



Agile Software Teams and How They Engage with Systems Engineering on DoD Acquisition Programs

featuring Eileen Wrubel and Suzanne Miller

Suzanne Miller: Welcome to 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 and operated by Carnegie Mellon University. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts.

My name is [Suzanne Miller](#). I am a principal researcher here at the SEI. And I am pleased to introduce you to myself and my colleague Eileen Wrubel. Today, we are going to be talking about our recently published research on [Agile Software Teams and How They Engage with Systems Engineering and DOD Acquisition Programs](#). First, a little bit about myself and Eileen.

Eileen supports the development and implementation of software-related guidance to improve the acquisition, management, and sustainment of the software component in Air Force systems, which is a lot of systems and a lot of software, if I may say. Her research efforts focus on Agile and acquisition with a special interest in sustainment and contracting issues.

My research focuses on synthesizing effective technology transition and management practices from research in industry into effective techniques use of Agile and lean methods in regulated settings. Although I do work with the Air Force, I have a little bit broader scope in terms of the different kinds of projects that I work with in government. So, Eileen, let's talk about this paper today. Welcome.

Eileen Wrubel: Thank you.

Suzanne Miller: I am so glad to talk about this because we have this thing at the SEI of software is carved in cement on the building, and yet we have so many interactions with systems engineering. We have had other podcasts where we have talked about the importance of systems engineering to the software community. This report takes that in a little different direction of looking at how systems engineering interacts with a newer set of practices, Agile methodologies that [we have also had podcasts on](#). This brings together a lot of different things.



SEI Podcast Series

One of the things that brought me into this is some experience I have had in the past with systems engineering where I really noted that systems engineering tends to fall into two kinds of activities. And, we talk about this in [the paper](#). There is the product side: the product evolution, the transformation of artifacts from requirements through design, integration, and test is a big part of systems engineering. But, there is this other part of systems engineering that when it is done well is really about communication and coordination and collaboration. And, *Oh, look, those are all things that are very important in the Agile community and Agile software development*. Right, Eileen?

Eileen: Absolutely.

Suzanne: This makes systems engineering and Agile a really interesting thing to talk about. We have three ways that we have found in this research, through our interviews, that systems engineering interacts with Agile teams. Eileen, why don't you go through what those three ways are that we have found.

Eileen: At a high level, these three models, we call them engagement models, were (1) Agile software engineering teams interacting with traditional systems engineering teams operating on that traditional [systems engineering V model](#); (2) systems engineers acting as Agile team members; and, finally, (3) system engineering teams actually applying Agile methods to their own work and systems engineering functions.

Suzanne: OK. So, let's talk about the first of these models where traditional systems engineering is being used, and that team is interacting with an Agile software team without being a member of that team.

Eileen: OK. So, what we saw is basically, at the boundary between the system engineering function and software functions, Agile software engineering teams were providing a deliverable or a delivery to the systems engineering function. Be that a delivery of code, or documentation and code, as well as work products to facilitate technical reviews. The Agile team engaged in their Agile practices up to that point, put together whatever they needed to put together to hand off to a systems engineering function, and then passed those things over those boundaries. So, the Agile team was free to operate in their iterative and Agile way up until they handed everything over. Then, it went into the systems engineering domain. And, the systems engineering teams executed according to their plans and processes, typically, with that traditional V model.

Suzanne: The impact that that had was that some of the things you would normally do in an Agile software project—where you are choosing high business value, high technical infrastructure value—you could not always make those decisions because you actually were

SEI Podcast Series

bound by the way the systems engineering function had allocated the requirements and had defined the work packages. So, those are teams where we did not always see all the benefit that we can see in what I would call a pure Agile space because they had to deal with this interaction.

So, let us talk about the second role that systems engineering can play, that of the product owner on a software Agile team. The product owner being the role that typically will be in charge of the requirements and their prioritization.

Eileen: So, a systems engineer would typically operate in a product owner-like capacity. They would be involved in the prioritization of features and functions going into that particular [sprint](#). This enables them to follow the testing of the features and functionalities, so that when work products come to that boundary between the software team and the systems engineering function, systems engineers know what is coming into test and evaluation. And, they have prepared on their end. There is a smooth transition as the work product flows from one part of the engineering process to the other.

Suzanne: We did see smoother transition. That was one of the things we saw in that case. We also saw some opportunity for making changes in the systems engineering ideas and designs because of some of the early learning, because of the more detailed involvement with software. Some of the software implementation can change some of your designs. So, software actually had a little more influence on the systems engineering products when they worked in this way. I thought that was a good result out of that.

Eileen: Yes, it absolutely was. Instead of throwing things over the wall, they were opening a door in between the two functions.

Suzanne: Right. Now that door gets blown wide open in the third model, which is where systems engineering actually adopts Agile principles and methods in doing their work. What this means is that they are going to an iterative approach to developing the requirements, developing the baselines, establishing the baselines, establishing the designs. I mean all the way through.

This is a newer set of practices. We are starting to see more of this, but this is where that product and service dichotomy of systems engineering comes in to play because systems engineering functions that have a recognition of the service side, are a little bit more prepared because they don't see that the only thing they have to do is the product artifact transformation.

We did study one large project that did adopt these kinds of practices. It is still very new to see systems engineering organizations go in this direction, but [INCOSE \[The International Council on Systems Engineering\]](#) has got a working group on Agile and systems engineering, The International Council on Systems Engineering. [The National Defense Industry Association](#), their systems engineering committee has an Agile and systems engineering working group going. So,



SEI Podcast Series

we are seeing a lot of interest in this arena from the professional societies that systems engineers typically interact with.

The other thing that is more recently happening is when you start looking at scaling frameworks for Agile, one of the dominant scaling frameworks used in the Department of Defense is one called [SAFe, the Scaled Agile Framework](#). The people behind that are actually starting to build what they are calling a continuous systems engineering model. Even people in the Agile community, the pure Agile community, are recognizing system engineering as an element that can have an effect on the success of projects.

So, when we did all this a few months ago lots of interviews and things, what was it that surprised you as we went through this research? What was the novel thing that came out of this for you?

Eileen: We talked to a number of program managers. We talked to Agile software team members. The thing that surprised me the most was that all of the feedback that we got about these interactions was about the service side of systems engineering. It wasn't really about the transformation of artifacts. It was about how the teams engage with each other, how they communicate all across the lifecycle of the program.

Suzanne: To me, that actually makes it more promising for these groups to interact more productively if you can acknowledge that dichotomy and see that that is where the focus is, then you are not going to be quite so caught up in, *Are we doing artifacts exactly the same way as we always did?* It is really more about the coordination, collaboration, the learning that is available to the systems engineers if they interact in this more detailed, direct way with the software development. So, I was very excited too.

I would say that thing that probably surprised me the most, I did expect more of the projects in model 3. I thought we would find more people at this point, we are 10 years in plus in software using Agile. I thought we would see more activity in systems engineering adopting it. That was my surprise, so...

Eileen: I would like to think that that is out there.

Suzanne: It is coming. I think it is something that, like many things of this nature, people don't necessarily advertise until they have got a success under their belts. This is a trial-and-error kind of thing. They don't have a set of frameworks to work from yet. So, they are building as they go.

Well, I want to thank you for joining us today, Eileen.

SEI Podcast Series

I think you have other projects in the Agile space. We have got a contracts paper coming up. We have got other papers in this area that you will be working on. So, I look forward to talking about those in the future with you. Thank you.

Eileen: Thank you, Suzie.

Suzanne: If you'd like more information about [our recent technical note on this topic](#) and the SEI's recent publications in all areas of our work, you can download all of our technical reports and notes at resources.sei.cmu.edu. That will take you to our digital library.

This podcast is available on the SEI website at sei.cmu.edu/podcasts and on [Carnegie Mellon University's iTunes U site](#). As always, if you have any questions, please don't hesitate to email us at info@sei.cmu.edu. Thank you.