Carol, for the great discussion today, and sharing your expertise as always. And then lastly, we wanted to thank everyone for attending. We had a great worldwide audience today, so we appreciate you taking the hour to spend with the SEI. Upon exiting, we ask that you hit the like button below, and share the archive if you found value in today's presentation. Also, you can subscribe to the SEI's YouTube channel by clicking on the SEI seal in the lower right corner of the video window. Lastly, we invite you to join-- or to attend our SEI research review, which will be held virtually this year, November 8th

through 10th, and registration information is available on our website now. Any questions from today's event, you can also send to info@SEI.cmu.edu.

So, Carol, that's it. It looks like there's no more questions, so again, thanks everyone for attending today. Have a great day and we look to see you at a future event.