SEI Virtual Learning

Learning Package 1: Agile in the DoD Landscape

SEI Continuous Deployment of Capability Directorate

April 2021

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

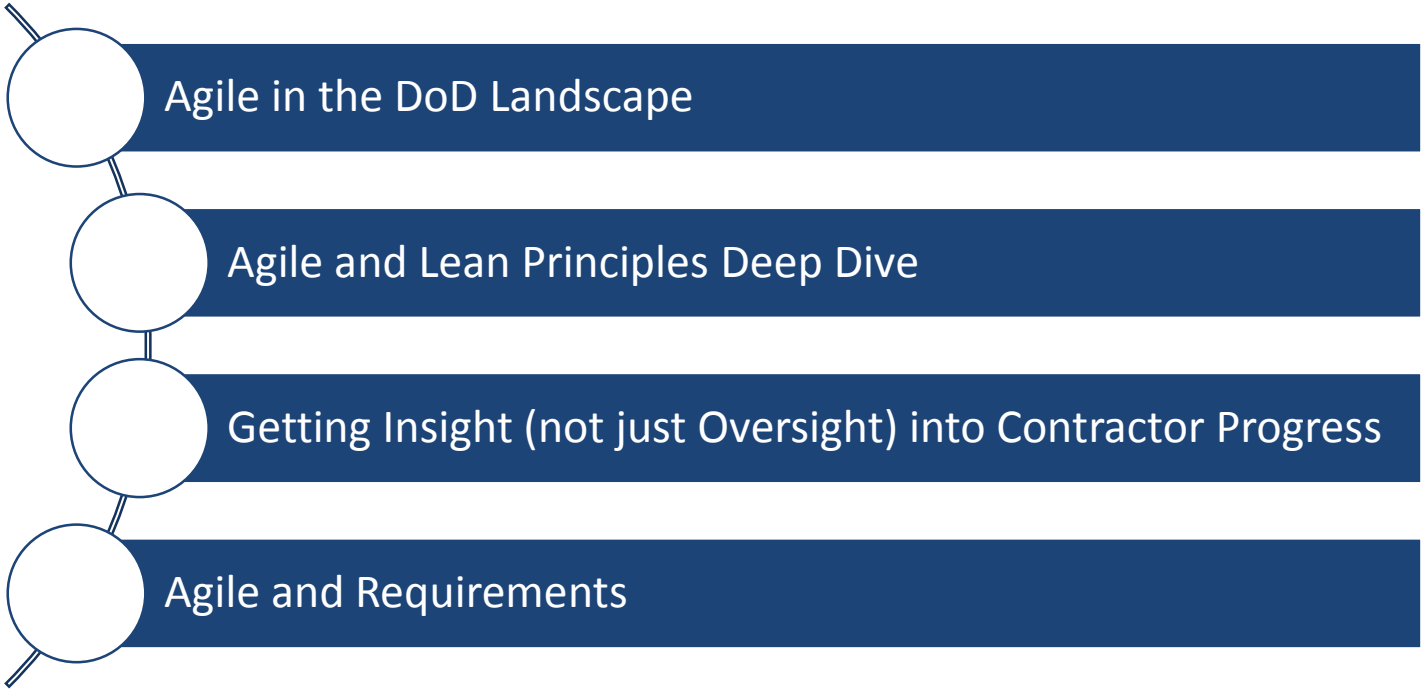
DM21-0359

Participant Poll

www.pollev.com/mainsummit799

What one word comes to mind when you think of Agile?

Course Roadmap



Topics the SEI will address in this course module include:

- **DoD Motivation for Agile and Lean**
- **Agile Foundations**
- **Recent Developments in the Acquisition Landscape**

The course module is scheduled for 2.5 hours:

- Hour 1 is primarily lecture
- 10 minute break at the one hour mark
- ~ 20 more minutes of lecture
- ~ 30 minutes of Q&A

Meeting Conventions for Today

Please stay on mute for the lecture portion of the course module.

If you are “in” the ZoomGov meeting via web or app, please ask questions via the Chat window.

- A facilitator will collect the questions and either pass them to the instructor if something immediate, or organize them for the Q&A portion of the course

Those on dial-in can email questions to your POC who will add them to Chat

Instructor will call for participation and discussion at various points. Please remember to come off mute before talking.

When you are done talking, before going back on mute, please say “Over” so others know you are finished.

Today’s Landscape

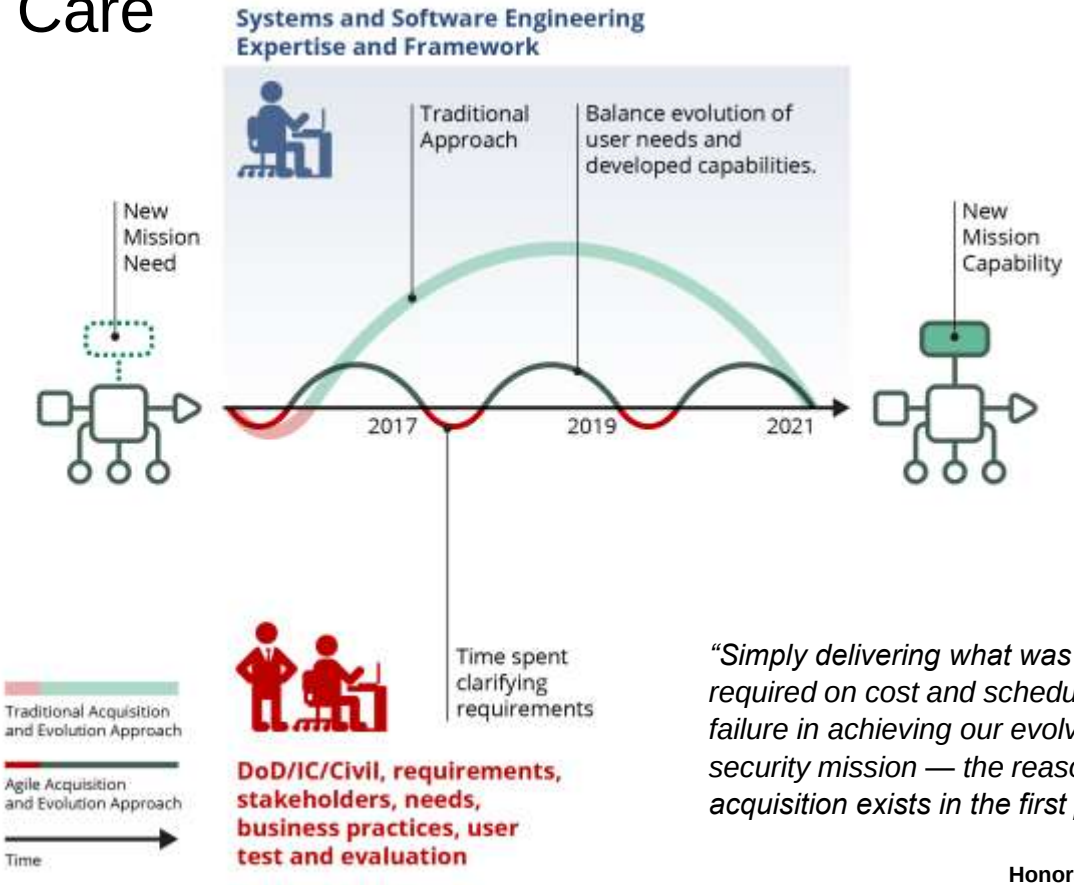
Why Does the DoD Care about Agile?

Deliver performance at the speed of relevance

Streamline rapid, iterative approaches from development to fielding

National Defense Strategy Summary
Jan 2018

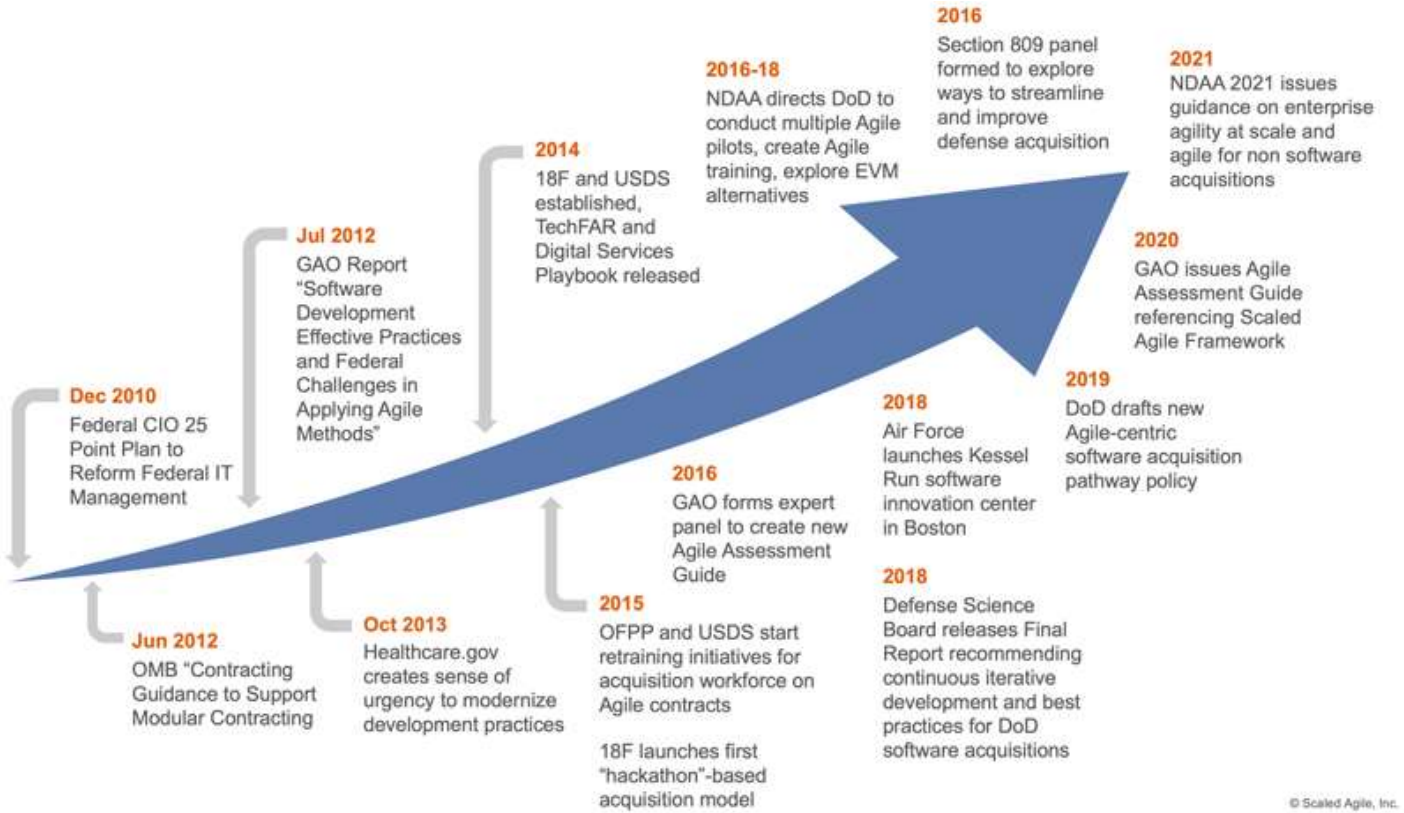
"Software never dies."
DSB Task Force, 2017



"Simply delivering what was initially required on cost and schedule can lead to failure in achieving our evolving national security mission — the reason defense acquisition exists in the first place."

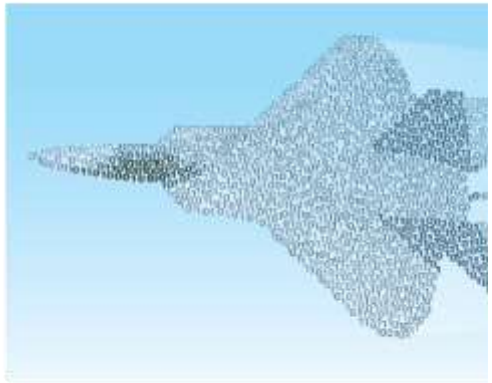
Honorable Frank Kendall
Under Secretary of Defense (AT&L)
2015 Performance of The Defense Acquisition System

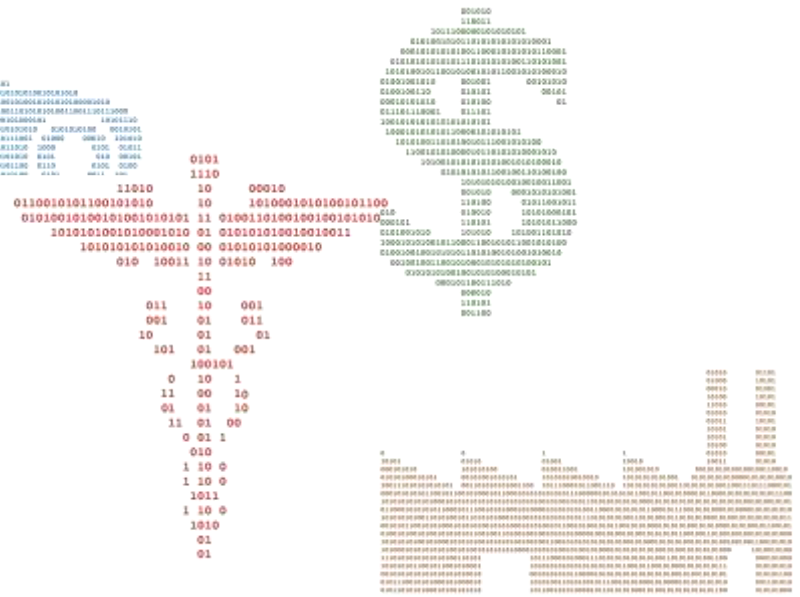
Events Driving Lean-Agile Adoption in US Government

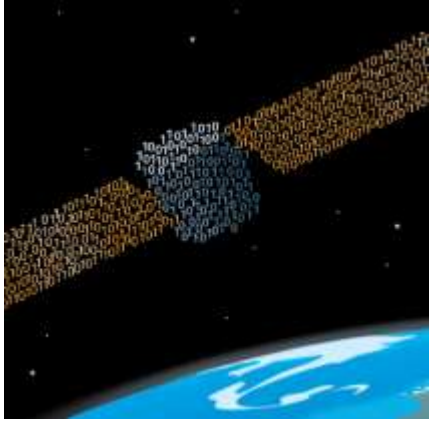


Source: https://www.scaledagileframework.com/government-article/, as of Apr 13, 2021

Why Does the DoD Care about Software?



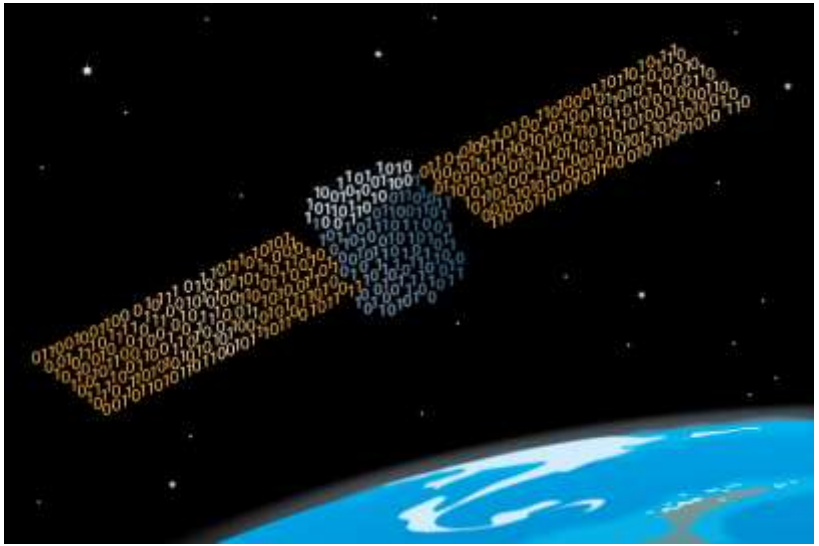




If the world isn't software-defined yet, it is certainly software-reliant!

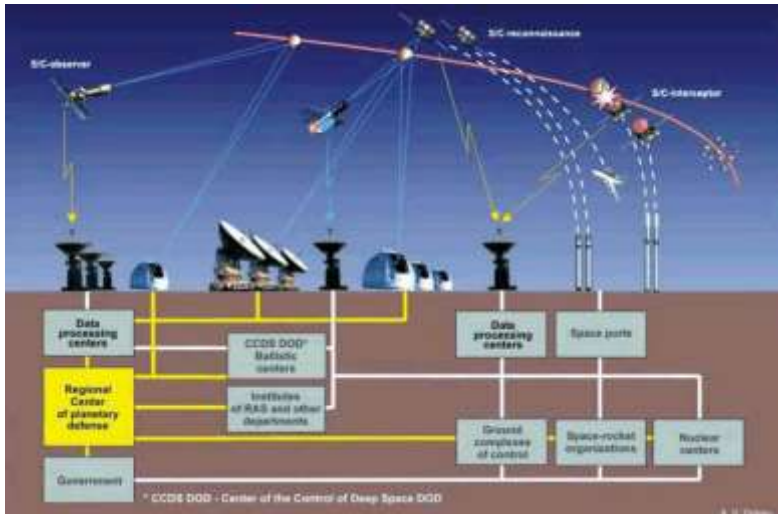
Software is often part of a larger System Development Activity

- Major system engineering endeavors in aircraft vehicles, satellites, economic infrastructure, energy management…
  - Often these are contracted systems, where the software part of the system is being acquired by the owner of the overall system



Software is a Primary Connector Among Systems (of Systems)

"The value of a system passes through its interconnections" [1]



1 scaledagileframework.com\apply-systems-thinking

Integrated functionality often means overlapping accountability.

Who "owns" the responsibility for software in a complex system of systems?

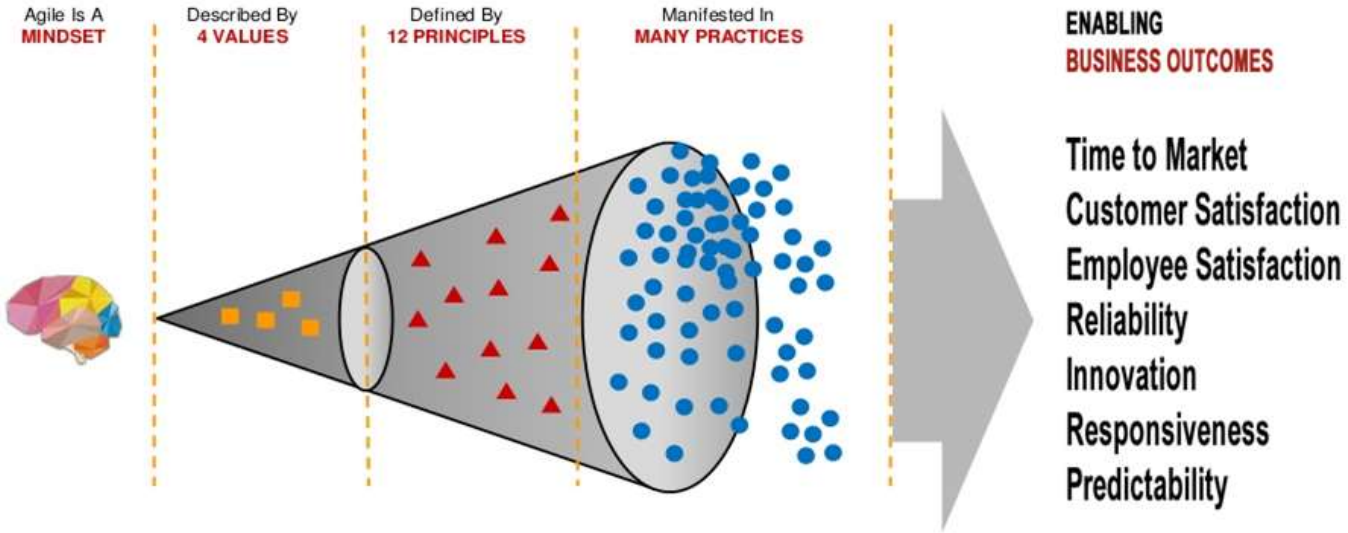
Agile Foundations

Working Definition of Agile

Agile An *iterative* and *incremental* (evolutionary) approach to software development which is performed in a *highly collaborative manner* by *self-organizing teams* within an *effective governance framework* with *"just enough" ceremony* that produces *high quality software* in a *cost effective and timely* manner which *meets the changing needs of its stakeholders*. [Ambler 2013]

[Ambler 2013] Ambler, Scott. *Disciplined Agile Software Development: Definition*.
http://www.agilemodeling.com/essays/agileSoftwareDevelopment.htm

What is Agile?



Source: https://www.slideshare.net/MichaelTarnowski/agile-mindset-for-executives

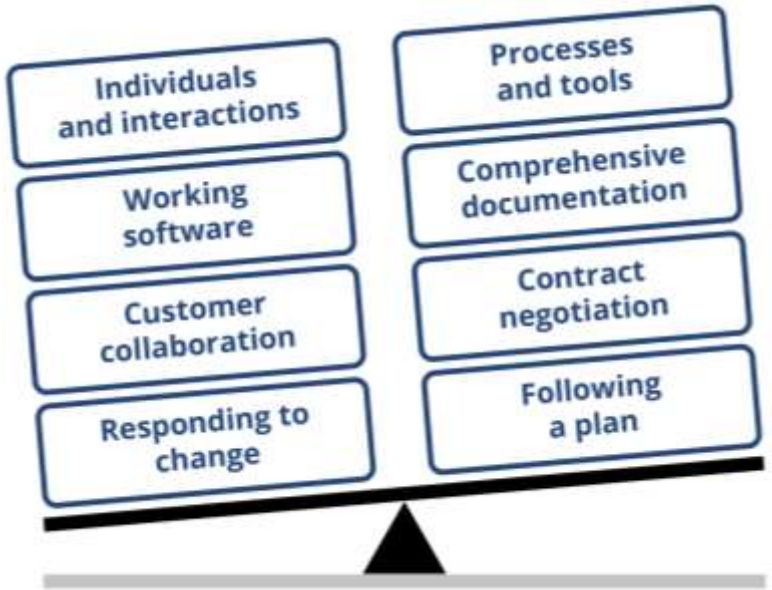
Implementing the practices, tools and processes **without** the Agile mindset, values, and principles of the Agile Manifesto

Is NOT Agile!

It isn't enough to adopt the practices of a successful team. You must adopt attitudes and a mindset for making decisions to adopt practices that will lead to your success.

Manifesto for Agile Software Development--2001

Through this work we have come to value:



That is, while there is value in the items on the right, we value the items on the left more.

https://agilemanifesto.org

"The Agile movement is not anti-methodology. In fact, many of us want to restore credibility to the word methodology. We want to restore a balance."

Jim Highsmith – Short Post on the History of the Agile Manifesto

http://www.agilemanifesto.org/history.html

Agile Principles

For most of these, substituting “working product” allows translation into non-software environments.

1. Our highest priority is to **satisfy the customer** through early and continuous delivery of valuable **software.**
2. Welcome **changing requirements**, even late in development. Agile processes harness change for the customer's competitive advantage.
3. **Deliver working software frequently**, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. **Business people and developers must work together** daily throughout the project.
5. **Build projects around motivated individuals**. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is **face-to-face conversation.**
7. **Working software is the primary measure of progress.**
8. Agile processes **promote sustainable development.** The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to **technical excellence and good** design enhances agility.
10. **Simplicity--**the art of maximizing the amount of work not done--is essential.
11. The **best architectures, requirements, and designs emerge from self-organizing** teams.
12. At **regular intervals**, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile Principles: http://agilemanifesto.org

Participant Poll

www.pollev.com/mainsummit799

Which principle(s) do you think are most applicable to the work you do?

Interpretation Notes for Using Agile Principles in 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.

Interpretation Notes for Using Agile Principles in 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.	Our 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 to 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.

Interpretation Notes for Using Agile Principles in 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 the 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.

Some Observable Characteristics of Agile Implementations

Iterative—elements are expected to move from skeletal to completely fleshed out over time, not all in one step

Incremental—delivery doesn't occur all at once

Collaborative—progress is expected to be made by stakeholders and the development team working collaboratively throughout the development timeframe

Modular Architecture—multiple self-organizing, cross-functional teams can work concurrently on multiple product elements (e.g., requirements, architecture, design, and the like) for multiple loosely coupled product components

Dedicated—team members are allowed to focus on the tasks within an iteration/release as opposed to task-switching across multiple projects

Time-boxed or Flow-based—relatively short-duration development cycles that permit changes in scope rather than changes in delivery time frame

Iterative & Incremental

"Scrum & Agile are both iterative and incremental. Iterative in that we plan for the work of one iteration to be improved in subsequent iterations.

Incremental because completed work is delivered throughout the project"

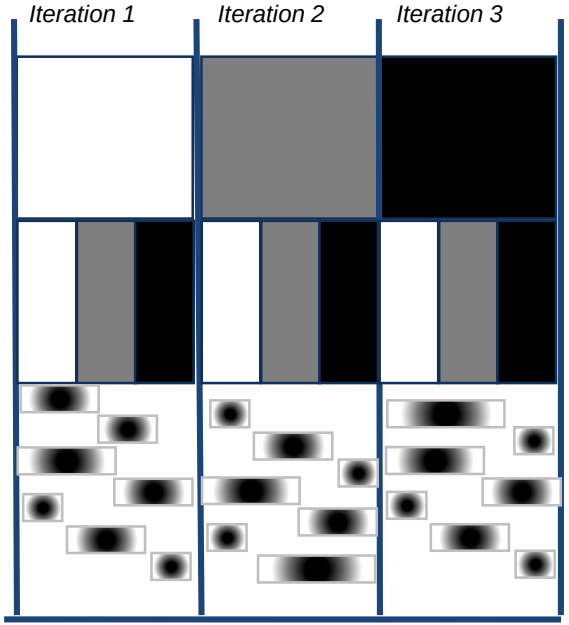
https://www.mountaingoatsoftware.com/blog/agile-needs-to-be-both-iterative-and-incremental

Taking an Iterative Approach

Single batch – one process step per iteration

Multiple batches - complete each batch at the end of an iteration; siloed process steps within each iteration

Multiple batches - decompose each batch into small packages, with multiple start-to-finish cycles in each iteration



All Common Agile Team Methods are Based on PDCA Cycle

Plan - a task, change or test, aimed at improvement.

- Analyze what you intend to improve - choose areas with highest rate of return

Do - Carry out the change or test (preferably on a small scale).

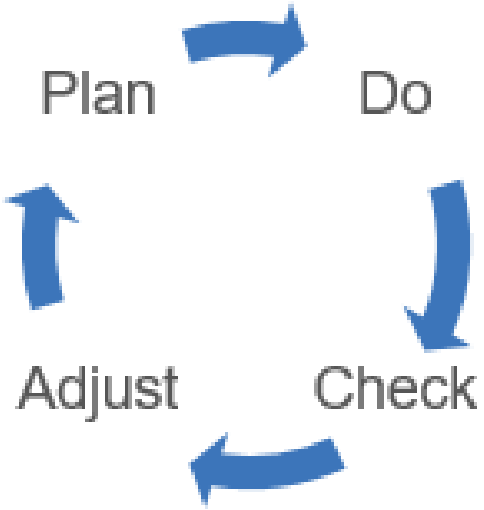
- Implement the change you decided on in the plan phase.

Check - the results. What was learned? What went wrong? (We have received empirical data! We are NOT guessing!)

- Measure / monitor the level of improvement.

Adjust – Make the change based on the empirical data.

- Adopt the change, abandon it, or run through the cycle again.



Plan – Do – Check – Adjust (PDCA) cycle is inherent in all of our work.

Traditional vs. Agile Approaches

Agile approach:

- Supports an environment of innovation, change & uncertainty
  - Programs with volatile requirements and environment
  - Programs where solutions are sufficiently unknown that significant experimentation is likely to be needed
  - Programs for which the technology base is evolving rapidly
- Requires an environment that can support collaboration
  - Programs with stakeholders who can engage with developers in ongoing, close collaboration

Nidiffer, K. Miller, S. & Carney, D. *Potential Use of Agile Methods in Selected DoD Acquisitions: Requirements Development and Management* (CMU/SEI-2013-TN-0006), September 2013.

Traditional vs. Agile Approaches

Traditional approach

- Is consistent with historical acquisition lifecycle guidance provided up until recently
- Assumes stability:
  - Programs with stable requirements and environment, with known solutions to the requirements
  - Programs for which the technology base is evolving slowly (technology is not expected to be refreshed/replaced within the timeframe of the initial development)
- Requires stakeholders who:
  - Know what they want before they see it
  - Communicate well via documents

Nidiffer, K. Miller, S. & Carney, D. *Potential Use of Agile Methods in Selected DoD Acquisitions: Requirements Development and Management* (CMU/SEI-2013-TN-0006), September 2013.

Participant Poll

www.pollev.com/mainsummit799

How do Agile principles, practices, or mindset influence decision making in your organization?

Recent Developments in the Acquisition Landscape

The Defense Innovation Board Software Acquisition Practices Report

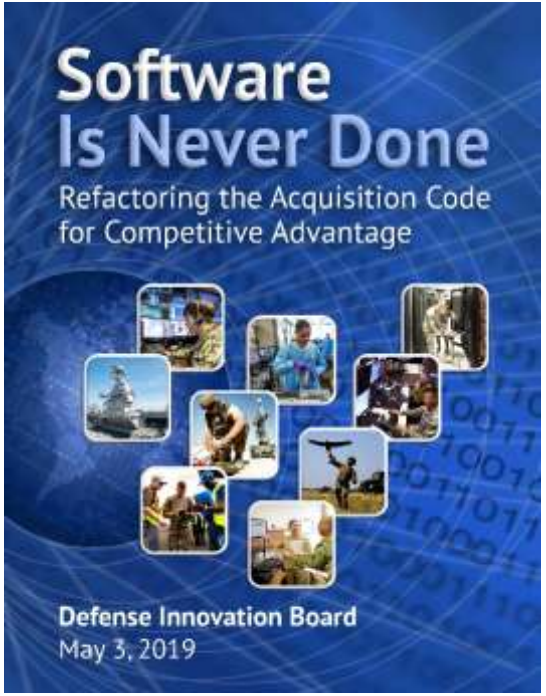
Most influential of recent reports

“Software is Never Done”

https://innovation.defense.gov/software/

The authors considerately created multiple versions, from a two page flyer to a large report

Strong influence on the 2019 and 2020 National Defense Authorization Acts



Motivation for DIB SWAP

Current state – the problem:

- **Software is ubiquitous and U.S. national security relies on software.** The ability to acquire and deploy software is central to national defense and integrating with allies.
- **The threats the U.S. faces change rapidly**, and DoD's ability to adapt and respond is now determined by its ability to develop and deploy software to the field.
- **The current approach to software development is a leading source of risk to DoD:** it takes too long, is too expensive, and exposes warfighters to unacceptable risk.
- **Software is not being used to enable a more effective force,** strengthen our ability to work with allies, and improve the business processes of the Department.
- **Nothing is changing:** most of this has been said before and the 1987 DSB report on military software pretty much says it all. What is it going to take to actually do something?

Main Points of the DIB SWAP

Main lines of effort:

- **Congress and OSD: Refactor statutes, regulations, and processes for software**, providing increased insight to reduce the risk of slow, costly, and overgrown programs, and enabling rapid deployment and continuous improvement of software to the field.
- **OSD and the Services: Create and maintain cross-program/cross-Service digital infrastructure** that enables rapid deployment, scaling, testing, and optimization of software as an enduring capability; manage them using modern development methods; and eliminate the existing hardware-centric regulations and other barriers.
- **Services and OSD: Create new paths for digital talent (especially *internal* talent)** by establishing software development as a high-visibility, high-priority career track and increasing the level of understanding of modern software within the acquisition workforce.
- **DoD and industry: Change the practice of how software is procured and developed** by adopting modern software development approaches.

Recommendations Have Resulted in Action

A few of the things resulting from the report:

- Creation of “Adaptive Acquisition Pathways” for Software
- Creation of “Chief Software Officer” (CSO) position in the US Air Force
- Impetus for 5000.02 rewrite currently underway
- Defense Digital Service (DDS) actively pursuing innovations culled from the report

Bonus: Detecting Agile BS

One of the Appendices: "Detecting Agile BS"
https://media.defense.gov/2019/May/02/2002127286/-1/-1/0/DIBGUIDEDETECTINGAGILEBS.PDF

Agile value	DIB maxim
Individuals and interactions over processes and tools	"Competence trumps process"
Working software over comprehensive documentation	"Minimize time from program launch to deployment of simplest useful functionality"
Customer collaboration over contract negotiation	"Adopt a DevSecOps culture for software systems"
Responding to change over following a plan	"Software programs should start small, be iterative, and build on success – or be terminated quickly"

The Defense Science Board Study

Primary study that led to the chartering of the DIB SWAP Study

DoD Defense Science Board

The Defense Science Board serves as the Federal Advisory Committee chartered to provide Department of Defense leadership with "independent advice and recommendations on science, technology, manufacturing, acquisition processes, and other matters of special interest to the DoD...”

Currently, the Board's authorized strength is forty-eight members and seven *ex officio* members, including the chairs of the Army, Navy, and Air Force advisory committees, and the Defense advisory committees on Policy, Business, Health, and Innovation. The Board's forty-eight members are appointed for terms ranging from one to four years …”

As of February 1, 2018, the newly formed Office of the Under Secretary of Defense for Research and Engineering (USD(R&E)) will serve as the formal sponsor of the Defense Science Board.

Task Force Findings and Recommendations on “Design and Acquisition of Software for Defense Systems” issued October, 2017.

https://www.acq.osd.mil/dsb/history.htm

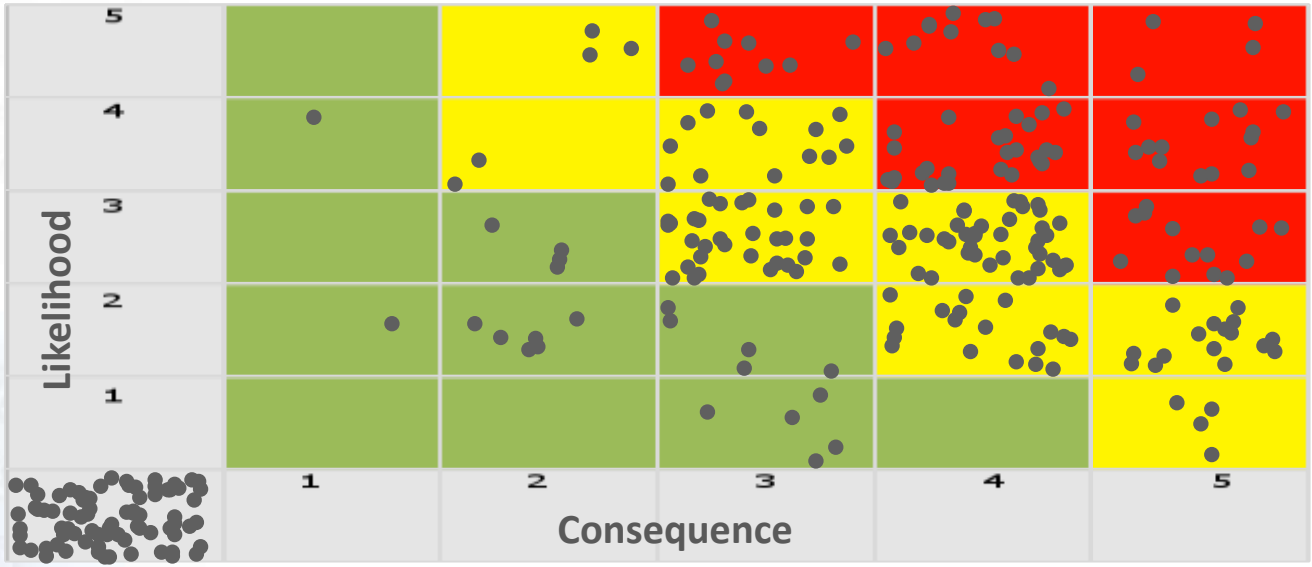
Importance of Software in Defense Systems

- Software is a crucial and growing part of weapon systems/national security mission
  - "The DoD is experiencing an explosive increase in its demand for software-implemented features in weapon systems…in the meantime, defense software productivity and industrial base capacity have not been growing as quickly."
    –Institute for Defense Analyses, 2017

- Software never dies. It will require DoD to update continuously and indefinitely

Software Risk Assessed by DoD Program Offices

FY14 - FY16



Software not in top program risks

Software assessed among most frequent and most critical challenges, driving program risk on ~ 60% of acquisition programs

Defense Science Board Task Force Findings

Commercial development practices allow rapid development, deployment and adaptability.

The DoD is behind the commercial sector, as are DoD contractors

DoD contracting must change to incentivize changes in suppliers

- Current contracts and incentives are built around waterfall processes used 20 years ago. This creates a self-reinforcing loop that stifles change

New resources are becoming available to guide acquisition organizations wrt contracting for Agile products and services

DSB Task Force Recommendations

DoD and its contractors need to adopt continuous iterative development best practices for software

Software acquisition is not about buying a black box end result

- The offeror's software factory should be a key source selection evaluation criterion
- Ensure software sustainment is considered in RFPs
- Contract for an ongoing stream of delivered value; not for a fixed point in time product

DSB Task Force Recommendations

Transition Strategy

- Ongoing development programs should plan to transition to a software factory and continuous iterative development
- Legacy programs should plan to perform a business case on whether to transition the program

Build metrics based on Agile best practices

Accept that there may be initial short term costs

Develop Agile competency in the workforce. (It takes 2 to tango—it can't all fall to the contractor to "do Agile")

USAF SMC State of Agile Report 1.0

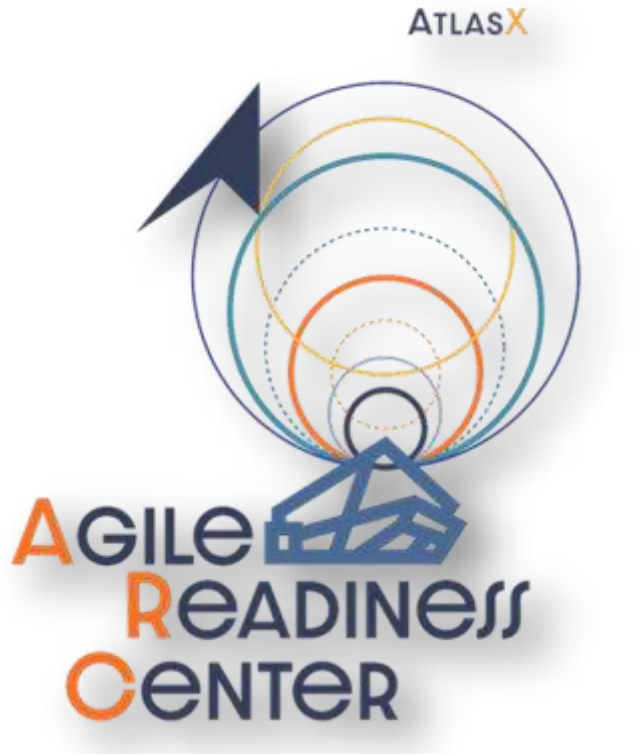
The SMC State of Agile Report #1 – September 2019

Sponsored by USAF Space and Missile Center's Agile Readiness Center

A questionnaire-based report of SMC staff and their use and perceptions about Agile, Lean, and (a bit of) DevSecOps

Modelled after the Collabnet Annual Agile Survey with adaptations made for SMC context:

- Demographic information collected is relevant to the government context, not the commercial
- SEI noted that the Collabnet survey focused on how far Agile adoption had reached
  - Additional questions were added to get to the question of "how routine has Agile become?"



https://drive.google.com/file/d/0B7UuG6BRYwvxMnZQdElKMnhzcjVvWl9uaXNPZy05X1Z0eDRr/view

Things Included in the SMC SOA Questionnaire to Address Pace of Adoption

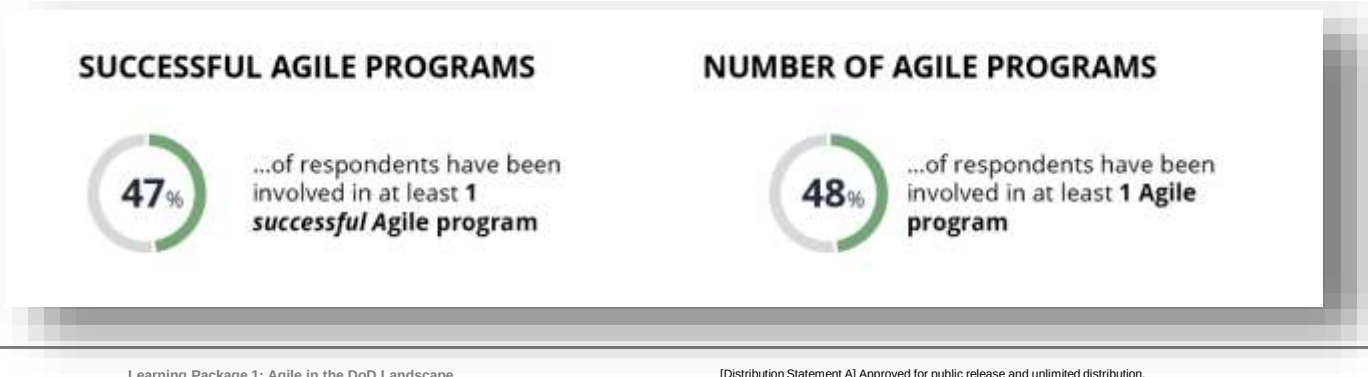
SMC Experience Using Agile Practices

- None→Considering→ Experimenting→Maturing→ Routine

Monitoring change to this measure in future will help to understand the pace of adoption throughout the enterprise

Pace of Adoption-2

A different “how many” question – how many respondents have been involved in successful (from their perception) Agile programs—almost half of SMC respondents reported involvement with at least one successful Agile program



Things Included in the SOA Questionnaire to Understand How Routine Agile has Become (Infusion)-1

SMC's State of Agile Questionnaire included several quantitative indicators of infusion

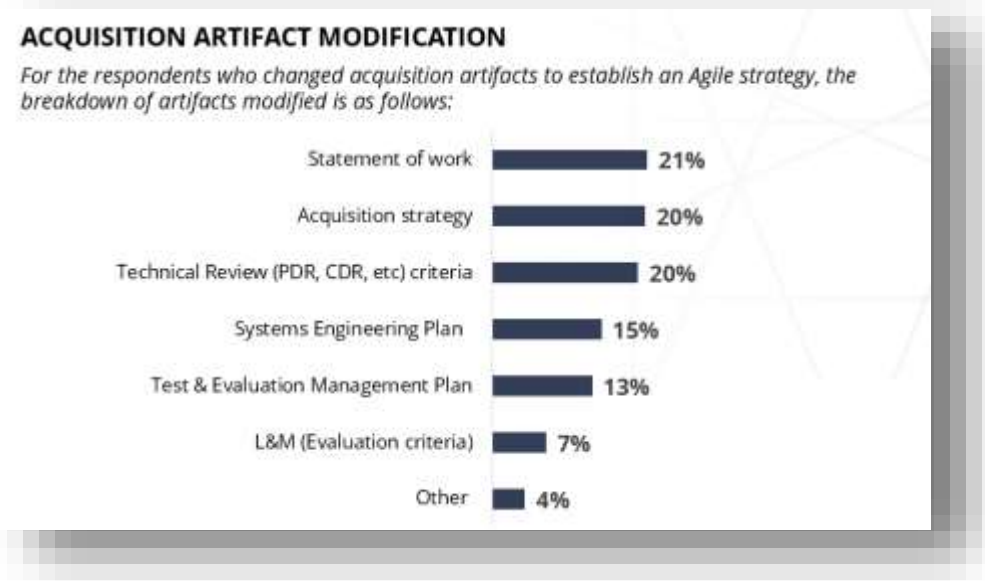
1) Agile's role in decisionmaking:
   - No Part→Plays a Foundational Role

This question directly addresses the Infusion question and will provide a basis in future for understanding how quickly Agile and Lean practices move into a routine state

Infusion Measures-2

2) Artifact modification – modifying acquisition artifacts to accommodate Agile practices – is an indicator that the practices are starting to “stick”

- Rather than going back to traditional acquisition, some respondents are modifying the acquisition eco-system to accommodate Agile practices



But Wait, There's More!

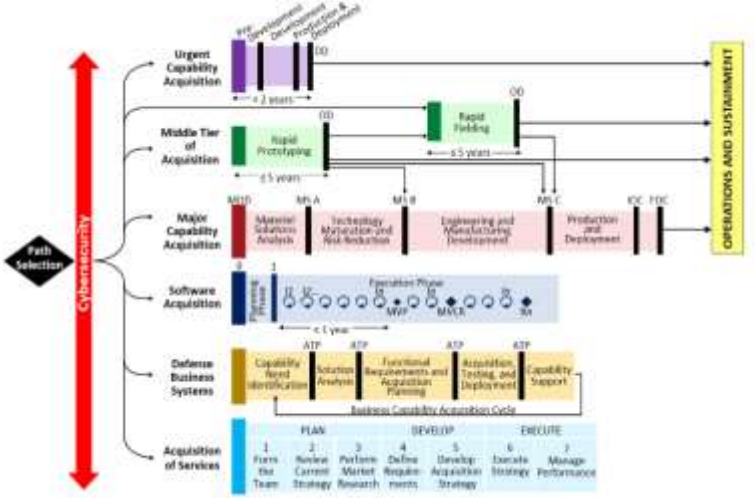
A Few Other Notable Recent Events/Links-1

Adaptive Acquisition Pathways: New ways of acquiring software-dominant systems (some of the guidance can also be applied to cyberphysical systems that are software-reliant)

https://aaf.dau.edu/

Section 873/874 Agile Acquisition Pilots: a group of small and large acquisition piloting Agile/Lean approaches to software development in different settings. Lessons learned document published 2020:

https://www.dau.edu/cop/it/DAU%20Sponsored%20Documents/AgilePilotsGuidebook%20V1.0%2027Feb20.pdf



A Few Other Notable Recent Events/Links-2

USAF Chief Software Officer:

https://software.af.mil/

https://software.af.mil/training/

SMC Agile/Lean Self-Service Learning Paths Catalog (in Beta, available to DI2E users):

Some of the materials in the learning package came from this site. If you have DI2E, you can access a larger set of materials directly:

https://confluence.di2e.net/display/AGILESMC/Self-Serve+Agile+Learning+Paths

Software Engineering Institute Agile Resources: podcasts, blog posts, Technical Notes on many Agile in Government topics:

https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=21345

Summary

Agile is Here to Stay!

Those who doubted that Agile practices in DoD would “stick” early on are learning about and applying Agile in their programs

Education is still spotty, and some Agile implementations fall short of mission and programmatic expectations

- More than just the software part of the acquisition eco-system is needed to be adapted in a large cyberphysical system program for the promise of Agile to be achieved

SEI has been observing and coaching Agile implementations across the entire government space, and especially in DoD, since 2009

- We’ve seen the good, the bad, and the ugly!

Participant Poll

www.pollev.com/mainsummit799

Where is Agile now?

Participant Poll

www.pollev.com/mainsummit799

End of Module Feedback

Contact Information

Pat Place
SEI Sr Member of Technical Staff
SSD/CDC
Email: prp@sei.cmu.edu

U.S. Mail
Software Engineering Institute
Customer Relations
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Customer Relations
Email: info@sei.cmu.edu
SEI Phone: +1 412-268-5800
SEI Fax: +1 412-268-6257