Selecting Metrics for Software Assurance

featuring Carol Woody as Interviewed by Suzanne Miller

Welcome to the SEI Podcast Series, a production of the 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: Hello, my name is Suzanne Miller. I am a principal researcher here at the SEI in the Lifecycle Innovation and Automation Directorate. I am here today with Dr. Carol Woody, long-time friend and colleague, who is a principal researcher in the SEI's CERT Division. Today, we are here in this podcast to talk about some of her work in putting two things together that people don't always think about together: software assurance and measurement.

I want to invite us to think about those things a little bit more broadly than we might have in the past. Before we do that, I know there's some people here that probably haven't listened to you before. Give us a little bit of insight into what is it that you do here at the SEI, and why do you think it's important?

Carol Woody: I manage a team that is focused on cybersecurity engineering. What we are looking at is how to better integrate the things that will allow systems to be more operationally secure through the development lifecycle, so that you can have the product at the end that will be effective and give you the right results you're looking for.

Suzanne: You are trying to avoid what we call bolt-on security.

Carol: Exactly, it is building it in.

Suzanne: Yes, build it in from the start.

Carol: The development cycle doesn't own all of the security issues. It is a combination of the context as well as the construction. What we are really focusing on is making sure that the construction pieces are well addressed.

Suzanne: We have got assurance. Software assurance in itself is sometimes a term that is not well understood. We have information assurance. We have software assurance. We have mission assurance. We have all kinds of assurance. Tell us just a little bit about what is the context of assurance that we are talking about when we are talking about this topic on cybersecurity engineering along with how measurement goes with it.

Carol: It is a combination of things. Software assurance itself talks about a focus on functioning as intended. So, it is built to do what you want it to do. It also is looking at the perspective of vulnerabilities and basically being free from vulnerabilities when it is operationalized.

Cybersecurity is really looking at the operational security aspects to make sure that something can be protected: your system and your software. The vulnerability problem and the protection issues all play together. Ultimately the fewer vulnerabilities, the much easier it is to protect it.

Suzanne: One of the ways I've heard it talked about is software assurance is about making sure the software does what it's supposed to but doesn't do any of the things that it's not supposed to. That is really kind of the scope of assurance that we're talking about.

Carol: Exactly. It's a combination of those things.

Suzanne: And the vulnerabilities play into not letting it do what it's not supposed to do.

Carol: Many people say well, *Why are you just focusing on the software?* But, essentially, at this point in time, software is handling almost all of the functionality. If we don't focus on the software, then we are leaving ourselves wide open for lots of issues.

Suzanne: What are some of the typical challenges in assuring software? Then we can get into, *How does measurement help us to address that?*

Carol: First of all, when you are looking at it, you are looking at establishing functions as intended. Well, that is defined in some way for some product. First, you have to look at, *How do you know that the specifications were sufficient to do the right thing*? Then, you look at the product as built and determine, *Did it meet those specifications sufficiently?*

In many cases, we have real trouble identifying what are good security requirements. What are good context needs? In some cases, software is perfectly good in one context and a disaster in another. So there's a tremendous variability here depending on what you are building and how you intend to use it.

Suzanne: There is also the fact that the threat changes. Even if at point A, everything works as intended in the specification, it's a good specification. Between point A and the software being

built, there can also be the possibility that the threats change that actually change the security requirements.

Carol: In essence it is the ability of the attackers to take advantage of the way we are building systems and the limitations and the choices we make in the trade-offs that essentially can constitute greater threat. Right now they have a certain level of capability, but their capabilities are growing just as our abilities to build software are growing.

Suzanne: It's a little bit of a race.

Carol: That is part of the whole challenge.

Suzanne: Where does measurement play into this? How do we figure out what is the right measurement data that helps us to understand whether we are actually moving towards assured software?

Carol: There is one more aspect of the software though that plays into the measurement challenge. That is that we talk about software assurance being free from vulnerabilities. In reality, no software is perfect. You always have defects, even in the highest quality software. Our research is showing that a certain percentage of those, usually up to five percent, are vulnerabilities. That means that no matter what the software is, it is not free from vulnerabilities, it is reduced vulnerabilities. So, you are dealing with, *How many can you tolerate?* It is a risk-tolerance question as opposed to a removal issue.

Suzanne: Risk tolerance has both a frequency—*How many are there?*—but also an impact. So how do we gauge…

Carol: That ties back to the context of what you are trying to do with this software.

Suzanne: The impact part relates more to, *We have contexts that are more prone to attack than others*, right? The threat picture is different.

Carol: Or the result of an attack could be much greater.

Suzanne: OK, so measurement. We need to measure, *Are we getting there?* We have to measure defects, right? Because just as you said, we need to understand what we do. What else do we need to measure, and how do we connect measurement that we would actually do as a normal part of our lifecycle to being able to aid us in understanding the assurance question?

Carol: Well, let me describe to you the process we have taken because you can plaster measurements out all over the place. There are all kinds of things that you could measure. There are all kinds of things you could do. The real question is what is going to gain you better

operational results. In some cases, that is subjective, but what we have built with previous research is a framework of good software assurance practices. Those relate to process management, project management, your engineering, your sustainment and operation activities, configuration, management control, and some of those support functions that you are dealing with in the development lifecycle.

Given those practices, then we can start to look at how those can be measured in terms of effectiveness and identify appropriate measures. But again, we need to figure out, *What is it we want to measure? Why do we want to do that*? There are really certain questions related to software assurance that would be tied to each one of those practices. Then the measurements would help you answer those questions to determine what is good enough for your certain environment.

Suzanne: So, that is another context-dependent kind of activity because in a cyber-physical system that is going to be out in a hostile environment, that is a very different context than a payroll system.

Carol: Plus, you are also dealing with, in the cyber-physical side, the relationships between what it's doing operationally in terms of generating power or things along those lines and then how that relates back to the software itself, which is more conceptual in terms of how it is functioning. Whereas most of your payroll is all ones and zeroes, so you're not dealing with quite as much.

Suzanne: It's transaction oriented, but on the other hand, people really care about their paychecks being right, on time. I mean, we have a bunch of quality attributes.

Carol: Or the right ones and zeroes being sent to the bank properly to influence the ones and zeroes on the account. So, we have a lot of interesting interfaces there.

Suzanne: Yes, but that difference has got to be accounted for. That is why being able to take that, *Here are the practices. Here are the questions that relate to assurance related to those, and then here are measures that can be used as a way for people to rationalize that.*

Carol: That is most of it. We are also looking at one more step. That is in most of the software development people are trying to do something that is effective and produce good quality. They are already doing a lot of activities. Now, how much of that contributes to the *functions as intended* and *free from vulnerabilities* is something that they need to clarify, so that they can determine what kind of software assurance levels they are producing.

What we are seeing by having generated from the practices the questions and the metrics is these can then be mapped against what an organization is already doing to identify, *How many of these*

questions can we already answer based on metrics we are already collecting, just repositioning them or restructuring them to map to those questions?

Suzanne: It costs money to collect measures.

Carol: Time and money is always an issue.

Suzanne: That is something we can never forget. That this is part of the point of making measures particular to the context, because if I'm doing payroll, if I'm collecting measures that are way more applicable to a cyber-physical system, I'm wasting money. We don't want to go either way. We don't want to do too much measurement that doesn't apply to the context, but we want enough measurement, so that we build confidence that where we are going is going to meet functionality and is going to reduce vulnerabilities.

Carol: Plus, by mapping what would be good measurement practices, essentially tied to the questions that you should be asking, you can determine, *Which questions haven't I been asking?* and identify the areas where you've got weaknesses in your current processes to address.

Suzanne: And should I be asking those questions? *Are they relevant*? *Then, if they are relevant?*

Carol: *Then, how can I get good information*?

Suzanne: Right, *How do I find out about them?* As you say, many times that is not going to be just a change in measurement, it is going to indicate a change in practice.

Carol: [For a] change in practice, you have to determine what you need to measure it within, where you need to measure it within the process so that you are going to get good results. Then, how are you going to interpret that information.

Suzanne: So if I don't have a chance to talk to Dr. Woody about this stuff, where would I go to read about it?

Carol: Certainly the tech note that we are talking about gives them a lot of background. In addition, if they feel that that is not sufficient, there are a good number of references in the tech note that can indicate other areas of specific analysis, like the Software Assurance Framework (SAF), and more in-depth in the practices.

For broader, general background, I have got a book that I co-authored with Nancy Mead, who is one of the SEI Fellows, on cybersecurity engineering. That can give you more of a sense of the total background and issues and principles and guidance in terms of addressing the whole area of software assurance.

Suzanne: There is quite a bit. My point is, there is actually quite a bit of information that the SEI has produced on this topic. This is not a brand new topic for us. It is one that continues to be very important. The TN you are referring to is *Exploring the Use of Metrics for Software Assurance*, if people haven't already seen that. I would also say I know you have blogs and other kinds of smaller communications as well if people aren't really ready to go after the whole book. Smaller bites.

Carol: There is also a training certificate that they can take in terms of cybersecurity engineering and software assurance. That will walk them through a lot of exercises and give them a chance to really explore the issues and what can be done.

Suzanne: This is work that is kind of in the stage of transitioning to the world. A lot of what we are talking about are ways for people to access that. That leaves you open to do new things. What are some of the new things that you are thinking about as either follow-on work to this or other research in this area to help us improve software assurance that you are thinking about?

Carol: One of the key areas that we are looking at is how do you influence the supply chain, because we have been talking about this measurement directly to a specific development lifecycle. But, what if you are contracting for that? Then, you have got a much more complex way of determining how do you communicate to that vendor what you want and then how do you determine that they have built it to meet your needs?

Suzanne: Our supply chains in many of the systems that we built are not just a single tier. It is not just one contractor. That contractor has contractors and maybe that contractor buys parts, some of which may be software parts. This can become complicated really fast.

Carol: Also, you have to recognize that in many cases the pieces that are being folded into the final product were not actually built for the purpose of creating that functionality or performing that activity. When you talk about, *Does it do what you want it to do and doesn't do what you don't want it to do*, there is a lot of gray area there.

Suzanne: To be fair, all the vendors of the world that are doing things in commercial spaces, to the extent possible, you want to take advantage of those commercial products. It is not anybody's fault that they aren't actually following all the secure practices. I mean it is probably better that they do, but that is not always a tradeoff that they are really thinking about when they are building something for a commercial space.

Carol: Again, it ties to the context. They are not usually not building it for your context. So, it is the difference between a generically built product and then the context you are actually implementing it in. The organization has to basically make that leap itself and determine whether the risks are sufficiently addressed.

Suzanne: I don't know if you have addressed this in your work, it is something that just came into my mind because we were talking about it in another conversation. Have the open source community started to apply any of the assurance framework and measurement practices to help people? Because that is one of the places that we are starting to see people want to use more open source software. That could be a source of vulnerabilities if that community isn't paying attention to these kinds of practices.

Carol: We don't see them using formal structures and the decision-making process as efficiently as they could. Because it tends to be individual developers that are contributing smaller pieces. They are doing their small pieces real well, but there is nothing that guarantees the composition. I think there is a lot of area there to improve.

Suzanne: OK. All right. Thank you.

Carol: My pleasure.

Suzanne: This is a very broad topic, and we have just really touched the tip of this iceberg. I do encourage all of our viewers to take a look at the blog posts, take a look at the tech note, Carol is one of the clearest writers that I know in the SEI, so you will get a lot out of looking at things from the perspective of measurement and assurance. We don't always think of them together, but it's time we really do. Not just in the government, everybody, open source, all of us need to think about it.

I do want to thank you for joining us today and talking about this. I know we'll have other opportunities to talk about other work you do because you're very busy. And I look forward to that. For those of you that are viewing, we talked about a blog post. We have talked about the technical note. If you want to find our blog posts, make sure you go to insights.sei.cmu.edu. The easiest way to find Carol's stuff is to search on her last name, W-O-O-D-Y.

Thank you very much for viewing today. I hope that you can create software that does what it does, is supposed to do, and doesn't do what it's not supposed to do. Thanks.

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.