



Applying the Second Agile Principle in the Department of Defense Second in a Series

featuring Suzanne Miller and Mary Ann Lapham

Suzanne: 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](#), and today I am pleased to introduce you to both myself and my colleague, [Mary Ann Lapham](#).

Mary Ann Lapham's work focuses on supporting and improving the acquisition of software-intensive systems. For Department of Defense [DoD] programs, this means working with program offices to assist and advise on software issues at the system or segment level. In addition, Mary Ann is the lead of the [ASP Agile acquisition research team](#) of which I am also a member.

My work focuses on technology transition-based stuff, and I apply it to all kinds of things. I've applied it to [CMMs \[Capability Maturity Models\]](#). I've applied it to systems of systems, and now I'm applying it to Agile acquisition and Agile software engineering. So, we both have come with different backgrounds but all for the same goal, which is to help the DoD with barriers to adopting Agile in their settings.

We've talked in other podcasts and blogs about what we mean by Agile, but for the purposes of this podcast, what we're really talking about are organizations that are trying to express 4 values and 12 principles that are published on the [Agile manifesto](#), that's one word, agilemanifesto.org website. It is essentially a philosophy of human-centric software engineering that was promulgated starting in the 2000, 2001 timeframe.

In this series Mary Ann and I have been talking about each of these [principles](#) in terms of what we see as barriers to adopting that principle in the DoD as well as some successes that we've seen with people trying to adopt or express a particular principle. Today's principle is the second Agile principle and is stated as, "Welcome changing requirements even late in development. Agile processes harness change for the customer's competitive advantage."



Mary Ann Lapham: When you say, “Welcome changing requirements, even late in development,” I figure our DoD audience is cringing in their seats right now, just cringing, because this is not how the traditional development, the traditional non-Agile work is done, where you define all your requirements up front. You have a [PDR \[preliminary design review\]](#), then a [CDR \[critical design review\]](#). Then you start doing your actual coding and so forth. To not have that comfort zone, that safe blanket that they have been taught is very difficult; it’s a very challenging change for them to overcome.

Suzanne: But, on the other hand, we’ve both worked in programs, and we’ve both observed programs, where if you lock down those requirements at too fine a level too early you do two things:

1. You don’t allow for changes in the environment that happen in between when you think of what it is you want to do, the system you want to create and when you implement it.
2. The other thing is you don’t allow for learning, because what happens when we are doing software engineering in particular is we’re learning things about how the system can work, things about how it can’t work, as we’re going along. If you don’t have an opportunity to change requirements easily then you really lose that opportunity for learning.

So, we’ve got a real paradox here in traditional acquisition of we want to have certainty about the requirements so that we can cost and schedule how they’re going to get implemented, but the real world doesn’t really accommodate fixing those requirements so that we can have this nice march towards completion, right?

Mary Ann: Correct. In some of the requirements—and that’s very true for the functional requirements—I want my system to do X. But then again, DoD has these other requirements for things like certification and accreditation. Those do have to pretty much stay stable and set. However, your vulnerabilities and the things you’re protecting against will change.

In today’s environment, as we all know, pick up any newspaper and “cyber attack this” and “cyber attack that.” You need to be able to defend against those, too. So, the Agile methods do allow a little bit of change, and you can [change]. It doesn’t have to be, “everything is allowed to change,” but there’s degrees of it that will allow them to react to these new cyber threats and not have to do the system totally all over again. So, it does have a place there too, but it’s a matter of degree. You have to figure out what’s most important for your user.

Suzanne: So, when I think about this principle, I look at it and I think, as you say, “There are some requirements, there are some places, where that’s just not going to fly.” Things like flight



certification. The laws of physics don't change and software has to make sure that it doesn't violate those laws, as well as things you talked about like certification.

The other thing that I look at when I think about this from a DoD view point is, "Who's the customer?" This principle says, "Agile processes harness change for the customer's competitive advantage." If you switch that perspective to the operational user's competitive advantage, now you've got a whole different thing going on, because we've got all of this rhetoric from the DoD leadership that says "We need to stop trying to provide 100- percent-perfect solutions. We need to be able to deploy an 80-percent solution, get it fielded quickly." You know, the [25 points](#), the [804 response](#). These are all things that we'll have references to on our website.

There's lots of rhetoric about how we need to get stuff out to the field faster. One of the ways to get it out to the field faster is to not presuppose that all of the requirements that we think of at the beginning have to be enacted and implemented. We need to prioritize them in a way that allows us to get something out there that the people can try, so the learning can occur.

Mary Ann: One of the key things, if you're going to use Agile methods, is have enough definition up front of what you want to do, but not so much detail that you can't learn, that it can't change, because your environment changed.

You've talked to all your users, the people, the customers. The user in the end is the guy who's really the customer. Now, one of the things that you have to be careful with there is, "How many different sets of users do you have? Are you a joint program?"—then that gets more complicated. Even if you just have different units within the same service, you have to make sure you account for all those. There will be differences of opinion as to what the requirement really is, so you need to get them to agree on the prioritization.

Sometimes it's a horse-trading thing where, "Well, I really want this, but you want that. So, if I help you with that one, we'll do that one first. And, you can help me get mine done next. We've seen programs that are using Agile methods; they have exactly that going on with their users. They do a lot of horse trading before the next release is defined. Everybody's happy, because they all get a little bit of what they want.

Suzanne: So, if you want to be a program that is able to welcome changing requirements, the first thing you have to ask is, "Is my acquisition actually capable of accommodating changing requirements?"

A couple of things you need to be aware of in writing your contract, writing your [RFP \[request for proposal\]](#), are things like, "Are we going to set the baseline up too early? Are we going to close things down too early? Are we going to allow for the detailed requirements to evolve?"



So, instead of having software-requirement specs that are very detailed, “Can we tolerate having software requirement specifications that are high-level and allow for the conversations to occur that need to occur between the operational user and the developer, so that we can get the best idea of what the priorities are?” Things like that are things you can do. Things like making sure that your technical reviews within your systems-engineering process allow for incremental reviews at the software level, so the preliminary-design reviews and the critical-design reviews [aren’t just reviewing documents, they are including demonstrations of working software](#).

We’ve seen one Marine program that has their preliminary design review inside of a three-week iteration, and their critical-design review information is presented at the end. They’ve got working software plus all their documentation for PDR and CDR all done within the same iteration. They had to set that kind of stuff up front, so that’s one of the big things. To do this principle, to enable this principle, you’ve got to think about things ahead of time. Or, if you’re in the middle of your acquisition, you may have to go get some waivers and some permissions from your systems engineering staff and some of your compliance staff to be able to enable that.

Mary Ann: Not only enable it, but you’ve got to make sure all the stakeholders—the compliance staff, the system engineers, even the testers and integrators and operational testers in particular—all know what you’re doing, and they have agreed to it. Because they know it’s not going to be business-as-usual for them. They will get the stuff, but they have to be patient in some cases where you’re doing it in an evolving manner. They all have to agree to that up front, or you just won’t succeed, because there’ll be so many road blocks. You’ll just have to stop, and go back to the old traditional methods.

Suzanne: One thing I think we both agree on is that you’re not going to see a complete expression of this principle in many DoD settings.

Mary Ann: Probably not, no.

Suzanne: Certainly in comparison to commercial, the “late” part, it’s going to be later than is expected in a traditional acquisition, but it’s not going to be as late as I would see in a commercial setting.

Mary Ann: That’s true, and it’s not like the traditional setting doesn’t have a way to do changing requirements, because they do. They have change boards, but the change boards have to be a lot more open and flexible, and not as rigid as they have been in the past, to allow for the evolving knowledge to come out from an Agile application.

Suzanne: So, I think we’ve covered most of the points about things that you should be aware of if you want to welcome changing requirements and a couple of reasons why it’s a good thing to welcome changing requirements even though it is very antithetical to our traditional acquisition environment. One of the things that Mary Ann and I are interested in is we have not seen it yet



but there is circulating a DoD-5000 draft revision, which is the governing document for many of the practices in DoD acquisition. We know that people have been trying to put things that are more Agile-aware into that document. So, it's going to be interesting to see if some of the changes they make to that document will enable this to be a little easier for DoD programs in the future. So, stay tuned.

Mary Ann: Good luck, and I'm anxiously awaiting that document too to see how it helps.

Suzanne: Anyone that has their own experiences that they'd like to share with us, please use the [info@sei.cmu.edu email](mailto:info@sei.cmu.edu). That is how you can give us any kinds of questions with any of our podcasts. Also, be aware that we will have this transcript up on the sei.cmu.edu/podcasts page, and our papers as well as other technical notes can be found at sei.cmu.edu/library/reportspapers.cfm.

In addition to the SEI Podcast Series site, we also host these podcasts on [Carnegie Mellon University's iTunes U site](#). And if you have any questions, again, please don't hesitate to email us at info@sei.cmu.edu, and we welcome your shared stories there as well. Thank you very much for joining us, Mary Ann, and thank you all.

Mary Ann: You're very welcome, Suz.

Editor's Notes:

- *In a few instances, we have edited the printed text, which may differ from the audio, to improve clarity for the reader.*
- *For additional information on Mary Ann and Suzanne's research in Agile Acquisition, please visit <http://www.sei.cmu.edu/acquisition/research/>.*
- *In the interim since this podcast was recorded, Suzanne and Mary Ann would like readers to note that "incremental fielding" is new wording that appears in the above-mentioned DoD-5000 draft revision. If it stays, the additional language will provide better support for agile adoption in acquisition settings.*