



Applying Agile Principles in the DoD: Third Principle

featuring Mary Ann Lapham and Suzanne Miller

Suzanne Miller: Welcome to the SEI Podcast Series, a production of the Carnegie Mellon Software Engineering Institute. The SEI is a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. A transcript of today's podcast is posted on the SEI website at sei.cmu.edu/podcasts. My name is [Suzanne Miller](#). Today, I am pleased to introduce you to myself and [Mary Ann Lapham](#), my colleague. Welcome to our ongoing series exploring [agile principles](#) and their application across the Department of Defense. In today's installment, we explore the third agile principle: deliver working software frequently from a couple of weeks to a couple of months, with a preference to the shorter time scale. And Mary Ann's over here is laughing hysterically, silently.

Before we get into that, a little about today's guests. That would be Mary Ann and myself. Mary Ann's work focuses on supporting and improving the acquisition of software intensive systems. For Department of Defense programs, this means working with program offices to assist and advise on software issues at the system and/or segment level. In addition, she leads research in topics that are germane to software acquisition and acquisition of software-intensive systems, particularly in agile adoption within the DoD.

My research focuses on synthesizing effective technology transition and management practices from research and industry into effective techniques for use in agile and lean methods in regulated settings like the DoD. So, we're pretty compatible in the way we like to approach things. All right, so Mary Ann, this is our third podcast about agile principles.

Mary Ann Lapham: Only nine more to go.

Suzanne: There will be nine more to go. And for those of you listening, if you like what you hear today, there are already two podcasts in the series, and we will eventually have all 12; one for each agile principle behind the Agile Manifesto. As a reminder, the four values and 12 principles of the Agile Manifesto can be found at agilemanifesto.org. All right, Mary Ann. Let's talk about what "delivering working software frequently" really means in DoD settings, and how delivering in a couple of weeks to a couple of months differs from what is currently typical in the DoD. You want to start?



Mary Ann: Oh, where do we start SuZ? Thanks very much for the introduction. As we all know, the DoD has been doing embedded software and IT [information technology] systems for many, many, many years. Over the years, typically, historically, systems take two, three, five, ten years. Right now, the current thinking is 81 months is the average time to deliver an IT system. Well, in today's environment, that's just not cutting it anymore.

Suzanne: That's not a generation, that's a lifetime.

Mary Ann: It's a lifetime. Think about it. You buy your new laptop today, and next week, something new comes out. Okay, you're not going to spend the money for the new one, but you certainly don't want to wait five years to get your next installation of the latest technology. It's become evident within the DoD community and the government in general that we need to do software delivery sooner. So, how do we do that? That's one of the questions that's being tossed around.

Suzanne: It's not to say that every maintenance upgrade on something takes five years, but the major system deliveries take an awfully long time.

Mary Ann: And, the needs change more rapidly than ever before. Your staff comes in with a lot better ...

Suzanne: They have better home technology.

Mary Ann: Better technology, better skills. I mean, you get some First or Second Lieutenant, just graduated from college and he's got his iPad and his iPhone and all his Android toys.

And then you hand him an old PC. Basically he'll look at it and say, "Clunker." And the software is not anywhere near current technology. He can't go and get an app off the internet for this thing.

Suzanne: So, this principle is one that really resonates with people in the field.

Mary Ann: It's resonating. Yes.

Suzanne: And, people in the acquisition process are trying to figure out how to do this.

Mary Ann: Yes, they are.

Suzanne: One of the things that I notice when I look at this principle, and go, "Whoa, boy, this is going to be fun," is that it asks us to deliver *working* software frequently.

What we see in a lot of acquisitions is we get lots of deliveries, but the early deliveries are more documentation, more review meetings, more review meeting slide decks. That's a really different focus for a lot of people in the contract settings where, you know, working software comes after



all that stuff is delivered. So this is saying, “Don’t wait to actually deliver working software.” So, you’ve got to really go from a document-centric lifecycle, and view of the world, to an implementation-centric focus.

Mary Ann: That’s true.

Suzanne: And that’s a culture change for the acquisition systems also, isn’t it?

Mary Ann: It is. It’s a big change. But, one of the things, before we get into that change is, “Okay, I have to go to something more like a working software thing.” Well, working for whom?

Suzanne: True.

Mary Ann: Is it working for the end user? Is it working for some intermediate person?

Suzanne: For the testing group?

Mary Ann: For the testing group. Is it working for the developers who need an infrastructure and so you develop that first, so then they can build the rest of that stuff on top of it. So, “Working for whom?” is one question related to that. Before you even get there, you have to have ground rules up front. This is not going to be your mother’s

acquisition effort. You’re not going to get as much of the oversight done by documentation, which is one of the many reasons why it’s (the documentation) there. There’s got to be a way to monitor these very large programs. And, over the years it’s built up from an acquisition standpoint to be a lot of documentation, because then you can see what’s going on. People tell you, “This is what I’m going to do. This is what I did.” “Yes, you did what you said,” and you move on. Well, with agile, you do similar things, but you do them so much more quickly that you can’t just do all the paperwork first. And, it’s counter to the agile process anyway.

Suzanne: And, one of the reasons you can do all that stuff first is some of the other principles talk about getting your business people and your developers working together. So, you’re really changing the relationship so that documentation, as an intermediary, isn’t the thing you’re focusing on. You’re focusing on getting much more direct contact, which has its own challenges. That’s agile principle number four. Stay tuned for that [podcast].

Mary Ann: Right. It’s also one of the values: working software being valued over documentation. Of course, when the early adopters of the whole agile movement started, some of them, (who didn’t understand the principles that well, I may add), said, “Oh cool! I don’t have to document anymore.” Now, any coder worth his salt would say, “Yippee!” But no, that’s not what it means. It means you do *just* enough documentation for the system and the environment you’re in. So, you still have to have some, but you have to negotiate with your end users, with your contracts people, your acquisition people, “What is just enough in my environment?”



Suzanne: Yes, and there are conditions that will mitigate what kind of working software you deliver. So, when we talk about delivering working software frequently, what's the shortest delivery cycle you've seen on a DoD program implementing agile methods?

Mary Ann: Oh, shortest. I've seen some folks deliver in as quickly as two weeks.

Suzanne: And, who were they delivering to?

Mary Ann: Well, some of them wanted to deliver to the field. In one case, they did that, but that caused different issues. Not that the software didn't work well, but, because of the wide variety of people they were delivering it [to], it caused a whole other set of issues, which we'll talk about in a bit.

I've also seen a month to two months. The latest one I saw on that schedule was really cool. They delivered the software every month or so, every month to six weeks. Well, it was kind of this round robin. It was a very large system, and it had a lot of operational test requirements.

So, they just couldn't deliver it to the end user, they had to go through the operational tests; those events didn't get cut out. So, they delivered a drop to the operational test guys every month, every other month, actually. So, they were offset by a month.

Mary Ann: Then, when the operational test guys got done with it, they would post it outside the program manager's office, so people could walk by and see, "Ooh, that's what we're going to get. That's really cool." And, then a month later, they'd get another drop out there, and they [would say], "And, they added this to it."

They showed what the software was going to give them, so they had an idea what was coming. And, because it was a small piece of a much, much, much larger system, they weren't going to get the full capability in the operational environment for several years. But, this gave them an idea of what they were going to get and also helped the users say, "Well, yeah, that's good, but could you change-- whatever?"

Suzanne: Right. So, they were able to interact with the system.

Mary Ann: They could interact with the system.

Suzanne: Much, much earlier than normal.

Mary Ann: Because they did it that way, not only did it give the users a chance to interact, it gave the sense of movement, if you will, doing it in a more agile method. It was a hybrid. It wasn't pure agile because of the constraints in the environment. It gave them a lot more proponents from their user community. They were like, "You mean, I can really get software that



works? Oh, that's really cool. Can you show me more?" It created a lot of excitement within the organization, which gave them more support and gave them more...

Suzanne: Ability to actually deliver it frequently. So, it created what Bill Novak would call a "virtuous cycle."

Mary Ann: Yes, it did.

Suzanne: Yes, and that's important. I've seen that one month-to-two-months delivering into operational test is actually seeming to become more common. I have never seen anybody try to deliver in two weeks on the DoD side. I've seen it in commercial, but not in DoD.

The trick is understanding, as you said, negotiating, "Who are we delivering to? And how are we eventually going to get out to those end users so that they get the benefit of this, not just the test community here or the certification community."

Mary Ann: Now, I must say, the one I saw, just to be clear here, for those who are going, "Two weeks? How could you possibly get through all the certification/accreditation stuff if you're in a DoD system?" Well, let's just say it wasn't at that level. It was a very small system. It wasn't a program of record. They had a lot less restrictions on what they could do. And, then they got more popular, and then they started to run into those other issues.

Suzanne: Yes.

Mary Ann: Slowed them down.

Suzanne: Success can also have its own issues. So, when you start delivering faster, what happens in the DoD? We've got bureaucracy on the acquisition side, but then we also have our set of bureaucracy on the operational deployment side. What have you seen?

I've seen some things in terms of people not being able to receive the software once the agile program starts delivering it more frequently. Why don't you talk about that a minute?

Mary Ann: Well, it's not that they can't receive it; they can get it. Okay, this one system we know of has about 600 users worldwide. Now, those are organizational entities; within each of those user groups, there's probably a hundred people, maybe. So, 600 organizations, installations, if you will, with multiple instances within their group. Well, okay, 600 different places. You send out the software. Now, you could send it out on [Pushit](#), CD, whatever's appropriate from a security standpoint. And, if you do it every month, then the guys in the operations have to upload it, they have to test it to make sure it's not breaking something in their environment. They have to make sure it doesn't screw up their data. Even though this is all tested, each operational environment, as we all know, is slightly different.



Suzanne: True.

Mary Ann: So, they get it ready to go, and they put it out there. Then, along comes, within a week or two, the next drop. Well, they hardly got the new one out for people to use yet. So, they end up with this churn of “I’m always putting in the new software,” as opposed to “I put it in. My users get to use it. They see the benefits. They get ideas of what they want to change next.” Then, they get another drop within a reasonable amount of time.

So, a lot of times, the programs that are putting the stuff out there that fast have to negotiate with their end user and say, “Okay, you can’t do it every month, probably not every two months. What’s the best cycle for you guys?”

What they do then is instead of having iterations that they send, they group them into releases. A lot of them end up with about an 18-month cycle because it allows the operational community to get it all installed across the board, all 600 installations or whatever number there is; get a chance to use it; shake it out in an operational environment; and then [provide] feedback to the developers. Then, the next one comes along with more improvements, and it seems to be a good rhythm.

Suzanne: Now, I have heard on one program I talked to, that they are deploying to mobile devices. In that environment, it seems to be this cultural difference where people are used to accepting ...

Mary Ann: Upgrades.

Suzanne: Upgrades of their software almost in the background, and they’ve actually been able to deploy very quickly because they’re not changing their security configuration and things like that. I find that to be really interesting; that we can get people to accept things faster on mobile devices because we have this culture that thinks that’s ok...

Mary Ann: But, it gets back to the culture that people come in with from their home environment.

Suzanne: Sure.

Mary Ann: How many people have an iPhone, an Android, and you get these little notices: “Updates Available.” You just push the button and next thing you know, you’ve got the new operating system or whatever it is, the new app, version of the app, and off you go. No big deal. And, they do it all the time. Some of these people have been doing it all their adult life.

Suzanne: So, one of the things that I think about with *delivering working software frequently*, it’s not just about the delivery. It really is about sort of changing the culture of “How fast can we accept change?” Not only “How fast can we produce change?”



That's one of the things that's not as highlighted in that principle, that in the DoD environment in particular, I think that's one of the things that we're seeing. It is one of the things that we do consider when we're talking to people about agile methods and DoD in regulated settings -- you've got to deal with both sides of that question.

Mary Ann: Right. You just made me think of something. A lot of these environments, it's not just updating the laptop or the PC or whatever it is that's hosting the software. A lot of the environments, the networks that these PCs are hooked to, there's a whole other layer of work that has to be done to get it accepted for use on that network and that base of operations.

Suzanne: Sure, from a security standpoint.

Mary Ann: From a security viewpoint and sometimes from an operational viewpoint. They have very rigid rules on how they're going to do it. Just because the stuff is coming in quicker, those rules aren't necessarily going to change. Now, again, they could evolve over time, but it's something that they're going to have to work on.

Suzanne: Other principles will get us into some of this.

Mary Ann: Other principles will get us there.

Suzanne: All right. Mary Ann, thank you so much for joining us today. In the next episode in this series, we will explore the fourth agile principle, which I briefly alluded to: *business people and developers must work together daily throughout the project*. So, that will be an interesting discussion, especially in the context of the DoD.

If you'd like more information about the SEI's recent research, this and other topics, you can download all of our technical reports and notes at sei.cmu.edu/library/reportspapers.cfm. Listings for papers, blog posts and podcasts related to our SEI research on agile adoption and the DoD can be found at sei.cmu.edu/acquisition/research.

This podcast is available on the SEI website, as I mentioned, 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.