

Practical Considerations in Adopting Agile/Lean

Table of Contents

SEI WEBINAR SERIES Keeping you informed of the latest solutions.....	3
Carnegie Mellon University.....	3
Copyright 2017 Carnegie Mellon University.....	4
Practical Considerations in Adopting Agile/Lean in Government Settings	4
Agenda	7
Polling Question 1	8
Motivation for Agile: Gov't Acquisition and Innovation.....	10
Agile Collaboration Group – Growing since FY12	11
Optional Inner Title Slide	12
Agile Manifesto—STILL the Basis for Agile Thinking in Industry	13
Polling Question 2	15
Useful Interpretation of Agile Principles for Government Settings (1/3).....	17
Useful Interpretation of Agile Principles for Government Settings (2/3).....	21
Useful Interpretation of Agile Principles for Government Settings (3/3).....	21
What are the Agile tenets and principles trying to enable?.....	26
Useful Interpretation of Agile Principles for Government Settings (3/3).....	27
What are the Agile tenets and principles trying to enable?.....	31
Going Beyond Small Teams Adds Complexity	35
SAFe is Also Principles Based – Lean Engineering and Startup Principles	36
Optional Inner Title Slide	40
Agile Methods are Used in Context of a Larger Program (More Often Than Not).....	41

Components of the Eco-System.....	43
Optional Inner Title Slide	50
Polling Questions 3& 4.....	51
Gimmes & Gotchas	56
Polling Questions 3& 4.....	57
Gimmes & Gotchas	59
First Steps Gimmees	60
First Steps Gotchas.....	63
Readiness Gimmees	65
Readiness Gotchas	67
Implementation Gimmees	68
Implementation Gotchas	69
Optional Inner Title Slide	70
SEI WEBINAR SERIES Keeping you informed of the latest solutions.....	73

SEI WEBINAR SERIES | Keeping you informed of the latest solutions



Carnegie Mellon University

Carnegie Mellon University

This video and all related information and materials (“materials”) are owned by Carnegie Mellon University. These materials are provided on an “as-is” “as available” basis without any warranties and solely for your personal viewing and use.

You agree that Carnegie Mellon is not liable with respect to any materials received by you as a result of viewing the video, or using referenced websites, and/or for any consequences or the use by you of such materials.

By viewing, downloading, and/or using this video and related materials, you agree that you have read and agree to our terms of use (www.sei.cmu.edu/legal/).

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

© 2017 Carnegie Mellon University.

Copyright 2017 Carnegie Mellon University

Copyright 2017 Carnegie Mellon University

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

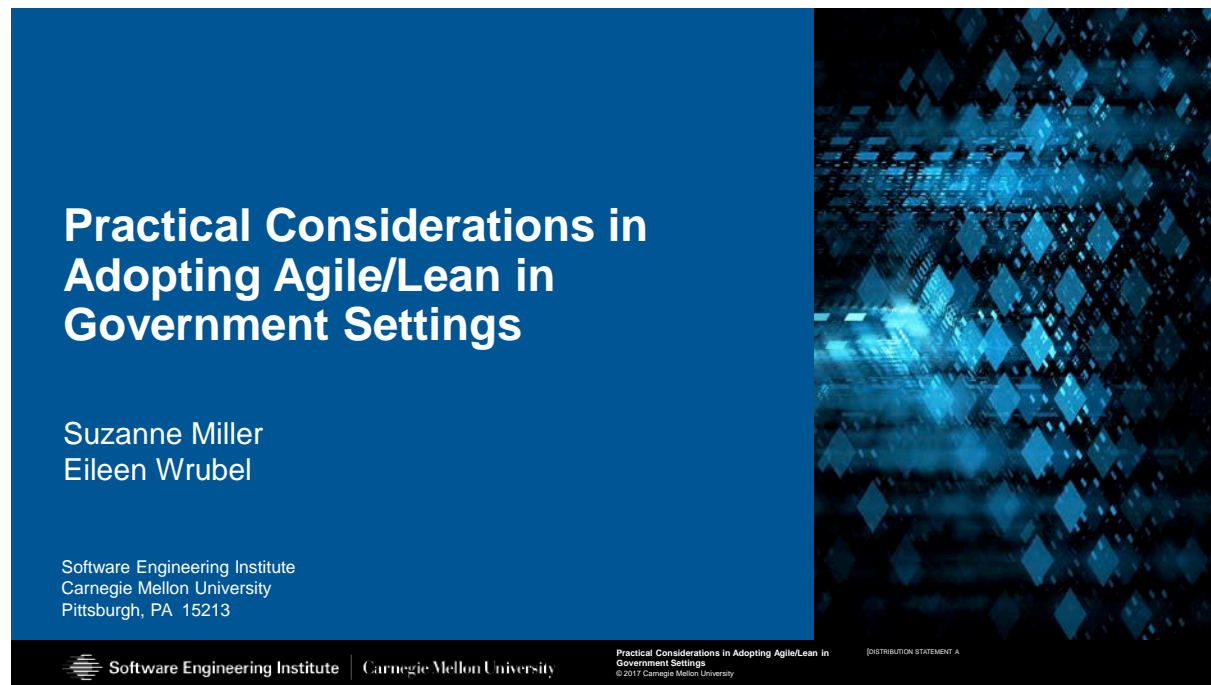
[Distribution Statement A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM-0004509

Practical Considerations in Adopting Agile/Lean in Government Settings



**004 Moderator: And hello from the campus of Carnegie Mellon

University in Pittsburgh, Pennsylvania. We welcome you to the Software Engineering Institute's webinar series. Our presentation today is "Practical Considerations in Adopting Agile and Lean in Government Settings." Depending on your location, we wish you a good morning, a good afternoon or good evening.

My name is Shane McGraw. I'll be your moderator for today's presentation. I'd like to thank you for attending. We want to make today as interactive as possible, so we will address questions throughout the presentation and again at the end of the presentation. You can submit your questions to your event staff at any time through the Questions tab or the Chat tabs on your control panel. We will also ask a few polling questions throughout today's presentation. They will appear as a pop-up window on your screen.

In fact, the first polling question we'd like to ask is, "How did you hear about today's event?" And the rest of the polling questions that we ask will help drive the flow of today's presentation.

Another three tabs I'd like to point out are the Download Materials, Twitter and Survey tabs. The Download Materials tab has a PDF copy of today's presentation, along with related work and resources from the SEI. For those of you using Twitter, be sure to follow @SEInews and use the #seiwebinar.

Now I'd like to introduce our presenters for today. Suz Miller is a Principle Researcher at the SEI. Her current research is focused on synthesizing effective technology transition and management practices from research and industry into effective techniques for use in the governance of programs adopting or contemplating adoption of Agile or Lean methods.

Eileen Wrubel is a Tech Lead for the SEI's Agile in Government Team. She's responsible for coordinating strategic planning and tactical execution of the SEI's initiative to explore Agile adopters in the DoD and federal agencies. Her work focuses on addressing adoption barriers through Agile acquisition and transitioning Agile software development methods to scaling approaches into those settings.

Eileen, Suz, welcome. All yours.

Presenter: Thank you.

Presenter: Thanks, Shane. Thank you all for joining us today. We've got a jam-packed hour talking about adopting Agile and Lean in government settings.

Agenda

Agenda



Setting the Stage

Interpreting Agile & Lean Principles in Government Settings

The Eco-System of Agile in Government

Gimmees and Gotchas in Adopting Agile/Lean in Government

Getting Off the Stage

**005 So we're going to take a few moments now and talk about why we're all here and then we'll get down to the nitty-gritty.

Polling Question 1

Polling Question 1

Which of the following best characterizes your interaction with Agile/lean methods?

- a) Government program contemplating using Agile/Lean methods
- b) Government program currently using Agile/Lean methods
- c) Government program that considered using Agile/Lean methods but decided against it
- d) Government contractor currently using Agile/Lean methods for government work
- e) Agile team or enterprise coach working in or has worked in government settings



**007 Moderator: Okay. Like I said, we're going to ask a few polling questions throughout today's event, so the first question to kind of get an idea of who's in the audience where we can tailor the talk is, "Which of the following best characterizes your interaction with Agile and Lean methods?" And I'll let you, we'll give you, about 15 or 20 seconds to vote, and we'll kick it back to you guys.

Presenter: Suz, while we're learning a little bit about the audience, why don't you talk to us about what makes Agile so interesting for you?

Presenter: So for me, this is a little bit different kind of work than what I've done at the SEI before. Working with CMMs in CMMS, working with systems of systems, architecture, a

lot of our focus has been on what are the practices that we need to recommend to people to use? Agile is principles-based. It's not--practice is derived from the principles, but they're not the key element, the principles are the element. So transitioning something that is about principles and mindset shift is different, and that, to me, is very interesting. It is useful for us to think about that in all of our work but Agile in particular gives us that challenge.

Presenter: Okay. Great. Thanks.

Moderator: And I'm going to chime in with those results real quick. So we got 32 percent are a government contractor currently working with Agile/Lean methods for government work; 29 percent Agile team or enterprise coach; 2 at 19 percent, one, the first one a government program contemplating using Agile, and 19 percent government program currently using Agile.

Presenter: Okay.

Presenter: Okay. Great. That gives us a--

Presenter: Thank you.

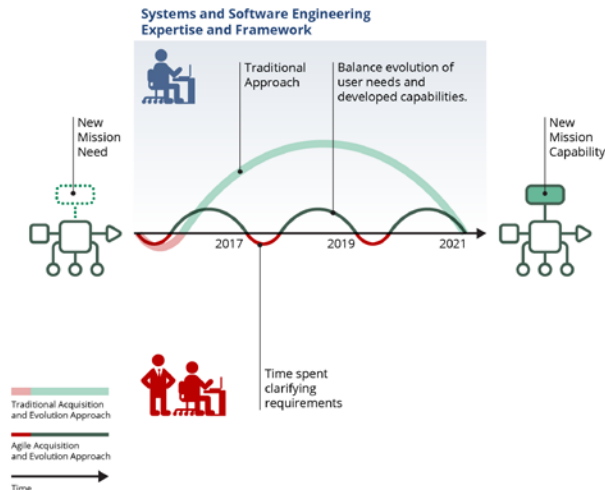
Presenter: Gives us a good understanding of who's here today.

Motivation for Agile: Gov't Acquisition and Innovation

Motivation for Agile: Gov't Acquisition and Innovation

Many regulated environments, like the DoD, NEED innovation and NEED incremental improvements to their systems.

Many of them are now willing to consider changing their approach *if they can do it without getting in trouble with their governing statutes and regulations.*



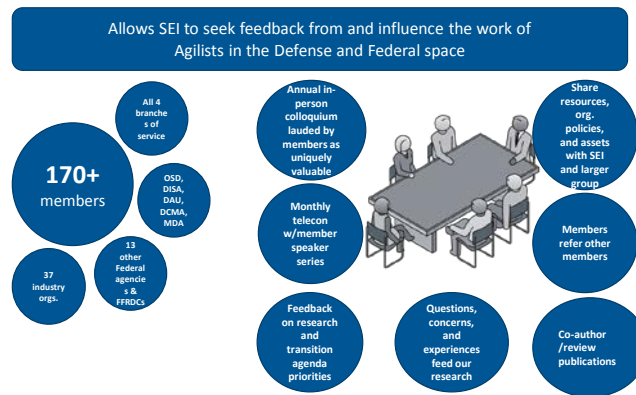
**008 So now that we've talked a little bit about why Suzi's so interested in this and why you're all here today, let's talk about why the government is so interested in looking into Agile methods for acquisition. Historically, in the interest of due diligence and being good stewards of taxpayer dollars, as we identify a need for a new mission or a new end-user capability, we have nailed down all of the requirements and all of the design way up front in the interest of due diligence, and then what we've discovered is as we're building a system, those decisions tend to get stale over time.

Meanwhile, we spend years building to that original set of requirements while the environment, while threats, and while technology changes around

us. So organizations are now seeking to put capability in the hands of end users much more quickly, and that's why all of our, our DoD compatriots and our colleagues, are expressly interested in approaching Agile methods today.

Agile Collaboration Group – Growing since FY12

Agile Collaboration Group – Growing since FY12



**009 The SEI does have an active Agile collaboration group. We have a group of almost 200 participants from across DoD, federal agencies, contractors and academia, and we meet on a regular basis to talk about issues and challenges facing Agile adopters in government settings, and this organization also helps our members help us shape our research agenda. They collaborate with us on developing publications and identifying areas that are really of interest to the broad community for problems that we can help to solve.

Presenter: Yes, and they are very active.

Presenter: They are very active.

Presenter: Very active and vocal, which we like. We like that a lot.

Presenter: And any U.S. person interested in Agile adoption in the DoD or government space is welcome to join the Agile collaboration group, and I think our contact information is available at the end of the webinar--

Presenter: Yes.

Presenter: --for those who are interested.

Presenter: Absolutely.

Interpreting Agile/Lean

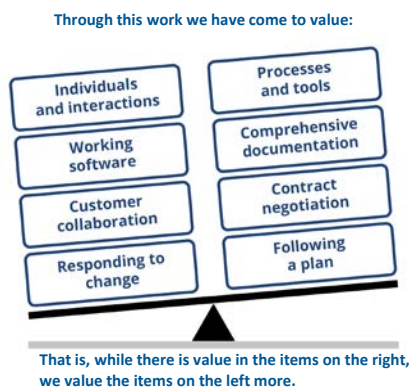


**010 Presenter: So now, now

that we've got sort of a foundation established, we'll talk about what it really means to, what does it mean to be Agile or pursue Agile and Lean in a government setting?

Agile Manifesto—STILL the Basis for Agile Thinking in Industry

Agile Manifesto—STILL the Basis for Agile Thinking in Industry



Common myth:

The manifesto is often misinterpreted to mean:
no documentation, no process, and no plan!

<http://www.agilemanifesto.org/>

**011 We like to say it all still comes back down to the principles that were spelled out in the tenants of the Agile manifesto, but what we've discovered is that our colleagues in the government tell us that historically they have operated in a very rigid environment on the right-hand side of our scale. Lots of projective documentation, a compliance-driven approach, with expensive and resource-intensive processes for managing change, and so shifting to that more collaborative, flexible approach that really values responding to change and constant collaboration. That's a big cultural

shift and it's a big change in the infrastructure that our colleagues are operating in as well. Yeah.

Presenter: And want to definitely point out the common myth, if somebody comes up to you and says they're doing Agile and says that, "Yeah, well, we're doing Agile, so we don't need documentation, we don't need planning and we don't need process," they're pulling the wool, to put it nicely. That is not-- there's more discipline involved in doing Agile well than there is in traditional settings in many ways. So be aware of that, if you haven't run into that already. You may. It's becoming less prevalent, which I'm really glad to see.

Presenter: Right. Right. But you can't sustainably deliver working code on regular short cycles without a disciplined development process.

Presenter: Exactly. Exactly.

Polling Question 2

Polling Question 2

Which of the following groups of Agile Principles would you like us to focus on?

- a. Principles related to requirements and engineering
- b. Principles related to teams and stakeholders
- c. Principles related to small batches and cadence

**012 Moderator: Okay. Our third polling question. Again, we want to maximize your time with us, so we'd like to know, "Which of the following groups of Agile principles would you like us to focus on? A, Principles related to requirements and engineering; B, Principles related to teams and stakeholders; or C, Principles related to small batches and cadence?" We'll give you about 15, 20 seconds to vote. We'll turn it back to you guys while they do so.

Presenter: So I haven't seen any questions yet, but I am going to, I'm going to--

Moderator: Yeah. No questions but a lot of people chime in. We got people from Columbia, Wisconsin, Albuquerque, Indianapolis,

Pittsburgh, France, so we're spanning the globe here today, so...

Presenter: The avenue of France.

Moderator: Yes.

Presenter: That's more French than I can speak.

Presenter: Eh.

Presenter: Guten tag to anyone, anyone joining us from Germany.

Presenter: Who's from Germany? There we go.

Moderator: All right. So we got an overwhelming response; 58 percent would like us to focus on principles related to requirements in engineering.

Presenter: Okay.

Moderator: Twenty-eight percent, principles related to teams and stakeholders.

Presenter: Okay.

Moderator: And 14 percent, small batches and cadence.

Presenter: Small batches.

Moderator: So maybe we can touch on it all somehow.

Presenter: All right.

Presenter: Okay.

Presenter: I can tell you, I'm not at all surprised that requirements in engineering is the one that won.

Useful Interpretation of Agile Principles for Government Settings (1/3)

Useful Interpretation of Agile Principles for Government Settings (1/3)

Agile Principle	Useful Interpretations in Government Settings
The highest priority is to satisfy the customer through early and continuous delivery of valuable software.	In government, the "customer" is not always the end user. The customer includes people who pay for; people who use; people who maintain; as well as others. These stakeholders often have conflicting needs that must be reconciled
Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.	Rather than saying "competitive" advantage, we usually say "operational" advantage. This principle causes culture clash with the "all requirements up front" perspective of many large, traditional approaches.
Deliver working software frequently, from a couple of weeks to a couple of months, with a preference for the shorter timescale.	What it means to "deliver" an increment of software may well depend on context. With large embedded systems, we are sometimes looking at a release into a testing lab. Also, for some systems, the operational users are not able to accept all "deliveries" on the development cadence – because there are accompanying changes in the workflow supported by the software that require updates.
Business people and developers must work together daily throughout the project.	In government settings, we interpret "business" people to be end users and operators, as well as the other types of stakeholders mentioned in Principle 1, since in many government settings, the business people are interpreted as the contracts and finance group.

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

<http://waysandmeans.house.gov/wp-content/uploads/2016/07/20160714SS-Testimony-Hayes.pdf>

**013 So the three principles that I, and this is Suz's personal binning and organizing of these, so you may not agree completely on this, but we'll start with this and maybe move a little further into some of the others. There's certainly the one that's second one on your slide here, "Welcome changing requirements, even late in development." There's another one that is continuous attention to technical excellence and good design, enhances agility, and the best architectures, requirements and designs emerge from self-organizing teams.

So we're going to focus on those first, and we may delve into a couple

of the others as we go. We're not going to go into depth into all 12 of them, because we only have an hour. Now, what you'll see on the table that we've given you is actually from testimony that we provided to the House Ways and Means committee this July of 2016. They asked us to help them understand how should a body like Congress, a high-level body, do oversight of Agile, and they wanted to know about "What is Agile?" So they wanted an explanation of that.

So we used this to help them understand the principles themselves, and the right-hand column is some interpretations that we put together for them to help them understand, "What does this mean in a government setting?" Because some of the words mean different things. Businesspeople, for example.

Presenter: Mm-hm.

Presenter: In a contracted government setting, they're going to think that's the people that write the contracts, where in the manifesto, that was really more about the people that are actually using the software that understand how the software's going to be used. So that's a different thing. Customer. Delivery is another one. Delivery is a word that really gets people in the government going, because in our world, delivery means delivery out to the end user.

Presenter: Right.

Presenter: To the war fighter, and delivery in the Agile context in a commercial setting very well might mean that, but we really can't always interpret it that way in a government setting. One of the things that we often have to do is special kinds of accreditation for security, for air worthiness, for different kinds of regulatory compliance, and so delivery, we may be able to deliver to those environments, but we may not be able to get out all the way to the customer. Now, when you add in concepts like DevOps, and we've done some webinars in DevOps before, we're trying to actually bring some of those ideas in to actually streamline all the way out to, through, those kinds of activities, but that's--we're not quite there yet. But let's just talk first about welcome changing requirements.

So Agile processes harness change for the customers' competitive advantage, and we, in our case, we talk about this being the war fighter's operational advantage.

Presenter: Right.

Presenter: So this is another change. We often talk about concept to capability instead of concept to cash. But the, the challenge here, is what Eileen talked about earlier. Is that we are accustomed to writing a full specification of requirements that we, we agree amongst ourselves are complete, even though in the back room we know they're really not complete and we know they're not

really finished and we know that they're going to change, and so we've got, this is one of the places we need a mindset shift, and that's happening, and one of the ways that that's happening is actually through DoD 5000.2, which is one of the regulatory vehicles that guides acquisition. The January 2015 version of that has given us some models that actually look a little Agile, even though they don't say the word. They are iterative, they're incremental. They're not showing all of the very heavy requirements, reviews and everything up front, and so that kind of change in the regulatory environment is making people a little bit more ready to accept changing requirements. But you've got to plan for it, right? If our change management system is really a change prevention system, then we're not going to have a really good experience with Agile.

Presenter: Right.

Presenter: Let's see, so--and then continuous attention to technical excellence and good design enhances agility.

Useful Interpretation of Agile Principles for Government Settings (2/3)

Useful Interpretation of Agile Principles for Government Settings (2/3)

Agile Principle	Useful Interpretations in Government Settings
Build projects around motivated individuals. Give them environment and support they need, and trust them to get the job done.	A frequent challenge in government is to provide a suitable technical and management environment to foster the trust that is inherent in Agile settings. Allowing teams to stay intact and focused on a single work stream is another challenge.
The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.	In today's world, even in commercial settings, this is often interpreted as "high bandwidth" rather than only face-to-face. Telepresence via video or screen-sharing allows more distributed work groups than in the past.
Working software is the primary measure of progress.	Out typical government system development approaches use surrogates for software – documents that project the needed requirements and design – rather than the software itself, as measures of progress. Going to small batches in short increments allows this principle to be enacted, even in government setting, although delivery may well be a test environment or some internal group other than users themselves.
Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely	This principle is a caution against seeing agility just as "do it faster." Note that this principle includes stakeholders outside of the development team as part of the pacing.

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

**014 When we use these principles--

Useful Interpretation of Agile Principles for Government Settings (3/3)

Useful Interpretation of Agile Principles for Government Settings (3/3)

Agile Principle	Useful Interpretations in Government Settings
Continuous attention to technical excellence and good design enhances agility	This is a principle that often is cited as already being compatible with traditional government development.
Simplicity—the art of maximizing the amount of work not done— is essential.	One issue with this principle in government setting is that our contracts are often written to penalize development organization if they don't produce a product that reflects 100% of the requirements. This principle recognizes that not all requirements we think are needed at the onset of a project will necessarily turn out to be things that should be included in the product.
The best architectures, requirements, and designs emerge from self-organizing teams.	Note that the principle does not suggest that the development team is necessarily the correct team for requirements and architecture. It is however, encouraging teams focused in these areas to be allows some autonomy to organize their work. Another complication in many government settings is that we are often re-architecting and re-designing existing systems.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	This principle is an attempt to ensure that "lessons learned:" are actually learned and applied rather than just being "lessons written"

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.

**015 One of the things we do in

class with these is we ask people to say which of these are compatible or incompatible with current acquisition practice? And this one on continuous attention to technical excellence and good design actually is the one that all, very frequently gets, "We do this. This one is compatible."

Presenter: Right.

Presenter: "We focus on technical excellence and we, you know, it--we just have to do that."

Presenter: Right.

Presenter: "We have too many things riding on what we do not to pay attention to that." So this one is actually well aligned with the way we think about things in our current way of doing things, and is not so big a stretch as some of the others.

The last one in this group, "The best architectures, requirements and designs emerge from self-organizing teams," is another one that you have to sort of parse a little bit to make sure you understand how it can apply in our settings. It's not saying that those self-organizing teams are teams of coders. I've actually had students that said, "Well, you can't let the coders do all the requirements and architecture and everything," and that's not actually what this is saying, although in some settings actually it is the coders that can do that in some smaller teams. But this is basically saying that the people that are responsible for the work, whether

the work is requirements or architecture, design or coding, they need to be able to organize how they do the work.

So instead of you telling Dan and Sally and Tom, "You're going to do the database architecture," and, "You're going to do this," and, "You're going to do the user interface," they decide how that work is going to occur. That gets in the way of a lot of the ways that we tend to parse people out into projects. So this actually has an effect on another area, which is not an explicit principle here but it is in, when we talk about Lean, which is limiting work in process, and having self-organizing teams be able to say what they need to do and how they need to do it kind of prevents people from being attached to too many projects at once, and that's, focus, is one of the things that helps Agile.

Eileen, are there any, maybe one or two of the others that you want to talk about, out of the teams and collaboration that we know give people problems when it comes to Agile in government?

Presenter: So I sort of spanned this across the requirements and engineering bucket and the small batch bucket, but the idea of simplicity. Maximizing the amount of work not done. This is one that creates some legitimate heartburn in the government environment, because people aren't used to not having those projective documents,

not having the design document up front completed before you do the design, and people confuse this maybe due to bad experiences in the past or just a lack of experience with these methods. But people confuse this with not documenting anything or doing as little work as possible.

Presenter: Right.

Presenter: And really what we need to be focusing on here is we're doing work that adds value.

Presenter: Value. Yeah.

Presenter: And we're not doing things that don't explicitly add value to some aspect of our program, and we'll talk more about what it means to add value to all of those stakeholders I think in a few slides.

Presenter: Yeah.

Presenter: That's one of my favorites to talk about.

Presenter: And that's the one that probably, if there's a single principle that in class people say, "This is not aligned with our current acquisition," that's the one.

Presenter: That's the one.

Presenter: Because as soon as they see the maximizing the amount of work not done, everybody thinks about, "Well, how can I pay a contractor if they didn't do the work for me?" right?

Presenter: Right.

Presenter: If they're in a contracted environment, "How could I get paid if I didn't do the work?"

Presenter: Right.

Presenter: And so this, this principle in particular gets us into the idea of prioritizing what's important and making sure we get that 80 percent of stuff that we can get done and the 20 percent that we maybe can't get done with the money that we have that is not as important. Make sure that 20 percent doesn't get into the things that we're really working on right now.

Presenter: Absolutely.

Presenter: So all right. So as you can see from our discussion, there's lots to say about each of these and one of the other resources I'll point you to related to principles in government is a podcast series that, an audio podcast series that Mary Ann Lapham and I did, about, ah, two years ago now, where we took each principle and did a little podcast, 110 minutes or so, about what we've seen in government related to that. So that podcast series is referenced at the back of the webinar, and we welcome you to go out and listen to that when you're in the car or otherwise looking for something to entertain you.

What are the Agile tenets and principles trying to enable?

What are the Agile tenets and principles trying to enable?

1. Increase speed of development of high business value software
 - Speed=the time it takes to move ideas from one person to another
 - A reason that all Agile approaches emphasize face-to-face contact as much as possible
2. Take advantage of the benefits of small batch sizes
 - Fast feedback, fast learning, etc
 - What is the next smallest thing I can do that increases the business value delivered?
3. Reduce non-value-added work and rework
 - Reduce “time to fielding”
 - BUT understanding what is truly non-value added
 - Non-value for designer may be HIGH value for sustainer

**016 Moderator: Suz, could we take a question now before we get too many in the backlog?

Presenter: Sure, absolutely.

Useful Interpretation of Agile Principles for Government Settings (3/3)

Useful Interpretation of Agile Principles for Government Settings (3/3)

Agile Principle	Useful Interpretations in Government Settings
Continuous attention to technical excellence and good design enhances agility	This is a principle that often is cited as already being compatible with traditional government development.
Simplicity—the art of maximizing the amount of work not done—is essential.	One issue with this principle in government setting is that our contracts are often written to penalize development organization if they don't produce a product that reflects 100% of the requirements. This principle recognizes that not all requirements we think are needed at the onset of a project will necessarily turn out to be things that should be included in the product.
The best architectures, requirements, and designs emerge from self-organizing teams.	Note that the principle does not suggest that the development team is necessarily the correct team for requirements and architecture. It is however, encouraging teams focused in these areas to be allows some autonomy to organize their work. Another complication in many government settings is that we are often re-architecting and re-designing existing systems.
At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.	This principle is an attempt to ensure that “lessons learned:” are actually learned and applied rather than just being “lessons written”

Source: SEI Congressional testimony July 14, 2016 to House Ways and Means Committee.



**015 Moderator: Just one that came in the chat said, "How do we reconcile the idea of milestones or program reviews such as PDR, CDR, TRR, et cetera, in Agile development? Most government programs require value based on schedules, based on major program milestones."

Presenter: Right. So that is one of our top three questions--

Presenter: Absolutely.

Presenter: --that we get when we talk to people about this. The other two, one of them is, "How do I deal with technical reviews?" Another one is, "How do I contract and incentivize this correctly?" and I can never remember the third one until it comes up, but there's another one. So technical reviews, we actually

have a whole technical note, of which Eileen is--

Presenter: Yeah.

Presenter: --the primary author. And it's actually titled Technical Evaluation is how you would find it in our digital library. But what we talk about there is the idea of using incremental reviews. So when you think about Agile, we're talking about incremental deliveries, incremental planning and we're talking, in many cases, around a quarterly cadence is what a lot of people use, an 8- to 12-week cadence. So it's a two- to three-month cadence, and in that time frame, you do something substantive, and that's feasible, to do something substantive with several teams in that time frame, and so that's the time frame around which you're looking at PDR and CDR kinds of things. What many people do is they have their PDR and CDR checklist that they would use in a normal setting and they adjust those for incremental kinds of deliveries.

Presenter: Right.

Presenter: And they actually talk about. So there will be a point at which they will, they will have a, what we call, a definition of Done that will include some elements of the PDR checklist or the CDR checklist or even a test readiness checklist, and so they include those, and that evidence piles up and at some point you can decide, usually there is a decision, that we need to

have a culmination, a capstone kind of event--

Presenter: Right.

Presenter: --where we summarize all of this information. But from a program office viewpoint, what you've been doing, if you've been paying attention and monitoring the work all along, your confidence in what's been done is much higher. You know what's been done. You know what hasn't been done in a much more substantive way than you do when you look at a slideshow, you know, of PowerPoint decks that are 300 slides long talking about an architecture that nobody's implemented yet at PDR, for example.

Presenter: I like to say that this makes PDR and CDR almost anti-climactic.

Presenter: Yes, yes.

Presenter: Because you've watched everything grow for the whole time.

Presenter: Yeah.

Presenter: There shouldn't be any surprises embedded in 300-page decks.

Presenter: Yeah. Now, the caveat there is you've got to plan for it that way. You know, this goes back to your acquisition strategy, this goes back to how you do source selections. This goes back to how

you plan your CDRL deliveries and all those kinds of things, and if you don't plan for Agile or at least plan what we call Agile-aware kinds of RFPs then you will have milestones that are set in stone in this date and you'll have a lot less flexibility.

We do know of settings, government settings, where they started out in a very traditional way and someone decided that Agile was a good way to go in the middle and they made some adjustments. Sometimes that involved tailoring and waivers, sometimes it involved not starting Agile really until after PDR or after CDR in one case, and just dealing with some of the, they got some of the benefit but they really couldn't get all the benefit. So there's a lot of different ways to deal with that, but we see this struggle is a struggle. Once you get through it, it's--actually can, as Eileen said, it can be anti-climactic, and that, that's a good thing.

Moderator: Right.

Presenter: Okay. Okay.

Presenter: So moving on.

What are the Agile tenets and principles trying to enable?

What are the Agile tenets and principles trying to enable?

1. Increase speed of development of high business value software
 - Speed=the time it takes to move ideas from one person to another
 - A reason that all Agile approaches emphasize face-to-face contact as much as possible
2. Take advantage of the benefits of small batch sizes
 - Fast feedback, fast learning, etc
 - What is the next smallest thing I can do that increases the business value delivered?
3. Reduce non-value-added work and rework
 - Reduce “time to fielding”
 - BUT understanding what is truly non-value added
 - Non-value for designer may be HIGH value for sustainer

**016 Presenter: Yeah, let's move on.

Presenter: So I like to summarize the things that we're trying to get out of these principles in the government. To help people focus on what it is that they really enable. They enable us to increase the speed of development, of high business value or high operational value software, and I like the definition of speed as the time it takes to move ideas from one person to another.

Presenter: Yeah.

Presenter: If we can increase that or we can decrease--

Presenter: Decrease.

Presenter: --the time, sorry, that it takes to move ideas around, then

you get a much more dynamic and you get a much richer set of ideas to work from in an engineering setting. We want to take advantage of the benefits of small batch sizes and they are enormous. Moving to small batch sizes changes a lot in how everything works. It's probably the single biggest shift in how people work and in what outcomes you can get, and I like to encourage people to think about, "What is the next smallest thing I can do that increases the business value of what I'm delivering?" And that just gives lots of clarity. You know, I've got this to-do list of a hundred things. What's the next smallest thing I can do that increases the business value of what I'm delivering?

All of a sudden, 90 of those things just drop right off the list, and it's a very powerful, powerful concept, and the idea of reducing non-value added work and rework. The rework part actually comes from the incrementalism and the iteration, because I'm learning as I go, and so I will have rework when I'm doing that. I'm going to learn something at week four and I've already worked on this piece of software in week one and I'm going to have to do some rework. Now, I don't consider that to be non-value-added because the rework came from a learning that was needed that I couldn't have in week one. The non-value-added rework that we're really dealing with here is the rework related to assuming, projecting, we call them projective documentation, projecting

all these wonderful ideas about what we could do without having any implementation basis for it and we just don't know what we don't know.

So those are the three things that I take away from the principles terms of what are they trying to do for the government settings? And that's one of the reasons I think Agile is relevant to our government settings today.

Presenter: Do you want to talk a little bit about this bottom bullet, that we tend to, we think is really important that I think skates by sometimes. The idea that understanding that the value-add applies to not just the primary end user stakeholder.

Presenter: Yeah.

Presenter: In the DoD environment, we're looking at systems that are operating 10, 20, 30 years even beyond their original projected lifespan. Sustaining those systems takes a different color of money, as we talk about in the government, and it's a lot harder to sustain something with a tightly coupled architecture. Things get stale.

Presenter: Yeah.

Presenter: At some point we need to patch them, we need to upgrade operating systems, and architecting with those things in mind, Agile lets us do that because the way we have to behave in a small batch almost buys us that for free.

Presenter: Pretty close. Yeah, and that's the thing is non-value added. So, you know, is sustainment documentation, is an as-built architecture diagram non-value added?

Presenter: Right.

Presenter: Well, to the designer it might be, because they're thinking, "Well, I've done it now, so what do I need it for?" But that sustainer who's going to have to come back when they want enhancements. They need that. They don't need the projected one. They don't need the to-be architecture, but they need that as-built.

Presenter: Right.

Presenter: So a lot of the value-added documentation in particular for Agile comes with thinking about sustainment and making sure that we address the needs of that, that community as well.

Going Beyond Small Teams Adds Complexity

Going Beyond Small Teams Adds Complexity

Lots of things beyond supporting a small team that you have to focus on when scaling above a few small teams, regardless of software or systems context:

- Managing the interfaces among the many products/system components that multiple teams are working with...
- Figuring out how to synchronize releases and events across multiple teams...
- Figuring out how to get the inventory (backlog) of requirements organized productively to support the development pace of multiple small teams....
- Dealing with specialty disciplines (UX, security, etc.) that have significant inputs to the evolving product, but aren't needed as full time team members....



**017 Presenter: Okay.

Presenter: Now, the Agile principles are really about small teams, right? The Agile manifesto was conceived by a group of software methodologists who were looking at, "What makes projects successful?" and in many of their experiences, small projects is what were successful. So they really focused on what are the things that make small teams work? Good, bad or indifferent, that's not our world in the government. We have very few small projects. We have lots of big projects, and so we have to go beyond those principles when we start doing what we call scaling, and we have to address lots of different things. Managing interfaces, synchronization, how to get a large inventory of requirements organized,

dealing with specialty disciplines,
especially security in our world.

So, you know, all of these things lead us to have to add some other ideas besides the Agile principles to get things to work when we've got not one or two teams but we've got 5 or 10 or 20 teams that are trying to achieve a large goal, and so that's where we get into looking at Lean principles.

SAFe is Also Principles Based – Lean Engineering and Startup Principles

SAFe is Also Principles Based – Lean Engineering and Startup Principles

#1 - Take an economic view

#2 - Apply systems thinking

#3 - Assume variability; preserve options

#4 - Build incrementally with fast, integrated learning cycles

#5 - Base milestones on objective evaluation of working systems

#6 - Visualize and limit WIP, reduce batch sizes, and manage queue lengths

#7 - Apply cadence, synchronize with cross-domain planning

#8 - Unlock the intrinsic motivation of knowledge workers

#9 - Decentralize decision-making

<http://www.scaledagileframework.com/>

**018 There's a set--Lean has lots of expressions of principles. The ones that we like to use come from the Scaled Agile Framework, because they, the Scaled Agile Framework, whether you want to, you know, build us it or not, is Agile aware, and it's lean and it's very Lean focused, and so these principles are ones that can help us to understand what some

of the things are when we go to scaling. Please.

Presenter: But the one thing I always like to talk about when we talk about lean engineering and startup principles too is this idea of this first principle where we talk about taking an economic viewpoint, and we're really talking about what-- these principles tended to be developed for commercial firms. But when we talk about economics we're really talking about whatever is valuable to our end users, to our stakeholders.

So just something to keep in mind from a government perspective.

Presenter: Right.

Presenter: We're not talking about managing to a profit and loss but to what really drives us value.

Presenter: And that's actually the very first one, "Take an economic view," is really about that.

Presenter: Yeah.

Presenter: What is value and how do we make sure that the value that we need, which is not necessarily dollars, right?

Presenter: Right.

Presenter: It's about what's the actual value to our end users? What's the value to our stakeholders? What's the value to Congress, in

some cases, who are the ones that funded this work? So these are things, if you know the lean sort of engineering arena, you've probably seen before.

I want to--based on the question we had earlier--I want to highlight number five, "Base milestones on objective evaluation of working systems," and this is the difference between our traditional technical reviews and this lean version of the world, because milestones, we do use milestones in Lean and Agile. But they're milestones that are based on evaluating the working systems, not evaluating the projected requirements, the projected architecture, the projected designs, and so those kinds of things are what, what we're really--that's the big difference is we're looking at things that are actually built.

I also mentioned visualizing and limiting work in process. Trying to get people focused on just doing the thing--

Presenter: Right.

Presenter: --to finish, you know. One of the sayings is, "Stop starting; start finishing," and I say that to myself several times a day, and, you know, and sometimes it works. There's others in here that are also counterintuitive to the way we like to work today. Number three, "Assume variability and preserve options." We're used to kind of locking things down.

Presenter: Yeah. Locking things down.

Presenter: And--

Presenter: And not realizing that the environment changes around us.

Presenter: Exactly.

Presenter: Our users learn about new things they need, new technologies are developed.

Presenter: And so this idea of making decisions at the last responsible moment is something we have to learn to do, and that's a different way of designing and implementing than in our past.

These are covered in the Scaled Agile Framework online resource. We also cover them in our "Leading SAFe Agile in Government" course in a lot of depth, because they do guide a lot of systems engineering, not just the software related to Agile.

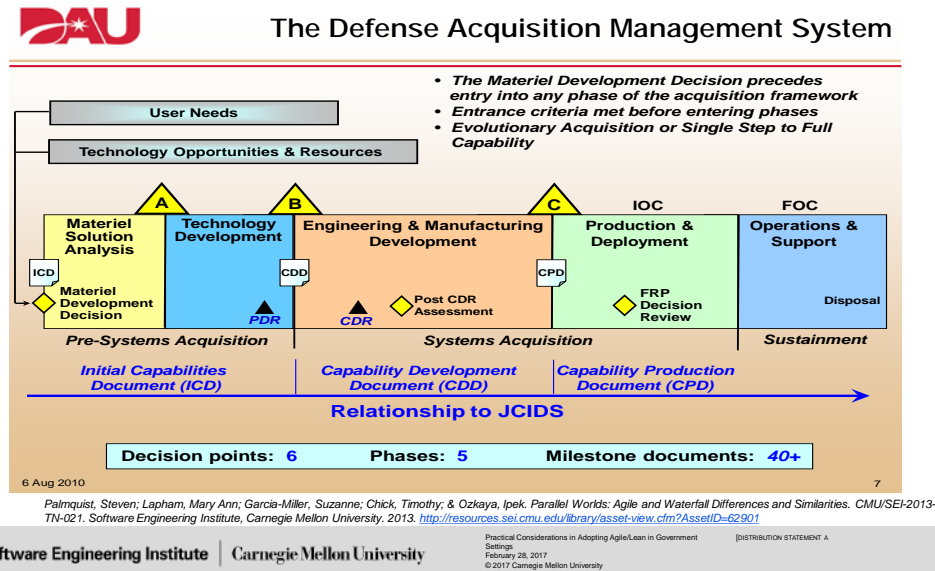
The Eco-System of Agile in Government



**019 But next we need to talk about the eco-system of Agile in government, because Agile software teams especially tend to live within these larger programs.

Agile Methods are Used in Context of a Larger Program (More Often Than Not)

Agile Methods are Used in Context of a Larger Program (More Often Than Not)



**020 Presenter: Right.

Presenter: And Eileen's going to talk about that.

Presenter: Right. So the slide that you're looking at now is actually a graphic representation of the Defense Acquisition Management System, and so as, so in the DoD and then again in the federal government, we are stewards of taxpayer dollars, and this system is built to ensure that at every step of the way we are making responsible decisions with those dollars and it's set up so that we review. We have these phase gate milestones at, documenting, the requirements, defining the design, and they're sort of lock-step as you have to complete one to go into the next stage, and that is, that is I don't want to stay antithetical, but it's a

culture that is very different. The behaviors that are locked in as a result of this tip, the traditional guidance, there's--is very locked in and it can be difficult to make the shift to, "How does this accommodate incremental and iterative delivery?" The question that we had earlier about PDRs and CDRs really drove to this.

Presenter: Right.

Presenter: So Suz mentioned earlier that the new January 2015 5000.2 specifically accommodates some new models that look to iterative and incremental delivery. You won't see the word Agile, but we're talking about delivering in small batches, tailoring where allowed by the regulation. But we recognize that using Agile, even with incorporating these, you know, periodic deliveries, smaller batches, that doesn't excuse us from compliance--

Presenter: Right

Presenter: --with the regulatory and statutory boundaries that we operate in.

Components of the Eco-System

Components of the Eco-System

Components include but not limited to:

- Program Management Office
- Contracts
- Finance
- System Engineering
- User (end, interfacing system(s) users, etc)
- Stakeholders
- Information Assurance
- Supply Chain
- Developers
- Testers
-



**021 And that's, that's where, predominately, that's where our team works, is helping to build those crosswalks and build those bridges so that teams operating in this environment can take advantage of the benefits of small batch.

Presenter: Right.

Presenter: The risk buy-down, the ability to collaborate more and be more flexible and drive capability delivery further to the left. One thing that we have to realize is that so the Agile manifesto started talking about software development, but this is not just a software engineering problem or a set of software engineering issues. The eco-system involves everyone, from the program management office, who's responsible for actually getting the

system built or getting the software built, purchased, to the contracting and finance business functions, the system engineering and integration functions.

But the user, the end users, end users' interfacing systems, those are all, those are critical, and that, the level of collaboration that's required in order to be successful with these rapid small batches, we can't really say enough about the need for subject matter experts who are empowered by the organizations that they represent to make the decisions in their swim lanes in a very short amount of time. Two-week development cycles or quarterly release cycles don't work if the end user representatives who are providing the voice of the customer have to keep going back to different parties to get approval to voice an opinion on a requirement, for example. So that's, that's a matter of having not just the people available, but having the right people engaged and having them empowered.

Other stakeholders in the DoD environment in particular, we're dealing with our information assurance and cybersecurity functions. Those are folks who have to certify that a system, you know, they have to check the boxes to make sure that it's resilient and it's reliable in order to grant it authority to operate on a network. Figuring out how those folks interact with an Agile development process is, has, been a long learning process, and

then everyone else in the supply chain. Multiple contractors. If you're, you know, if you're building an airplane or if you're building a ship, you have hardware, you have other suppliers in your ecosystem, and you have a variety of development and independent test organizations as well.

In order to achieve the benefits that we're talking about, those faster cycle times and that improved quality, that buy-down of integration risk, everyone needs to understand how the development lifecycle works and how it changes the way they do their job.

Presenter: Exactly, and most of these other stakeholders that we're talking about, their lifecycle has been built against the traditional model, against big batch, and you're going to give me a big batch of stuff to test. You're going to give me a big batch of stuff to certify, so that, that's one of the reasons I say that small batch is probably one of the biggest concepts in this because it changes how everyone works.

Before we move on to our gimmees and gotchas, there's a couple of comments, questions I wanted to address.

Presenter: Okay.

Presenter: Somebody was saying that when I said, "Stop starting; start finishing," you know, "That's Kanban." You're absolutely right.

Kanban is a just-in-time, pull-based method that comes out of the lean engineering, lean manufacturing community, and is very compatible with Agile. As a matter of fact, you will hear in the larger Agile community the term Scrumban, the use of Kanban kinds of activities within a scrum.

If you look at things like the Scaled Agile Framework, at the team level they actually have kind of an "or" statement of terms of doing scrum, which is a time-box based team method, or Kanban, which is flow-based, and especially with systems engineering in some of the other communities that we work with, we see Kanban being a very effective method for helping them to move to small batch and to instantiate these kinds of principles, and there was one about federal--this is going to be very interesting to the federal government because of cost, cost cutting--and yeah, and there are a couple of resources.

If you're in federal government, I'll point you to one of them is the U.S. Digital Service, which was chartered by the White House and is doing all kinds of projects with federal civil agencies that mostly involve Agile and Lean. There is now a Defense Digital Service and an Air Force Digital Service and an Army. I don't know about the Navy yet, but I have to believe that there will be before too long. Was there anything else that I-let's see.

Presenter: There was a question about incorporating some specialty disciplines like human factors--

Presenter: Oh, sure.

Presenter: --and HCI.

Presenter: Yes, and--

Presenter: That's a good question.

Presenter: Yeah, and that, those are, stakeholders certainly that belong in this list. They're in the dot-dot-dot right now. But, you know, whether it's a data analyst, whether it's a--the new term is not UI, it's UX, user experience.

Presenter: Yes. Showing my age.

Presenter: Or HCI or, you know, all of those kinds of specialties have a role. They are, depending on the method you're using, we often think of them as a shared service because you don't need them necessarily for a whole six months. You need them to do work for you to help you get through a particular piece of the puzzle, and you have to account for them, but they need to be part of the planning. They need to be involved so they know when and how they can be of use to the project, and they are probably the kinds of resources that will actually be shared among several projects, because they are very special, specialized and don't tend to have, be doing, kind of yeoman's work the whole time that they're involved with a project. I think--

Moderator: We actually had another one from Mary too--

Presenter: Yes.

Moderator: --while we're here. It says, "How do you resolve needing stable teams with gaps in funding? Many times a team works on a project at a prototype stage but has a funding gap before it can work to transition to it fully. How do you fund a team and keep it together until the next increment of funding comes in?"

Presenter: So this comes back to where is the focus of your Agile effort? Is your focus of Agile really just at the team level? Is it at what I'll call the project or the program level or is at the enterprise level? And my experience is that until you get Agile, Lean kinds of concepts to be accepted at the enterprise level, the problem you described is not going to go away and is not going to be easy to solve. Because there's not--until you get to the enterprise level, people still conceive of things as project.

Presenter: Right.

Presenter: They don't conceive of things from a flow view of, "There's stuff that we're going to need for this. I'll use, you know, there's an airplane." Well, you know, as long as that airplane is flying, there's going to be a need for things to happen. There are going to be monetizations, there's going to be upgrades and

things like that. How we organize that work currently is we, 12 months ahead, we try, 18 or 5 years, you know, we get the budget for a project, right? Which we call a program in the government, and then that, that has a bounded finish, and when it's done then, you know, if we don't have another project then what are we going to do next? Agile and Lean think of things in terms of more of this flow of, "What are the things coming in, what are the things we're getting done?" and it changes how you view contracting, it changes how you view working with your PEO, if you're in a government program, your executive office.

We are starting to see some program executive offices that are starting to make a shift to more of a flow-based system for how they treat the requirements that they're getting from users in how they build their material solutions, and in the IT world, as opposed to perhaps the weapons systems world, there's a concept called an IT box that comes out of the joint requirements kind of facility that looks at, you know, setting some boundaries, and as long as you stay within those boundaries you can sort of continue the work. So there are some things that are starting to change, but, you know, what--there's a term of bureaucracy hacking that has become popular and there needs to be a lot more of that before that problem is solved.

In the interim, you do what you've always done. You basically try to

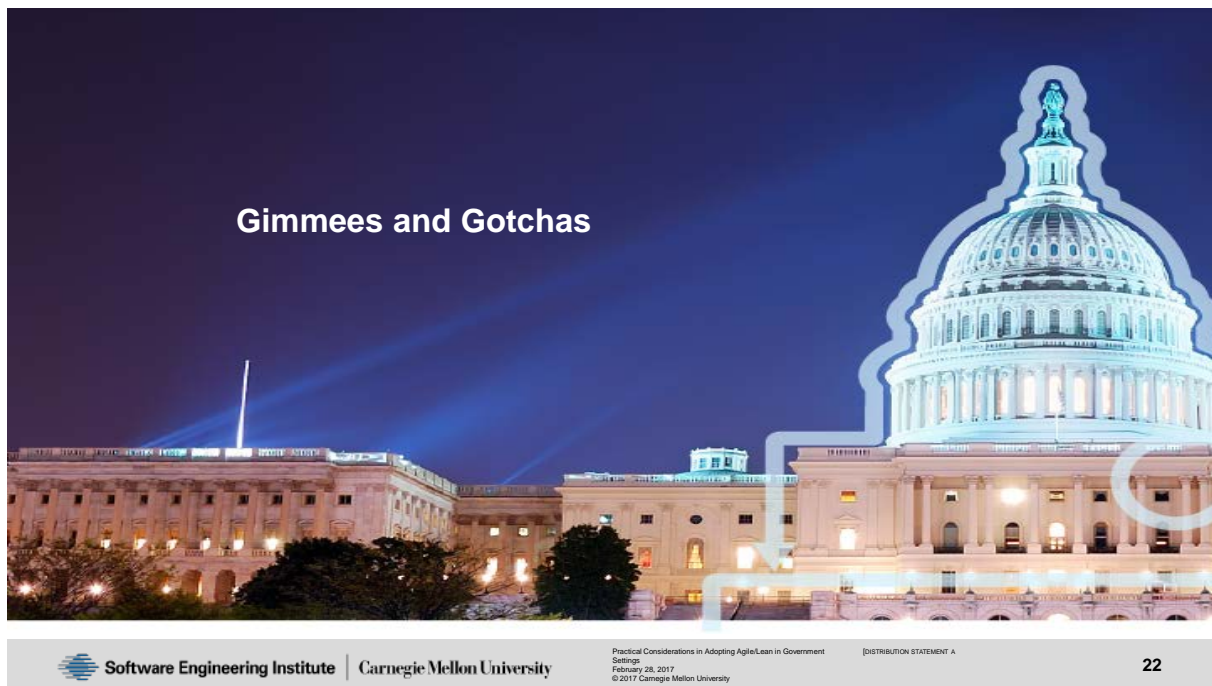
make sure that the people that are working on these projects have got other work that is useful to be doing and that can be funded in some other way. The hard part is getting them back when the next increment of that project comes, and I don't have a local solution for you. Different people have solved it different ways. That was probably longer than you wanted to hear about that.

Presenter: That was a very good question.

Presenter: Yeah.

Presenter: It was a hard question, but it was a good question. So now, why don't we-

Gimmees and Gotchas



**022 We'll do some fun stuff.

Presenter: Okay.

Presenter: We'll talk about gimmees and gotchas.

Presenter: Sure.

Polling Questions 3& 4

Polling Questions 3& 4

Are you working in a government setting that has used Agile or lean methods in the past and they were considered successful at helping you meet your acquisition's or system's goals?

- a. Yes
- b. No
- c. Not Sure

Are you working in a government setting that has used Agile or lean methods in the past and they were considered unsuccessful at helping you meet your acquisition's or system's goals?

- a. Yes
- b. No
- c. Not Sure



**023 Moderator: Okay. Before we get to the fun stuff we're got two back-to-back polling questions which I'll just launch, and while they're voting, one more question I think is somewhat--

Presenter: Okay.

Moderator: --quick and then we can get that. So the fourth polling question's on your screen now. It's, "Are you working in a government setting that used Agile or lean methods in the past and were they considered successful at helping you

meet your acquisitions or systems goal?"

While people are voting on that one, a question from our friend, Hassan, wanted to know, "how to handle constant changes on agree requirements." For example, between contractors and integrator and end user.

Presenter: So constant change is relative. Let's start with that. So if you're changing requirements in between iterations or in between releases, that's actually expected in Agile. We're reprioritizing based on what we've learned and we expect that we're going to have a slightly different direction going forward than what we started. If on the other hand what is happening is inside an iteration, which a nominal iteration is two weeks, in many environments, you know, if inside of that two weeks people are coming to the developers and saying, "I want you to do this instead of that. This is more important to me today," that's not acceptable.

Presenter: Right.

Presenter: And as a matter of fact, the length of an iteration, one of the ways that you decide how long you want an iteration to be, is by how long your stakeholders can tolerate not changing your requirements, right? So if I come to you and I, "Eileen, I really need this new thing in my, in the package." If Eileen asks me--

Presenter: "Can you wait six months?"

Presenter: "No. I can't possibly wait six months."

Presenter: "Can you wait two weeks?"

Presenter: Different answer, right? Can I wait two weeks? "Yeah, I can wait two weeks." Right? And so that, that's the difference, and that's part of what the small batch gives you, is it gives you that ability to plan change and plan change based on learning. Learning from implementation. So that's the thing that you have to guard against. Now, there's roles typically in us, in many of the methods like scrum, a product owner is the person who's authorized to bring requirements into the iteration, into the team, for prioritization. They've done prioritization and they're going to be the ones that protect the team from everybody should go to the product owner before they come to the team in terms of new requirements.

The scrum master, who's the process facilitator for the team, is the other protection. Somebody comes to me and says, "I need you to do this," I'm sending them straight to the scrum master and the product owner. You know, you're not on my list right now. I love you, but you're not on my list and, you know, so you've got to go talk to these guys and we have to figure out how to get it in the backlog. There are always conditions

of emergency, you know, when something has to change and it has to change right now. We figured out, you know, some security vulnerability or something that is going to turn the world upside-down. Those are rare. But we do have lots of people that think that their change is the one that's always the emergency, and that's not always the case.

Presenter: Right, and defining those classes of exceptions is part of how you define your--

Presenter: Yes, that's right.

Presenter: --Agile development process--

Presenter: Your backlog. Yeah.

Presenter: --and your backlog.

Presenter: Yeah.

Moderator: And so the fifth polling question is launched now. If we can get one more quick question. If it's not quick, we can come back to it then.

Presenter: Okay, that's fine.

Moderator: But a good question from Paul asking, "Are government acquisition managers becoming more accepting of Agile techniques?"

Presenter: Yes.

Presenter: Yes, absolutely.

Presenter: We spoke, Eileen and I spoke to a group of generals and senior executive service about three weeks ago, so these are policy level people in the Air Force, and their question was not, "Should we be doing Agile?" Their question was, "How do we institutionalize this?" and that, that question, is a different question than we heard five years ago. So that is happening. The other thing that's happening is DAU is starting to incorporate, the Defense Acquisition University, is starting to incorporate material about Agile in their courses. So people who are being trained in how to be acquisition program managers, depending on which courses they get, are actually getting some training in how to deal with Agile. So that gives them more confidence that they actually can do this in a way that is going to be useful.

There's also, I'll just advertise for DAU for a second, there's an Agile 101 course that we're in the final stages of reviewing called CLE 76. That's an online course that anybody that has access to DAU will be able to take online. So the resources for the government in learning about Agile and how it can be used in their settings and what changes they have to make to their way of overseeing and planning their projects, is definitely becoming more available.

Moderator: Okay. Now we can move on.

Presenter: Okay.

Gimmes & Gotchas

Gimmes & Gotchas

Gimmes are a list of behaviors that give you confidence that your program office and/or contractor is embracing an agile process.



Gotchas are a list of behaviors that may indicate problems currently exist or on the horizon in your agile program.

These Gimmes & Gotchas **are not** intended to be all inclusive nor are they a checklist. The goal of these is to help identify areas to investigate further and focus your energy toward a successful program.

**024 Presenter: Okay. What was--what were the percentages?

Presenter: Oh, yeah.

Moderator: Oh, I'm sorry. Okay.

Polling Questions 3& 4

Polling Questions 3& 4

Are you working in a government setting that has used Agile or lean methods in the past and they were considered successful at helping you meet your acquisition's or system's goals?

- a. Yes
- b. No
- c. Not Sure

Are you working in a government setting that has used Agile or lean methods in the past and they were considered unsuccessful at helping you meet your acquisition's or system's goals?

- a. Yes
- b. No
- c. Not Sure



**023 Presenter: On successful versus unsuccessful?

Moderator: Okay. So I'm going to show them now, and preview results. We had 44 percent "yes," 35 percent "no," 21 percent "not sure."

Presenter: On the first one? Okay.

Moderator: On the first one. Second one was--

Presenter: Okay.

Moderator: --30 percent "yes," 46 percent "no," 23 percent "not sure." And this--so Hasan who asked that question, some of these people are saying, "Yes. We've been successful with this." So it's happening out in the world, and those are numbers I

wouldn't have gotten five years ago either.

Presenter: Yeah.

Presenter: As a matter of fact, when we started this research back in 2009, 2010, the first set of interviews that we tried to do answering the question, "Can we use Agile in government?" we had people that didn't want to talk to us because they thought that we were going to write a report that would quash what they were doing, you know.

Presenter: Right.

Presenter: And it took a while before they would actually tell us what they were doing, so--

Presenter: Sort of stealth Agile.

Presenter: Yeah, yeah.

Presenter: Was what we call it.

Presenter: So that, the whole, it's become a whole different world from that viewpoint. Okay.

Presenter: Okay.

Presenter: So let's talk about--

Gimmes & Gotchas

Gimmes & Gotchas

Gimmes are a list of behaviors that give you confidence that your program office and/or contractor is embracing an agile process.



Gotchas are a list of behaviors that may indicate problems currently exist or on the horizon in your agile program.

These Gimmes & Gotchas **are not** intended to be all inclusive nor are they a checklist. The goal of these is to help identify areas to investigate further and focus your energy toward a successful program.

**024 Presenter: Let's talk about gimmies and gotchas.

Presenter: Gimmies and gotchas. I love these.

Presenter: Yes. So Gimmies are confidence indicators, maybe enablers, things that give you some confidence that you're on the right path if you're looking to pursue Agile or if you've started down the path of Agile, and gotchas are warning signs, if you will. Maybe potential potholes that indicate there might be a problem, there might be a risk area. So we are not trying to build a checklist of everything that must be in place or not in place. We certainly don't want these to be viewed that way, but based on our experience working directly with programs, based on the vast experience of our

collaboration group participants, and feedback from the industry and government, these, these signs, if you will, come up over and over again. So our next polling question is about--wasn't it about gimmees and gotchas?

First Steps Gimmees

First Steps Gimmees



- Your motivation, trade-offs, benefits, and expectations for using an Agile approach are clearly understood and communicated
- There is an explicit understanding that the requirements are expected to evolve
- Automated testing is planned for and budgeted
- Contract and CDRLs allow flexibility and incremental delivery
- Entire program team is aware that the Jan 2015 DoD 5000.02 has new lifecycle descriptions that support more "agile" approaches
- Hindrances and enablers for agile implementation are acknowledged
 - paths to success are identified, preferably through a method like SEI Readiness/Fit Analysis

**025 Presenter: No.

Presenter: No. Okay.

Presenter: Nope. Changed our minds.

Presenter: All right. Changed our minds.

Presenter: Changed our minds.

Presenter: Changed our requirement.

Presenter: All right. So we have these organized as First Steps, Readiness and Implementation. First Steps is really about when you're thinking about getting going and what are some of the things that will enable you to have a good start, as it were.

Presenter: Right.

Presenter: And a couple of these that I want to point out is that the number two, "There's an explicit understanding that the requirements are expected to evolve." That understanding by all the stakeholders that are going to have to approve requirements, approve changes to requirements, is really important, and then there's my favorite.

So automated testing is planned for and budgeted. So anybody who knows Suzanne Miller, who used to be Suzanne Garcia, who was a process improvement, you know, person from way back and you've heard Suzanne Garcia say lots of times, "You have to understand your process before you start putting the tooling into it." This is my exception to that rule.

Because automated testing is probably the one technology that made Agile possible in terms of besides practices, because it allows us to robustly test over and over, do regression testing, build automated test suites, that can do lots of different kinds of testing. Not everything, but they can do a lot of

testing that allows us small batches to evolve safely, and if you haven't figured out how you're going to do your automated testing and you haven't budgeted for it, then you're going to find yourself in a place where--and this happened to a real program--where it took longer to do the regression testing than it did to do the two-week iteration, and at that point they had to stop the project, pull back, build an automated test harness, populate it before they could get started again.

You don't want to be that, that project. You don't want to have to do that to your users. So think about that really, really early. It's the only sort of techno-tool thing that I advocate for getting going early on. You can do a lot of stuff with Post-its on the murals and things to manage the work, but doing manual testing is going to kill you.

Presenter: We should probably move on to some gotchas for First Steps.

First Steps Gotchas

First Steps Gotchas

- Senior managers and stakeholders reluctant to use agile or are unaware that Agile is in use
- Software development is constrained to a hardware-based Work Breakdown Structure
- Mindset that document completion equals progress is prevalent
- Program exhibits a risk averse culture
- Integration testing isn't planned to occur until just before final delivery
- Testing isn't budgeted until much later in the program
- Agile is being treated like a silver bullet



**026 Presenter: Oh, all right. So the flip-side of that is a gotcha, I suppose.

Presenter: Right.

Presenter: Let's see. What are some of your favorites on this one?

Presenter: Program exhibiting a risk-averse culture. Agile isn't actually inherently more risky than our traditional lifecycle approaches, but it's a fundamental shift in the way everybody has to do business, and we know that change is difficult for organizations. Oftentimes there's a little bit of stumbling at first as everybody's figuring out what they need to do, and because we get those benefits of small batch where we learn fast, we know right away if we've gone down the wrong road with something.

Presenter: Right. That's right.

Presenter: And we don't want to have a culture of beheadings when we learn that first thing after a two-week cycle and we have to scrap something or pivot and go a different direction. Agile actually lets us buy down risk, buy down integration risk, buy down design risk, but a lot of times, because it's a new way of business, it's seen as risky and if people aren't willing to collectively as an organization take the steps to try something new, that can be, you can be like Sisyphus pushing the boulder.

Presenter: Yeah, and where a lot of this comes from, and I've heard this said by people, is, you know, "Well, I may not believe that I'm going to be successful with traditional, but I know that nobody's going to fire me because I did things traditional."

Presenter: Right.

Presenter: Because that's what's in, you know, that's what the regulations were written for, that's what all the templates were built for. But if I do this, which is different from, even if I've gotten permission, it's still different from and it doesn't work, I might get fired.

Presenter: Right.

Presenter: Or, you know, transferred or demo--and so that risk aversion, that personal risk, is the thing that you have to be able to get behind. So this is a leadership thing.

Leadership has got to exhibit the behaviors that you're not going to get fired because you tried something and it didn't work as well as you expected it to.

Readiness Gimmees

Readiness Gimmees



- Your Agile approach has been tailored to best meet your program's needs
- Program office staff including systems engineers understand their role in the agile process you're using
- The agile manifesto and principles are understood throughout the organization
- Appropriate training has been provided for the entire organization
- Expectations and artifacts necessary for milestone decisions have been agreed to and documented
- Agile roles and responsibilities have been clearly assigned
- The definition of done has been established and includes what documentation is required for iteration and increment deliveries
- The program office is open to changing roles
- Leadership and staff are educated on differences from the way they are used to doing business
- Program develops and utilizes adoption support (communication and implementation) mechanisms

**027 And most leaders, if you approach it in a way that helps them to understand the learn fast, fail fast kind of idea, they get it. I'm working with people that are very, very good about not beheading people because the answer was not what they expected. But it takes them understanding that this is really an area that they have to work with.

Presenter: And that takes us nicely into our Readiness gimmees.

Presenter: Okay.

Presenter: The idea that leadership and staff are educated on the differences in the way that they're doing business and that everybody then, everybody in the process is educated about how their responsibilities need to be adapted.

Presenter: And I'm, you know, technology transition is one of my research areas and I, I get upset with people lots of times when the only transition mechanism that they talk about is training. They don't think about all the other things and communications and implementation supports that are needed. But I do have to say, this is an area where education and training makes a huge difference. I had one program where they started over a year before they were going to implement Agile on the contract and they were educating everybody. Contracts people, test people, security people, engineers, program managers, leaders, everybody they could get into the class, to understand the different way they were going to do business.

Another program, they, they trained kind of a core cadre, but a lot of those ancillary groups, they really didn't spend time training, and when they got going, they ran into all kinds of roadblocks because they were running into other people's rice bowls, they didn't understand how they were supposed to work, and they had to go back and do a lot of retraining, you know, sort of backup training, to get that. So this is an area where investing in training for

the people that are going to be doing the work, you need to look, you need to understand your value stream enough to understand who's really affected by this so that you can get the right people trained, and let's see. There's some gotchas that go along with Readiness, too.

Readiness Gotchas

Readiness Gotchas



- Your testing function/organization has not been integrated into the day-to-day activities
- Requirements stability, operating environment, and the evolution of the technology base has not been fully assessed
- Constraints are imposed for the sake of tradition
- Contract progress payments are based on "earned value" for the accounting period vs the Agile working cadence
- Regulations are cited as a reason not to embrace agile approaches

**028 Presenter: There are.

Presenter: And so along with automated testing, there's the idea that testing function needs to be integrated into the day-to-day development work, and I'm not necessarily talking about the people that are in the what we would call the OT&E and DT&E necessarily, that do the mandatory independent testing after the system is built. They may not be involved day-to-day although they can be involved much

earlier and they can be involved in an incremental way as well. But there's a level of testing that is needed just to get through integration that you want your testers to be involved in on a daily basis almost in many of these teams, and that is something that's easy to forget about.

Implementation Gimmees

Implementation Gimmees



- Users and other stakeholders can accommodate incremental deliveries
- Necessary and beneficial documentation has been identified
- Requirements can be prioritized without pushback
- Agile requirements constructs –stories, features, epics, etc. -- have clear completion criteria
- (Incremental)technical reviews are structured to understand technical issues and mitigate technical risks
- Iteration and release reviews are used to build a case to demonstrate readiness to pass milestone reviews
- Agile measurements are integrated into overall management metrics
- Measurements are focused on "are we producing sufficient value fast enough?"
- User requirements are validated during the creation of user stories and features
- The program office is changing how they perform them their responsibilities (e.g. incremental technical reviews, CDRL deliveries...)

**029 Presenter: So on the implementation side, we actually had some pretty good questions that brought us down our list of implementation--

Presenter: Sure.

Presenter: --gimmees already, talking about accommodating incremental delivery. We talked about tech reviews. We talked about reprioritization of requirements on a regular basis, and we talked about

the idea that everyone is engaged in changing how they perform their responsibilities, is a great gimmee.

Presenter: Yep. Yep.

Presenter: Yeah. So why don't we--

Presenter: Very good.

Presenter: --press towards implementation gotchas in the few minutes that we have left.

Implementation Gotchas

Implementation Gotchas



- Your program or contractor is proposing agile as a quick fix for existing failures on the program
- Team metrics are used for comparisons
- Users and stakeholders who are not actively engaged in the Agile processes can't adapt to small batch deliveries
- Oversight activities are abandoned
- Multiple organizational change initiatives compete with Agile for the attention of leadership
- Cadence of multiple interacting teams hasn't been synchronized
- Focus is on compliance rather than mission success
- Derived requirements can't be reprioritized based on what is learned in early development iterations
- Contractor leadership and engineering implementation staff are not aligned on process changes needed to support Agile delivery
- Incentives to contractor don't reflect Agile/lean principles

**030 Presenter: The one I want to hit is multiple organizational change initiatives competing with Agile. Because that's one of the things we find, if you've got--we're doing government as the integrator, we're doing open systems architecture, we're doing Agile, we're doing this, we're doing that. Agile

can get lost, and so you may want to look at your timing if that happens. So that's one to be particularly mindful of.

Getting off the Stage



**031 Moderator: So we're at the two-minute warning. Can we fire off a couple quick questions to you guys and then I'm going to give the audience some learning opportunities that we have coming up here?

Presenter: Okay.

Presenter: We can do that.

Moderator: Question from Lance asking, "How do you incorporate security requirements in Agile?"

Presenter: So first of all, you've got to incorporate security people.

Presenter: Yes.

Presenter: Right? Into your stakeholder group, and we are seeing a lot of willingness in the security community to work incrementally, to work early, and to keep up with things. There's some that are not as willing, but I'd say the movement is towards being willing to work incrementally. But you've got to ask them and you've got to include them. So they need to help you. Now, we, a particular thing that we recommend is that in the definition of Done for each layer, iterations, releases, solutions, security requirements or reference that you've done the required work, that's one of the ways to deal with that.

Presenter: Right.

Moderator: Okay. And we'll wrap up this one from Sarah asking, "Automated testing works well in development. For example, unit testing. Can you talk about how to automate customer acceptance testing, or is this a good idea?"

Presenter: Yes, it's a good idea.

Presenter: Mm-hm.

Presenter: But it's some different things. So a lot of that is around user interface stuff, around exploratory testing and those kinds of things. The exploratory testing you tend not to automate, but if you've done a lot of the volume testing, a lot of the load testing, a lot of that kind

of stuff comes into customer acceptance testing. That's very easily automated and then that leaves you more time to do the exploratory testing. So yes, we're seeing that as well.

Moderator: Okay. So we got just a couple seconds left. I wanted to thank you guys for a terrific presentation today, thank everyone for attending. We had lots of questions about recording of the event today, was archived. It'll be available tomorrow. Everyone will get an e-mail. I did mention some upcoming learning opportunities from the SEI. There's a three-day course "Leading SAFe in Government," taught by Suz and Eileen.

Presenter: Yeah.

Presenter: Yeah.

Moderator: And the first offering's May 9th through 11th, so we will send that information about that course. Anything you want to add about that course quickly?

Presenter: It'll be fun.

Moderator: Okay.

Presenter: It's a good time. Yeah.

Moderator: Also, another opportunity is the second annual Software Solutions Symposium, which is taking place March 20th through 23rd in Arlington, Virginia, and there is a whole track actually on

"Agile In Government," speakers from government and industry," so if you found value in this, you'll definitely want to attend that. Yeah, and we'll send information about that, and lastly, our next virtual event will be "Five Things You Need to Know for Leading a Successful IT Modernization Project," and that's going to be March 13th, and we'll send more information about that. So again, great presentation. Thank you.

Presenter: Thank you.

Presenter: Thank you, Shane.

Moderator: Thank you for everyone's time today. Have a great day, everyone. Thank you.

SEI WEBINAR SERIES | Keeping you informed of the latest solutions

