Agile Metrics
featuring Will Hayes and Suzanne Miller

Suzanne Miller: Welcome to the SEI podcast series, a production of the Carnegie Melon University Software Engineering Institute. The SEI is a federally funded research and development center sponsored by 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. Today, I am very pleased to introduce myself and Will Hayes. In today's podcast, we're going to be discussing Agile metrics. This is part of our ongoing work to help the Department of Defense and other organizations in their efforts to adapt and use Agile software development methods. So, this is a chance where I get to be both the interviewer and one of the interviewees because I participated in this research. So I'm very excited about this. But first, a little about myself and Will Hayes.

Will provides direct lifecycle management support to major software-intensive programs in government and military organizations. He also researches and consults in the application of Agile methods in highly regulated settings as well as the innovative application of measurement and analysis methods in any setting. My research focuses on synthesizing effective technology transition and management practices from research and industry into effective techniques for used of Agile and lean methods in regulative settings. So, you can see why we like working together. Welcome, Will.

Will Hayes: Thank you. Thank you.

Suzanne: Why don't you give some background on our research, obviously it's the DoD—Department of Defense adoption of Agile methods—and how you got involved in this.

Will: With many of the contracts that we hold and the work that we do in supporting program offices in government and military organizations, they are finding that the providers are adapting Agile methods and finding new ways to deliver product on a more rapid basis in smaller increments than is traditionally seen in those kinds of environments. And it's a bit challenging for program offices to find the best way to take advantage of those techniques.

We are finding ourselves working with many such program offices, as well as providers, to find the ways that they have found to be successful with these new approaches in what some might view as a more traditional, old-style way of managing contracts.

Suzanne: You have never been one that holds with old-school ways of managing things. I know that Will has been involved in SEI metrics research for-- I'm going to make him sound old here because it has been decades now, Will.

Will: Yes, it has.

Suzanne: You bring a particular focus on measurement into this space, which is one of the reasons that we are very excited that you were able to be the lead author on this paper. So, let's talk about that piece. Our technical note is called *Agile Metrics: Progress Monitoring of Agile Contractors*. Talk a little bit about why we focused on the progress measurement aspect of this and the contractor aspect of this because this is a little bit more specific than some of our other technical notes.

Will: When it comes to the due diligence-based monitoring that is traditionally seen in government and military contracts, measurement plays a fairly important role. There is an elaborate infrastructure and a fairly well understood set of definitions for measures that are traditionally used in that space.

In contrast to that, when we look at most of what is written and trained and discussed about the term Agile, the focus tends to be on a team of seven plus or minus two individuals working together in a self-directed team with a high focus on trust and an empirical approach, but mostly within the team. This is not to say Agile methods don't work or aren't being used with larger groups, it's just what is discussed and what is published tends to focus fairly narrowly as if the attention to those other concerns isn't of issue. We are finding in application, there are people really doing a good job filling in those abstractions between the small team-focused measurement and what you would need at an enterprise or at least at a program level. So, one of the challenges we have is to find a good way to answer the needs of the large-scale-program-management focus without violating the kind of space that is created for a self-directed team to use Agile methods and be successful.

Suzanne: One of the things that—in any measurement program, it's not just Agile management but especially Agile programs—measurement needs to be used to serve the team not to punish individuals. That is obviously one of the things that gets written about a lot. So, one of the challenges in trying to get progress metrics is how you do that at the management level without violating that premise of not using metrics to punish individuals. So, why don't you talk about some of the things we found in terms of how people are dealing with that?

Will: So, one of the challenges we have in almost any methodology you might choose to use in a medium to large size program is that measurement is often done for other people. It's an obligation I take on to collect some data that someone else is going to use. So, often people have a fairly defensive posture when they are thinking about providing metrics. In fact, many people who would train you in Agile, people who would help you to become a Scrum Master, think of the word metric as perhaps akin to one of those words that George Carlin told you, you can't say on television. But in fact, Agile methods are touted, and indeed serve as, an empirical approach to software.

Suzanne: So, explain to those that aren't familiar with that term. What is different about an empirical approach to software in terms of other approaches?

Will: So, the idea is to really enact that plan-do-check-act cycle that Edwards Deming spoke of so long ago, but at a much more near term, much more immediate, much more individually focused level. So, there is a much more frequent conversation among the people who are working together to develop product, and they are speaking in much more direct terms about the product itself, not the intermediate work products that eventually lead to working software. So, the idea that we see expressed in the Agile Manifesto and the principles that accompany it is that the demonstration of the capability is the most direct way to view the progress that you're making. When they say empirical approach, they are referencing often that we're looking at actual working product, not an abstraction represented only on paper, though obviously the abstraction served an important purpose in arriving at that product. The idea is to get to demonstrable results and to speak about data collected by the team for the team's use. Things like *velocity*, which is an expression of how many story points have been delivered successfully at the conclusion of a given iteration or sprint.

Suzanne: Story points are a common way of doing relative estimation in Agile methods as some of our listeners will be familiar with. But if you are not, we have references on the website you can look at for story points [in particular, the SEI technical note, Agile Methods: Selected DoD Management and Acquisition Concerns]. The thing about progress measures that are empirical, that are observation-based, but that don't violate that team spirit—that to me was the real crux of what we are trying to get at with this technical note: *how do we help program managers and program office personnel appropriately measure progress and still allow the team to work in their own environment and use metrics for them that really are only for them and differentiate from the metrics that are used for measuring at the program level*. So, talk about some of the program level measurements that we thought might be useful in this setting.

Will: One of the most commonly referenced measures—I've already mentioned it, and I probably will be speaking of it multiple times before we are done—is this notion of velocity counted in story points. The thing about story-point estimation and relative estimation is that it is not a universally calibrated metric like we might consider lines of code or function points to be.

It is designed specifically for the team to be able to look at the work they are doing with the set of expertise they have on hand and get a sense of what it is they can commit to. By monitoring a team's ability to consistently meet its commitments—when we look at velocity as a metric at a higher level than the team—what we are really wanting to see is not necessarily how many story points they push out each iteration, but how consistently they are able to meet their target. To some extent, the degree of fluctuation in how many story points are delivered or the extent to which they meet or exceed their story-point estimate is the more interesting thing to look at than just the number of story points themselves.

Suzanne: So, velocity is one of those measures that can be used both by the team for their internal purposes, and it can be used externally as you say to understand the team's consistency as long as you don't try and compare the velocity of one team to another. One of the pitfalls of measuring velocity is that it is a measure that is unique to an individual team. It is not something that is comparable across teams, which is one of the things that drive program managers a little bit crazy because they are accustomed to more absolute measures that you mentioned like function point and source lines of code, that are traditionally more comparable, at least in theory, across multiple teams.

Will: Having said that, it is, I suppose, a certain matter of orthodoxy on whether or not you strictly abide by this team-unique calibration versus some of the efforts that we have seen for people to try to calibrate across teams at a release level when we have different teams working together.

Often, the notion of ideal days is used as a calibration method. That is basically to ask people who are doing estimating to try to size the work at hand described in a user story based on how many ideal days of work it would take to accomplish that. Now, ideal days are uninterrupted eight-hour periods, nine-hour periods, whatever the customary day length is depending on where you are in the world. And, to understand what would it take to understand, implement, and test the function as described in the user story. With ideal days as a bridge, we are seeing some people proposing ways to calibrate more broadly. On the other end of the extreme though is this notion that velocity is yesterday's weather. That is a term that I've heard used most specifically with Scrum. It does a nice job, I think, of explaining the local nature of velocity for a team. It is affected by seasons, and it is affected by geography if you think of it that way. So, different teams will have different velocities. Perhaps the same team—at different periods of time or with new work or as they finish a large bit of work—they have kind of hit their stride. And so, yesterday's weather will be different than last week's weather. But yesterday's weather is a good way to anticipate what today's weather will be. So, you won't necessarily use just yesterday's weather to plan a whole season ahead.

Suzanne: Especially if you are a farmer, but, OK, I take that one back.

Will: We're in Pittsburgh.

We have farmers though.

Suzanne: So, if I don't speak about this particular measure, the people in program management will say, *Well, these people clearly don't know what they're talking about*. Will is nodding his head because he knows what I'm going to ask him about next, which is [that] earned value measures is a particular approach to measuring progress that is mandated on government programs above a certain size. The idea is that you declare ways of indicating progress towards your goal, and the goal being a product. In the more traditional settings, you are looking at some of those abstractions that you talked about, the intermediate work products. You are measuring your progress against those with the idea that that is going to give you some sense of how close you are to the end product. How does earned value measurement play out, and what is its utility in an Agile program?

Will: This is a particularly challenging area for people who are working on very large defense contracts because there is a mandate. One of the things about earned value management is that it relies on a work breakdown structure that for most organizations tends to be deliverable-item and individual-task focused. In an Agile setup, you are focused more on the functionality as expressed in a user story, and you don't focus so much on the size of the intermediate work products that express that capability with an eye toward getting to that working code as quickly as you can, in contrast to having a very well specified software requirements spec, an elaborate design that gives you full view of the architecture in various implications. The idea is to slice a narrow piece of those constructs, if you will, and to implement working code, so that we can demonstrate to the user the function that we are speaking of in the user story. There is some very nice work done by individuals that are researching this from a perhaps more academic perspective under the title [Agile EVM [earned value management]](#).

There are also corporations that have started to define their only implementation of [earned value management](#) where story points end up being the fundamental atomic unit of measure that serves the computations. One of the biggest challenges that Agile proponents and consultants will express is the idea of committing to a work breakdown structure in contrast to committing to delivering functionality that is approved by the user at each step along the way. Those are seen by many Agilists to be in conflict.

Suzanne: One of the things that came out of this work from my view point is a resurgence of interest and confidence in the rolling-wave planning methodologies that I learned about in the early 1980s, because they actually provide a way of connecting the earned value kinds of constructs with the product backlog idea that is consistent with Agile methods. One of the promising things to me was that if you can construct your work breakdown structure in a way that acknowledges the kind of work package sequence that is common in rolling-wave planning,

then you can get a little bit of a bridge. We saw some examples of that in our research. So, I thought that was very promising as well as some of the more theoretical research around Agile earned value management.

Will: That is one of the places where people that are proposing well-defined methodologies for scaling from the team level to the larger level and really give a lot of assistance in talking about architecture, talking about the whole picture and not just what a team of seven plus or minus two people can do. So, there are notions of intentional architecture, notions of having some degree of resolution and visibility further down the road than just the next sprint or two. That is a place where I think we're going to continue to see a lot of development and a lot more publication and help coming to organizations that really want to make Agile work for them because, I think, we've heard people talk about how it really does speed up realization of operational capability. It allows users to give feedback, and, therefore, the notion of enhanced quality moves beyond just lack of defects and a lot more toward the functional capabilities that serve the missions that the software has to support.

Suzanne: Those have always been challenging aspects to measure: usability, reliability, and availability. So, having the kind of fast feedback loop that we get in Agile means that you can actually tune those measurements. I know one of the things that Will is excited about in terms of Agile methods is the fast feedback loop. So, in many projects, in a traditional setting, you didn't have very many instances of the design-implement-test cycle.

With Agile, you have many more of those cycles, which for a measurement specialist means more data points. We know how that makes you light up.

Will: In fact, there are things that we've encountered in speaking with teams that are using these methods that I've never seen described in writing and publications, and that is things like the use of static code analyzers, understanding through using modeling techniques the likelihood of meeting particular targets. Those are not things that are considered part of the Agile orthodoxy, and some may view them to be in conflict. But, what we are seeing, when we interact with teams that are successfully using these techniques, is that the insight they get from analyzing the code with specialized tools helps them adjust what they do. Those adjustments can occur more frequently.

Suzanne: That is back to the inspect and adapt mentality that is common in the Agile framework.

Will: The insight they get too from things like Monte Carlo simulation gives them a more confident basis for some of the commitments they are making with teams. These are not people who are doing these things because they are somehow required to or rewarded to do so by some external certifying body. These are people who are using tools to help them get that empirical

feedback, to get the success they are after with delivered product, not conformance to a particular set of steps for getting there. It is really exciting to see those techniques in play.

Suzanne: So, that leads us into the future. So, what are some of the areas that you see us pursuing in the future related to Agile metrics, both in progress metrics and other kinds of metrics?

Will: So, one of the great privileges we've had here at the SEI is to have access to a very substantial amount of actual operational data from one of the vendors of an application lifecycle management tool. We had hundreds of thousands of rows of data to look at. In trying to analyze the data, one of the things that I learned from the experience was, in a traditional use of measurement, in old style programs if you would, we are waiting for the data to accumulate at the bottom of the waterfall, so we can aggregate it, analyze it, and give comments from a retrospective perspective.

What we are seeing with Agile teams is they don't want to wait. We need to analyze that stream of data before it has a chance to pool at the bottom of the waterfall. I am using the metaphor somewhat liberally here. But, the techniques that you need to do—*How do you snap a baseline? How do you have reference point when you're looking at things in constant motion?*—that is a very nice challenge to have from my perspective. The graphical representations that are required in order to really make sense of that are perhaps not the traditional ones we have seen. The kinds of analysis techniques you need to apply, the baselines you need to have available to you, those take on a slightly different form in that context.

I think we are going to find that the industry is going to move toward more near-term, more immediate feedback, and the intelligent use of historical baselines. I think the quality of the criteria that will be applied will have to improve, because we can't wait for the large numbers to kick in for us to say, *We're reverting to the mean or not*. Those sorts of reference points are not as germane when we're looking at near-term, shorter cycle…

Suzanne: With smaller data sets.

Will: Small batches, yes.

Suzanne: But they are meaningful data. That is meaningful to the near term. Not meaningful to two years from now.

Will: So, it's like on your automobile you've got a gauge that tells you what kinds of efficiency you're getting now. *How many miles per gallon am I getting now?* I don't wait until I get to the rest stop and fill up to guess or calculate what the efficiency of the engine is. Not that I would ever be able to use that data in the context of driving my automobile. But, the ability to have measures that help you gauge what you're doing now as opposed to what we did last month or

last quarter. I think that leads to a lot more efficiencies and the opportunity to have much more innovative focus on the product, the design, the architecture, because we are able to fine tune and get a steady cadence for the development work we're doing. I think that is a very exciting future for us.

Suzanne: I agree. I am looking forward to working with you on it.

Suzanne: Will, thank you so much for joining us today.

Those of you listening, if you're interested in accessing our research on Agile adoption in the DoD via our papers, blog posts, and podcasts, please visit sei.cmu.edu/ acquisition/research.

Also, please remember that you can access a series of podcasts that I conducted with Mary Ann Lapham exploring all 12 Agile principles and their application across the Department of Defense at www.sei.cmu.edu/podcasts/agile-in-the-dod/.

The SEI technical note [*Agile Metrics: Progress Monitoring of Agile Contractors*] describing our research as well as recent publications in all areas of the SEI's work can be downloaded at resources.sei.cmu.edu.

Finally, this podcast is available on the SEI website at sei.cmu.edu/podcasts and on Carnegie Melon 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 for listening.