Measuring DevSecOps: The Way Forward

featuring Bill Nichols and Hasan Yasar

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. I am a principal researcher in the SEI's Software Solutions Division. Today, I am joined by my friend and colleague Bill Nichols, a researcher on the Measurement and Analysis team, and by Hasan Yasar, who is technical director of the Continuous Deployment of Capability Directorate in the Software Solutions Division and also a friend and colleague. Today, we are here to discuss DevSecOps metrics. So, DevSecOps itself some of you may have heard about and read about in many of our blog posts. We have lots of them about that. Here we are going to focus specifically on the measurement aspects of that. I want to welcome both Hasan and Bill to the podcast today.

Bill Nichols: Glad to be here.

Hasan Yasar: Thanks for having us, Suzie. It's a delight to talk to you always all the time. Thank you.

Suzanne: So, we are going to have some fun today, but for the viewers that don't know the both of you, I would like you to start by telling us a little bit about yourself and the work that you do here on a day-to-day basis. Bill, why don't we start with you? Tell us, how did you get into this area of research in particular?

Bill: Well, my background is in physics. So, I got into programming because we generated a lot of data and had to do a lot of data analysis for my experiment. And that's how I got into programming. And eventually I ended up at the SEI working with the TSP [Team Software Process] team. The current work is really just kind of a natural extension of looking at software-generated data for improving the process, for measuring effectiveness. I have worked with teams. Now, we are looking at the DevSecOps because there is the promise of generating lots and lots of data automatically as the systems operate.

Suzanne: OK. Hasan, how did you end up working in this area?

Hasan: It's a long question you're asking, Suzie. I'll try to make it short. I joined SEI in 2010. I was a practitioner in the community. My background is doubly in electronics and electronics engineering, but I have been writing software almost 25 years. I carry out my industrial experience on building up very complex systems, maybe like building up simulators and writing accounting software, and healthcare-related software. So, I ended up at SEI. Since I joined SEI I really started to think about what we can do better for the software-development and developer perspective. Since day number one, I have been focusing on how can we get the industry knowledge, make it more engineering principles and concepts, make it a science. Because there is a science behind it. There really is a thinking, there is an engineering concept behind all we do. Think about the end-to-end approach. Since I joined the SEI I have been focusing more on improvement, on delivering speed with the quality, with the right reliability. With all the quality attributes in it.

Then, after we focus on the DevOps specifically, last six, seven years, and I have been teaching DevOps course as well at CMU. The same concept again and the new technology, new toolsets, new mindset, makes that goal achievable. Now we are using *DevSecOps* term, which is adding the security into the software-development process and practices. Even though it is a part of it, even it is part of quality attributes, as a community we often ignore the security as not included at the beginning of the lifecycle. Now, when we look at the journey of the last couple of years, we fixed the setting up of the DevSecOps environment. It is not a problem anymore. It is much easier to build up an end-to-end pipeline, either using various ready tools or using a cloud platform or using various ready solutions. But, one thing is really missing in this pipeline concept or the fast concept. We can go fast. It is great. But we don't want to be creating an application or a product first, that is not meeting the business needs. Second, we are acquiring a lot of technical debt in our work, which we don't have the right measurements. That is the reason that really strikes us, and Bill and I, we have been focusing on how can we really make sure that we are delivering the right product [while] at the same time meeting the needs of the users and also helping us to go faster, at the same time saving time. So, now, we need to measure a lot of things in the pipeline, which we are creating a lot of data, as Bill said. Since we have end-to-end connectivity, all these tools, so we have end-to-end processing practices. It is creating enormous data for us. If you don't know how to use data properly, that begins to fail. That is the reason I ended up this one, and carry out what is the next step after DevSecOps. This is the time. Let's talk about the metrics the next step.

Suzanne: OK.

Suzanne: Bill recently published an SEI blog post, and talking about the current state DevSecOps metrics, and I really do encourage people to go and look at that because I think it

gives a very nice history of where have we come in measuring software in general, not just about...We landed in DevSecOps, but it's a very nice history of the measurement questions that we have been answering. From my perspective, and from Hasan's, it is obvious why we should care about DevSecOps metrics, but Bill, I wanted to hear your perspective for a minute on what is it that you think is important for our viewers to understand about metrics in the DevSecOps environment.

Bill: Well, a lot of our customers like the DoD are making the transition to DevSecOps, and there is a lot of talk about moving faster. But, the basic question is, *How fast does it have to be? What is good enough?* And, *How do we know if our program is actually being effective?* You want to have some concept of, what is your overall program health. *Am I developing a sustainable, effective, efficient system?* That is important to all of us whether you are in business or in government, and you actually want to know that you're delivering value. So, are you doing things that will consistently deliver value? We want to look at the metrics. What I try to express in the blog post is that there are a lot of metrics that are being used, but you have to think about them in terms of the right questions. What are you trying to answer? What do you want to know? And, if you don't understand what your fundamental questions are, all that data can really just lead you down the rabbit hole. I mean, *Should I be working faster? Well, how is that going to affect my quality or other productivity aspects? Am I working faster on the right things? How fast does it have to be? I mean if it doesn't have to work, I can do it really fast.*

Suzanne: Most of our systems are going to require that they work [laughs].

Hasan: I am just going to add in one thing Bill said, Suzie, if you give me permission.

Suzanne: Sure.

Hasan: Bill started a good question about the *why*. Why we need to have data. That reminds me of a good statement from Lord Kelvin. He is a physicist. What he said that really struck me he said, *When you can measure what you are speaking about and can express it in numbers, you know something about it. But, when you cannot measure it, when you cannot express in numbers, your knowledge is a marriage of the unsatisfactory kind.* It is really striking because if you don't know our system, then how can you say that we are doing well? To show what *well* means, we need to measure it. If you are looking for development practices, if you are looking for some cycle, anything else, it starts from *why*. We have to understand the system. That is really a good statement from Lord Kelvin really describing why we have to do it. Because we have to know our system. To know our system, we need the numbers.

Suzanne: That is—as Bill was saying earlier and you, yourself, Hasan—with this DevSecOps pipeline…So, one of the things that is a characteristic about this pipeline is that we can run

many, many tests. We can run many, many cycles through it in an automated fashion, which makes it much faster. We go back to the punch card that was four hours to compile, and then it was desktop, and now we have got this pipeline. So, it is very, very fast, which means we get lots and lots of data. I take Bill's point about knowing what data is important has to do with what questions you are trying to answer. But, the other thing about that is how do you understand which data you need to trust and which data you need to question? That is one of the things that for me is underneath a lot of the churn about DevSecOps metrics. I am aware of one program I work with. Not just DevSecOps, but they have 150 measures defined related to software in their software development plan. I looked at this list, and it was literally a list of 150 long without categorization, and I just kind of went nuts because there was no sense of, *Why do I need this, and what are the limitations*? *What data do I need, and what data needs to be trustable for me to be able to trust the measures that come out of it?* Talk a little bit about how do we go from this pile or lake full of data to knowing that we have data we can trust and knowing how we can use it?

Bill: Well, now you are talking about the entire data chain. It always starts with, *What are your goals?* And when I say *you*, you have your own personality, your own role, your own responsibilities in the organization. If I am a developer, I have a different set of needs than my manager has. If I am the project or program manager, I have an entirely different set of needs. It is foolish to think that the same data is going to be as useful to one as to the other. So, the first thing you have to do is really understand, *Who is my user? Who is the consumer of this information? What decisions must be made?* And, *What do I need to know to answer those questions?* Now it is not just a matter of trying to find relevant data. It is actually searching through the system to see, *What are the sources of information? What are the indicators that I could use that will answer this question? With these indicators, what can I actually measure?* Now you have got other questions about, *What is the reliability, the repeatability, of those measures, how do they correlate with other measures?* That is, *What are the factors that might otherwise influence this?* So, you have to understand that entire toolchain.

One of the beauties of DevSecOps is that you have instrumentation at so many points along the development chain that you can actually put some of these pieces together. You can look at things in isolation, or you can look at the entire gestalt, the production as a whole, and start to understand how these pieces come together. But, it always starts with what kind of decisions do I need to make, and what could be important. Then you look for the appropriate data to inform that decision. And, I share your frustration. I see a lot of laundry lists of metrics, and it is like, well, *What was this metric? Who needs this metric? What is the context? What are the conditions under which this needs to be taken?* If you haven't defined those, you are really just generating a list of write-only data.

Suzanne: I am not going to quote Lord Kelvin, but this is a...I don't even know who originally said this, but I use this a lot of times: Just because you can doesn't mean you should, right? That is what I a seeing in a lot of the measurements, DevSecOps and otherwise, is, *Oh, I can measure this item. So, here's a graph that shows what that looks like. And I can measure this.* It is disconnecting the data from the purpose. You can tell I have a little passion on this subject.

Hasan: When I get that question from my students, I am getting similar things. I always say, *What matters to you?* If we are measuring something, it is really what matters to a developer because the developer has different needs, the developer, versus the program manager's different needs. Their interests are different, but what about the business case? If I am a developer, I would like to see my defect ratio. I would like to see my success build.

Maybe my manager is looking for my productivity based on how much I produce in my code repository. It really it depends on a consumer, it depends on the users of the datasets. That goes back to the, *Why I did that*. Let's find out that answer. Because it's generating data.

But, the one thing I have to open up, SuZ, you said at the beginning, like the trust of the data. I feel more confident than ever before because now we are generating it automatically from the DevSecOps environment. We can come up in details and discuss that. We are not using manual data creation anymore. In the past like we were...I'm not saying making up, but sometimes we are measuring the data based on what enters it manually, which is a human. If a human enters the data, that is error prone in terms of quality. We may see some issues with the quality of data. Now data is being generated automatically from the DevSecOps environment. That is the beauty of an end-to-end pipeline. So, I am not worrying about the truth of the data, but I am worrying about misuse of data for something else. If you misuse data, we will have a wrong decision for any criteria that we would like to do. So, the source of material is corrected, but misusing of data, it is going to get more. So, what are we using to decide something?

Suzanne: But I do want to caution that we still have humans involved. Right? The example that comes up in my mind a lot is, *I am a developer working on code, and I make the choice of whether I am going to commit that code to the repository as soon as I finish that segment or if I am going to go off and do something else, and then I am going to commit that code later*. The code hasn't changed. Now what I did do is I affected the cycle time of how long that code was in development versus when it went in to integration and commitment. That is a very minor thing, but we still have to be aware I think that it is not a complete...Until some robot is actually writing the code and submitting it, we still have humans involved in this process. So, it is not a completely automated process as we might like to think about it.

Hasan: I think maybe I should open up a little bit. When I say automated, Suzie, I am talking about generating of data is automatic. Yes, a human is part of creating some actions. As you

describe, actions are…Committing is an example. Actions can be creating an estimate, so another human is in the loop. So, a human is in the loop, but the generation of data is done automatically from the system. When I say the automation, it is basically the data generation is automatically done. Like I can get the number of…

Suzanne: The human doesn't have to record the time that they put the commitment in.

Hasan: Exactly.

Suzanne: The system automatically generates that.

Hasan: Exactly. The system does it automatically for us. We will discuss again probably. The data that we are collecting, the system is generating for us. We can't really find out what is the root cause of analysis if the cycle time is not reflecting the truth. Maybe we can look at, *Yes, developer not checking the code. Maybe there is something else in between messing up the numbers generated from the systems. That is the reason we can go find out, and we can trace out some actions we have to do*. I am worrying about whether data generation is more true, but how we are entering the system, yes, that is right, the human is involved. Human enters the estimate. Human writes the code. Human builds up the stuff. Human can do, that varies, but generation of data is automatically done. That is the portion I liked.

Suzanne: OK. Fair enough. Bill, did you want to comment on this?

Bill: Yes. One of the things we found back with many years of taking TSP data is there were some types of data that we could very reliably get. Oddly enough, getting programmer time was actually pretty easy. But, it is the things that they didn't do all the time or routinely where it got progressively less consistent. For example, we had maybe one-third of the teams that were accurately recording size data. Well, that should in principle be one of the easiest things to get because you can write a simple script to query your source repository and get very accurate accounts of size. But, if that process isn't automated, it doesn't happen, and if people have to do it by hand, well sometimes it happens, sometimes it happens with errors. Being human, every time we do something we have got a chance of introducing our human defects. So, the automation is a chance for consistency, repeatability. It provides a lot of opportunities. I don't know that we necessarily want to get away from humans recording some of their own data. That is yet to be determined. But we do know that there are lots of things that we should be able to get automatically that we haven't been getting in the past.

Suzanne: So, that was one limitation of DevSecOps metrics is that where humans are in the loop, human action may change the actual source of truth, if you will, in the data. What are some other limitations that people should be aware of? Because I just want to make sure that…This is a very burgeoning area, and as Bill and I've spoken about, lots of different measures. But, what

are some things people should be aware of and cautious about in terms of the DevSecOps metrics that are currently in use commonly or that they may be seeing coming up?

Bill: One of the things that you definitely don't get is, when things are recorded automatically, you have this same kind of labeling problem that we often see in machine learning. *Are you doing supervised or unsupervised?* And, when we had humans gathering data, they had a lot of contextual information that could give you a very rich dataset that would include some of the context. It was very easy for them to look at a defect and do some things like categorization, ranking severity. We don't have good mechanisms in place to do those sorts of things automatically. The things that come out most naturally are some of the more obvious ones like time stamps. It's very easy to measure flow rates. Unfortunately, I think what we've seen in a lot of the DevSecOps measures so far is that's the easy thing to get, and that's primarily what they've been focused on without asking the deeper questions about what are your real information needs. So I think part of our challenge moving forward is being able to dig a little deeper and trying to uncover some of that contextual information and take more advantage of what the automated tools can offer.

Suzanne: Hasan?

Hasan: Yes. So, Suzie, I have been seeing so far, as Bill said a couple of them already, but I would like to open up a little bit. Current usage of the metrics is more about the tool-generated data. I saw many examples. When you look at the code complexity, or you look at the specific code commit as an example, the consumers are looking for that tool generated data specifically. So, relying on a tool dashboard, it may not have the correct metrics for them. Like looking for a number of fields on a Jenkins server. If it is not tied to the business goals, it could be great to have just specifically the number of failures, but sometimes metrics are the correlation of multiple datasets. By looking for just a single tool, it is limiting our metrics. It is limiting our looking for an end-to-end lifecycle perspective. What I see right now, most of the implementers or the users are limiting what the tools are generating in the software pipeline. Maybe there is an issue tracking system and the issue tracking system has a number of cases that we can generate. If you are looking for the specific repository, maybe you have a version, a login on it, each of them actually generates from the tool as an output. By looking at metrics, it is limited to what tools offer. We have to look at overall and tying back to the *why* question at the beginning. We may have to pick the data from various tools and combine those datasets and present what we are trying to measure. That's the portion of the…

Suzanne: The data-visualization aspect.

Hasan: Data visualization, but aggregating data is also missing. If you are going to calculate some metrics, it is requiring multiple sources to aggregate those and calculate that. By

calculating just a number of cases, it is limited. I am thinking of one example. If I am looking for my cycle time, if my cycle time I will measure based on the number of cases and then case open and close, it is not reflecting the true picture. I have to look at the deployment of that feature that goes into production. That is going to give me a true metric. It is not just the ones that were generated.

Suzanne: One of the things that we have talked about in the measurement community is the atomic measures, which is what you are talking about, the actual data that I can gather from a source, in this case, the tool. But the measure itself is usually aggregated from and composed of multiple atomic measures. What I am hearing you say—and I heard Bill say it a little differently—is we need to be careful not to limit the questions to what the tool can provide. We need to ask the questions that are the real business operational questions. Find the data wherever it is, and compose it into the measures we need rather than letting the tool drive, *Well, I can give you this. So, this is what you must need to have. This will answer this question. So, that must be the right question because I can give you this.* Is that a fair statement?

Hasan: Yes. That is fair. That is right. We don't want to limit ourselves to just the tool only. We have to think about beyond the tooling and look at what matters to us and aggregate the datasets. I ask that question many times. Always we are getting response from such a tool dashboard. That is not enough to measure the data from just one tool dashboard. Technically if we focus on the tools, then the tool is dictating our metrics. No. Metrics or measurement or business, they should dictate how we are aggregating data. We don't want to go dictating based on tool because another thing I have to open up as well, in the toolset we measure it is basically more about some code repositories. If you look at it, it is going to be a little bit technical, but if you look at just the static-analysis perspective, we might be analyzing with just the code base only, which is the only single repository. But, if you look at the project, project has multiple code repositories in it. We have multiple components in it. So, by looking at one element, and saying that we are doing well in terms of bettering the defect ratios for a given repository, it is not enough. We need to aggregate multiple projects, combine together...say, *Here is our metrics on a system or project that we are building*. But, the current metrics about the project [are] just tool-centric.

Suzanne: Well, and to take it a little further, doing static analysis of a code base does not tell me how it integrates with hardware components. It doesn't tell me how it integrates with other things it interfaces with. Again, looking at what is the question becomes the question, as Bill would say. Bill, would you like to comment?

Bill: Yes. Let me offer another real-world complication. And that is that not all the pipelines are going to use exactly the same tools. They may use different subsets. So, one of the things we have to be conscious of is different pipelines are not going to be producing exactly the same data. That is one of the things we are going to have to address. As you try to aggregate things at the

program level, it is going to be one of the big questions coming up. How do you deal with the heterogeneous pipeline situation?

Suzanne: Got you. Based on this discussion so far, what is your vision for the future of DevSecOps metrics based on what you are seeing in terms of the ability to add in things like machine learning, about how we are training people differently, things like that that all gives us a forward-looking perspective. I will let Bill start, and then go to Hasan.

Bill: Well, I think fundamentally the vision is that we are going to have a lot of data to work with. It is going to be collected automatically. We are going to be able to use this data to reason about how we do our work. Not only the program health, but how we are actually working individually and how the product or the project is transforming into a product. That should be there in ways that we can reason about it based on real numbers. That is a future state because right now a lot of this is done with a wet finger in the air and a lot of subjective feel. I think we could make this much more quantitative, and that is not really a bad thing. This quantitative capability is how all pro athletes work. If you have ever seen someone like a golfer, they are recording essentially every stroke on the course. It is not just counting the strokes. I am talking about what kind of club they used, how far it went, what were the conditions. If you really care about being a world-class performer, that is the kind of thing you have to do. But, it is a really hard thing to do, and I think the quantitative stuff, the measurement as a Fitbit that you carry around with you, that kind of capability to standardize measurement, make that available, is something that could be an end state. Data is to a large extent just there for you. People know how to collect it. They know what the metrics should be. You can go and visualize those metrics in a way that is meaningful for you.

Suzanne: OK. Hasan?

Hasan: I think a few things Bill said. Yes. We need to get more aligned for the consumer perspective. Also, what I am envisioning is building up great visualization capabilities. Then we can get a bunch of sources of input into the pot and then let the consumer use the data, which is a true data, first of all, it comes from the machine. At the same time, it will help us to do a better decisions, which is our business case, in an environment so we can consume in a better way.

Also, all data should be actionable for us. We don't want to be creating data just for the sake of data. We would like to create some actionable result for the data that we are collecting. We are looking for what type of actions can be done. If I am measuring the deployment lead time or delivery frequencies or the cycle time, what actions do I have to take and go back and change some processes and practices in that environment? It is more about actionable and more about visualization of data, and we can use various platforms.

Other things we have been looking for and working with Bill together, is there any way we can abstract that model, getting away from higher level, and tie into the key software metrics, so we can apply into DevSecOps environment? Yes, we are collecting data from the tools, but how can we aggregate those and make the model, so we can use various tools?

It doesn't matter what type of tools we are using, but data is the data. Let's have a common element for what type of data we need to measure certain datasets or certain monitoring components, so we can really collect various components of the DevSecOps environment. So, setting up what the DevSecOps framework it is and describing how to collect it, and then also make an actionable result of the dashboard. Because it is really important to build up a common dashboard or some dashboard that is going to help the other users, such as developers and lead managers or the security persons. It depends on who are the consumers to define that actionable metric.

Suzanne: I will add one thing to that. I want the equivalent of a beer game for DevSecOps metrics going all the way up to the business. I sort of heard it in what you said a little bit, Hasan, but to me the vision of DevSecOps metrics is becoming an essential element of running a software business. That means not just the models, but the connections to the business elements are in there too. I am going to add my own little vision piece to that.

Hasan: Thank you so much. You open up a great topic. I always teach in my software and security class, the reason that we are not able to get security in active and early is because we are not able to tie it to the business. The metrics should give us a reason to tie it to the business. Usually if the business doesn't care, nobody's going to put time and effort for the security. If you show the value for the business, if you show any breach in the data, [how] it would affect the business, what risk is associated, now the managers will say, *Let's get it done because I don't want to be hurting my reputation. I don't want to hurt some legalities. Let us get it done.* So, tying into the business case, and tie into the specific risk. It is going to solve a lot of problems. And other things like manager perspective, SuZ, usually the managers they will like to say, *What is impact to me. If I don't do this, what is impact?* The best way to show impact is with the data. With the numbers. If you don't do it, here is the impact. If you spend X number of days, you will get better, but here is the impact for you, which is exactly for business type. Glad that you brought it up.

Suzanne: Any final comments on that, Bill?

Bill: I like to look at some of the analogies we have seen in pro sports analytics because that has kind of exploded over the last 20 years. As they have looked more carefully at some of the underlying data, most of the insights they gather just reinforce what we already know because we already knew Babe Ruth was one of the greatest hitters ever. OK, but you start looking more

deeply and you start realizing, *Oh, these basketball players who are leading their teams in points are really hurting the team. Why is that? Oh. They don't have great shooting percentages, and numbers of shots don't really matter much for basketball because every possession ends in a shot almost*. As you gather more information, you understand the system. You start to understand how the little things actually contribute to the actual outcomes, and that gives you a lot of capability to affect those outcomes early in the work. That is where a lot of the value added comes from. As software becomes more and more of a commodity, that kind of competition I think is going to come to the fore, and the organizations that are better at that are the ones who are going to succeed.

Suzanne: Excellent point. Excellent point. I want to finish up by asking a little bit for our viewers, if they are interested in this topic, how do they get started? It is like, *Oh, we have a DevSecOps pipeline, and nobody likes how our measurements are being used now. How do we get started in doing better at that, and what resources are available from the SEI* (of course from the SEI) *that will help me with that?* So, tell us a little bit about resources that you recommend, and where you expect people to get started. We'll start with Hasan this time.

Hasan: I would advise our listeners to look at SEI website and podcasts and webinars, so they can get a lot of information understanding the commonalities, understanding the concept or principles of DevSecOps and DevOps. And understand the metrics. We have a lot of references of the software metrics. They can utilize it. Also, I advise to the team they have to understand what tools they were using, what type of data they are generating, and they should think about what matters to me. You start from that, and I will advise them to ask, *What wakes me up in the night? What matters to me?* I am sure they are going to find an answer, and then when they need help they will look at how can I collect what metrics methodologies are. I want everyone to go to the SEI website and take a look at the DevSecOps space, and Agile space. A lot of write ups and content over there, SuZ.

Suzanne: Bill, what are some of your favorite pieces of advice?

Bill: Well, one of them would be think about your own work practices very carefully. Just be mindful of them. There is a lot of work going on right now in coming to terms with the kinds of metrics we can get out of DevSecOps, so I wouldn't tell you that we have something today that is going to necessarily solve your needs, but the fundamentals of being able to understand your work practices, understand what you are trying to do overall and in each of these steps, is where you start. Once you understand what your overall objectives are, you can think about, *What do I need to do to get there? What steps do I have to do individually? Now how can I actually measure these steps? What is an efficient step? What is an effective step?*

Suzanne: I am so surprised, Bill, you didn't mention our work. We have a history of work in goal-question-indicator metrics (GQIM). And so that's the thing I would add into this is if you don't know how to get from the data to the *why*, the SEI has got tons of stuff on using goals, questions, indicators, measures, as a framework. And, there lots of stuff: blogs, technical notes and things like that. And I would point people to that as well. Even though it's pre-DevSecOps in terms of when it was generated historically, that's exactly what we're talking about today is using that method of thinking about things. Even though we may not be talking about tools, the thinking framework I think is still good for today. Would you disagree? We can have that conversation.

Bill: Yes, and if you want to look at the GQIM, that material, the actual course material, is posted under the Creative Commons on the SEI website. Also, that is a great place to go and start and think about using the indicator templates to record all the information as you go.

Suzanne: There you go. All right. I want to thank both of you for joining me today. This is a topic, as you can tell, that I enjoy, and enjoy thinking about and helping people with. I want to thank both of you for your work in this area. We will probably have more things to talk about in the future as this area evolves, and so I look forward to that.

I do want to tell our viewers that we will include links in the transcripts to some of the resources we mentioned, the Creative Commons and some of the blog posts related to both DevSecOps and measurement. I want to thank all of our viewers for joining us. Please do not hesitate to send questions to info@sei.cmu.edu on this topic, and we will do our best to answer those. Thank you.

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 do not hesitate to email us at info@sei.cmu.edu. Thank you.

~~**Jonathan:** Great. Well, Carol, thanks so much for joining us today. To our listeners, we will, of course, include links in our transcript to all of the resources that we've mentioned in this podcast. Thank you again for joining us.~~

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.