

Agile and Acquisition: The Obvious and the Not-So-Obvious

July 2022

Suzanne Miller, Bart Hackemack

**Software Engineering Institute
Agile Transformation Team**

Copyright 2022 Carnegie Mellon University.

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.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

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.

DM22-0619

Topics in this Workshop include:

- **Why do we care about Agile in acquisition?**
- **Differences between acquisitions with/without Agile in play**
- **Deeper dive into CDRLs to review for Agile acquisitions**
- **Summary/What else do you want to learn?**

Introduction/Purpose

This workshop is intended to help staff who are well-versed in Agile and Agile scaling concepts but who don't have a long history of working in DoD acquisition activities to recognize:

- Why programs are trying to move to Agile in the current environment
- What the key differences are in acquiring systems with/without Agile in play
- A few of the particular activities and artifacts that are affected by Agile acquisition

This workshop will explicitly not address the following topics:

- Agile & EVM
- The OTA (Other Transaction Authority) contracting mechanism
- The basics of defense acquisition as would be taught in DAU acquisition courses

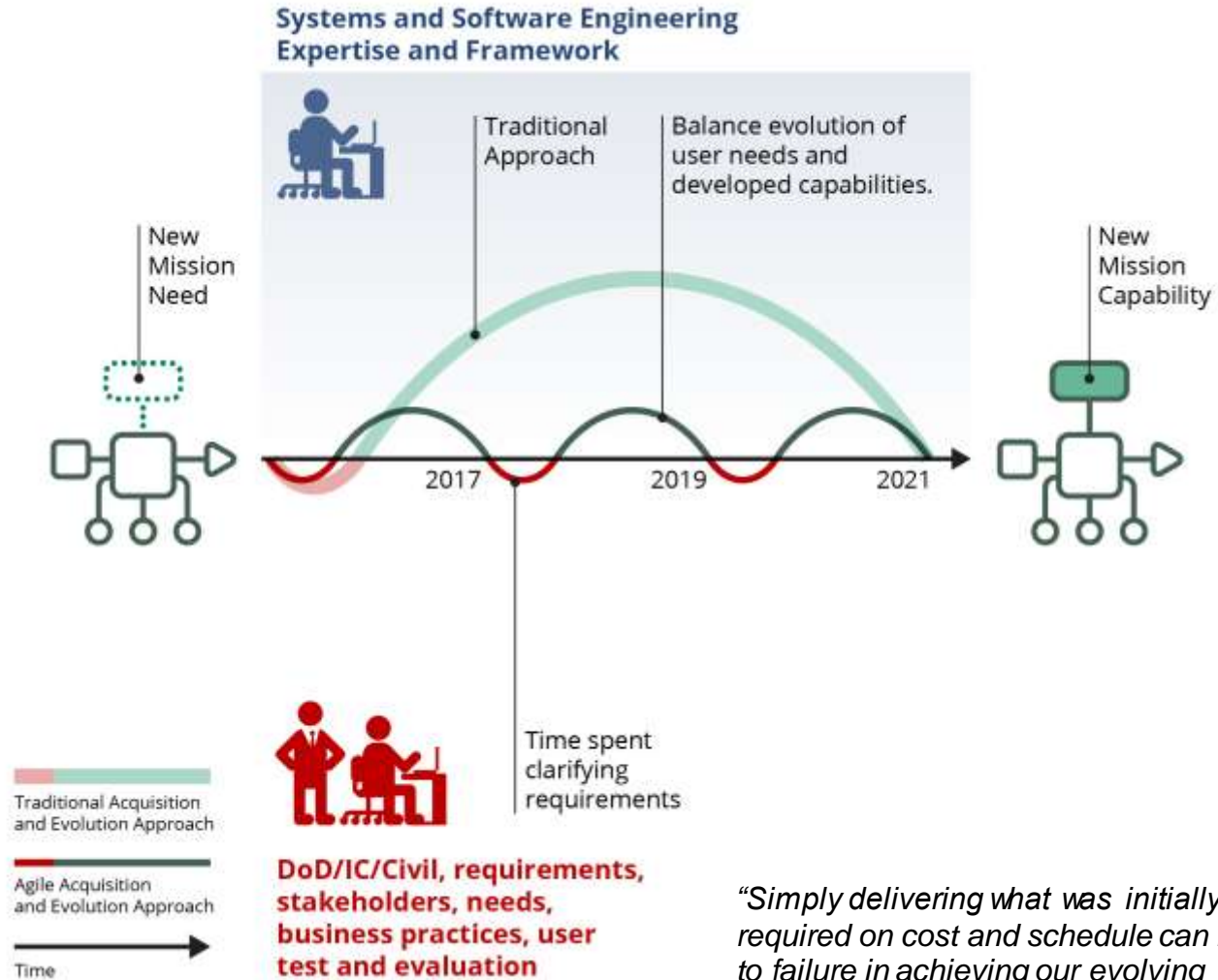
Why do we care about Agile in acquisition?

Why does the DoD/Govt care?

Deliver performance at the speed of relevance

Streamline rapid, iterative approaches from development to fielding

National Defense Strategy
Summary
Jan 2018



“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.”

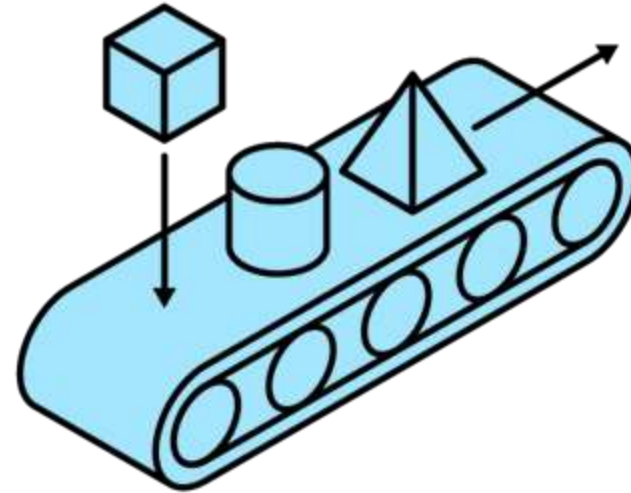
Honorable Frank Kendall
Under Secretary of Defense (AT&L)
2015 Performance of The Defense Acquisition System

A Whole New Acquisition Mindset



Building a
Box

Engineering a (most of the time) single product defined in one large batch and delivered one time



Building an ongoing
delivery stream of capabilities

Contracting for a product that is not completely defined at the start, and explicitly evolves and is delivered multiple, potentially many times

Large Software Projects Rarely Succeed

Project Size	Successful*
Grand	6%
Large	11%
Medium	12%
Moderate	24%
Small	61%

Source: Standish Group 2015 CHAOS Report

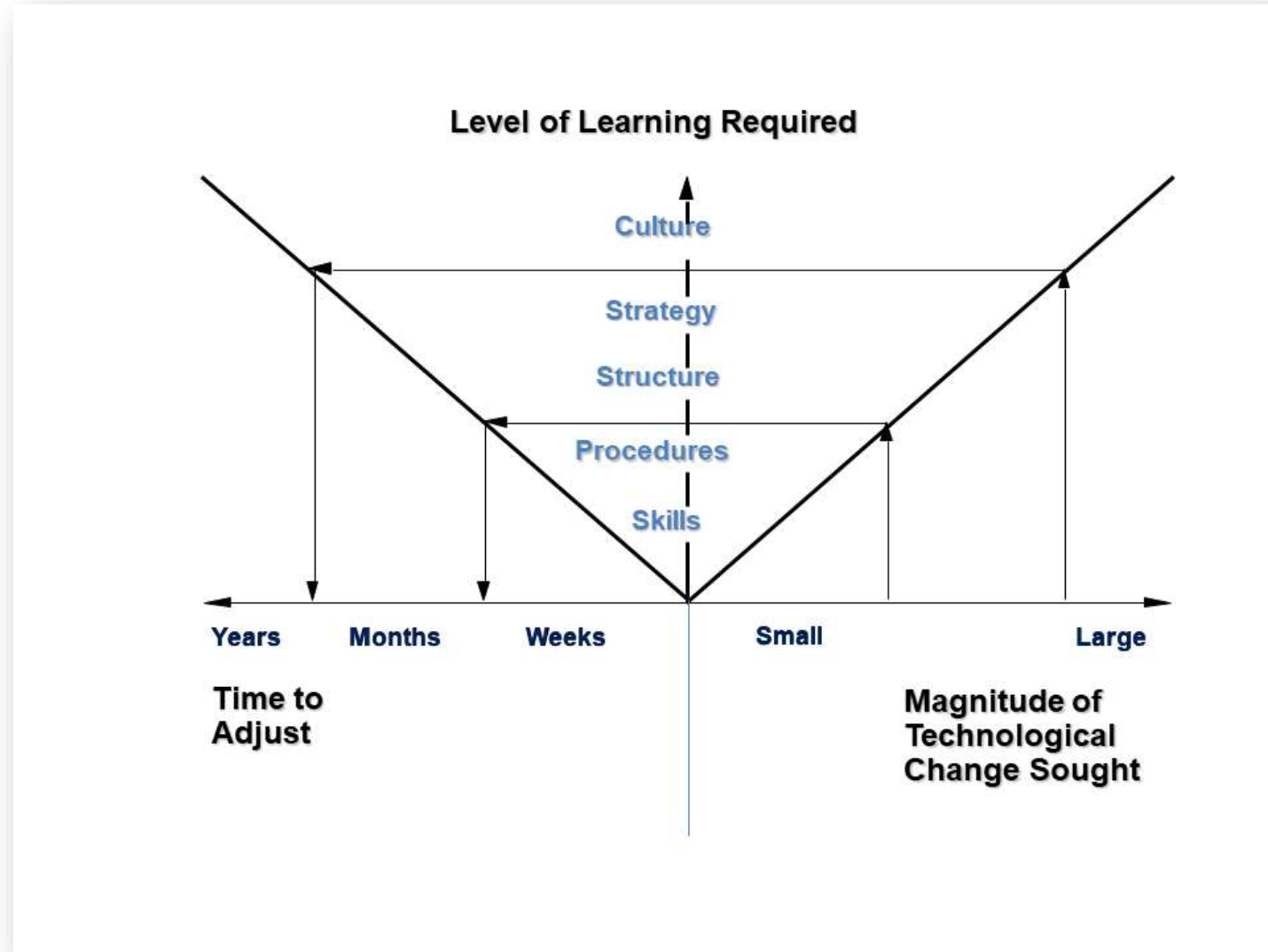
Advantages of small, incremental deliveries

- Fast feedback from stakeholders
- Less investment to move project goals forward
- Less time spent refining low priority items

* **Success:** On Time, On Budget, Satisfactory Result



No surprise—
 Most projects in DoD are at least medium if not large, and some are definitely in the “Grand” category!
 • Agile’s explicit focus on small batch can help, if acquisition strategy accounts for Agile practices

This Isn't an Easy Adoption Path for Most DoD Organizations





Source: Adler, Paul. "Adapting Your Technological Base: The Organizational Challenge", Sloan Mgmt Review, 1990.

Comparison of Agile and Traditional DoD Cultural Elements₁

Knowledge Piece	Method	
Organizational Structure 	Agile DoD <ul style="list-style-type: none">• Flexible and adaptive structures• Self-organizing teams• Collocated teams or strong communication mechanisms when teams are distributed	Traditional DoD <ul style="list-style-type: none">• Formal structures that are difficult to change• Hierarchical, command-and-control-based teams• Integrated product teams that have formal responsibilities
Leadership Style 	Agile DoD <ul style="list-style-type: none">• Facilitative leadership• Leader as champion and team advocate	Traditional DoD <ul style="list-style-type: none">• Leader as keeper of vision• Leader as primary source of authority to act


<http://www.sei.cmu.edu/library/abstracts/reports/11tn002.cfm?DCSext.abstractsource=SearchResults>

Comparison of Agile and Traditional DoD Cultural Elements₂

Knowledge Piece	Method	
Rewards System 	Agile DoD <ul style="list-style-type: none">• Team is focus of reward systems• Sometimes team itself recognizes individuals	Traditional DoD <ul style="list-style-type: none">• Individual is focus of the reward system
Staffing Model 	Agile DoD <ul style="list-style-type: none">• Cross-functional teams including all roles across the life cycle throughout the lifespan of the project• Includes an Agile advocate or coach who explicitly attends to the team's process	Traditional DoD <ul style="list-style-type: none">• Uses traditional life-cycle model with separate teams, particularly for development and testing• Different roles are active at different defined points in the life cycle and are not substantively involved except at those times

<http://www.sei.cmu.edu/library/abstracts/reports/11tn002.cfm?DCSext.abstractsource=SearchResults>

Comparison of Agile and Traditional DoD Cultural Elements₃

Knowledge Piece	Method	
<p data-bbox="529 376 868 468">Communications & Decision Making</p> 	<p data-bbox="1182 372 1365 415">Agile DoD</p> <ul data-bbox="1029 451 1518 993" style="list-style-type: none">• Daily stand-up meetings• Frequent retrospectives to improve practices• Information radiators to communicate critical project information• Evocative documents to feed conversation• "Just enough" documentation, highly dependent on product context	<p data-bbox="1696 379 1982 422">Traditional DoD</p> <ul data-bbox="1600 451 2091 1179" style="list-style-type: none">• Top-down communication structures dominate• External regulations, policies and procedures drive the focus of work• Indirect communications, like documented activities and processes, dominate over face-to-face dialogue• Traditional, representational documents used by the PMO throughout the development life cycle to oversee the progress of the developer• PMO oversight tools focused on demonstrating compliance vs. achieving insight into progress

<http://www.sei.cmu.edu/library/abstracts/reports/11tn002.cfm?DCSext.abstractsource=SearchResults>

Technology Assumptions Table for Agile-1

Fit Dimension	Agile & Devops Assumptions
Business & Acquisition	<ul style="list-style-type: none"> • Program acquisition strategy and practices enable, or at least don't dis-able, differences in developing and deploying using Agile approaches
Organizational Climate	<ul style="list-style-type: none"> • Reward systems, values, skills, sponsorship explicitly support Agile values and principles
Project, Team & Customer Environment	<ul style="list-style-type: none"> • Frequent collaboration between development team and test, operations, customers, and end users is actively supported • Program management practices don't force teams to work across different projects
System Attributes	<ul style="list-style-type: none"> • System architecture is loosely-coupled (interfaces are external vs internal among system components) • System solutions benefit from fast user/operational feedback

Technology Assumptions Table for Agile-2

Fit Dimension	Agile/Devops Assumption
Technology Environment	<ul style="list-style-type: none"> • Technology support for virtualization, automated testing and continuous integration are in place • Integrated collaboration platform is in place, including monitoring and feedback
Team Technical Practices (subset of Practices)	<ul style="list-style-type: none"> • Technical practices that support high quality code production in small batches from a prioritized product backlog are in place • Technical practices integrate automated testing and integration
Team Management/Coordination Practices (subset of Practices)	<ul style="list-style-type: none"> • Decentralized decision making that allows team members to self-organize their work are in place and supported • Team management practices that support short (2-4 week or less) time boxes are in place • Coordination practices among dev, test, and ops stakeholders are routinely used
Program Practices (subset of Practices)	<ul style="list-style-type: none"> • Synchronization of multiple teams is occurring • Practices that reinforce respecting team management and measurement boundaries are in place • Automated governance mechanisms are used where appropriate

From what you know today....

On one flipchart, answer:

“How are Lean/Agile principles
COMPATIBLE with conventional DoD
acquisition practices?”

On second flipchart, answer:

“How are Lean/Agile principles
INCOMPATIBLE with conventional DoD
acquisition practices?”

Differences between acquisitions with/without Agile in play

Oversight/Insight



Oversight

Oversight and insight are closely coupled concepts, similar to verification and validation.



Insight

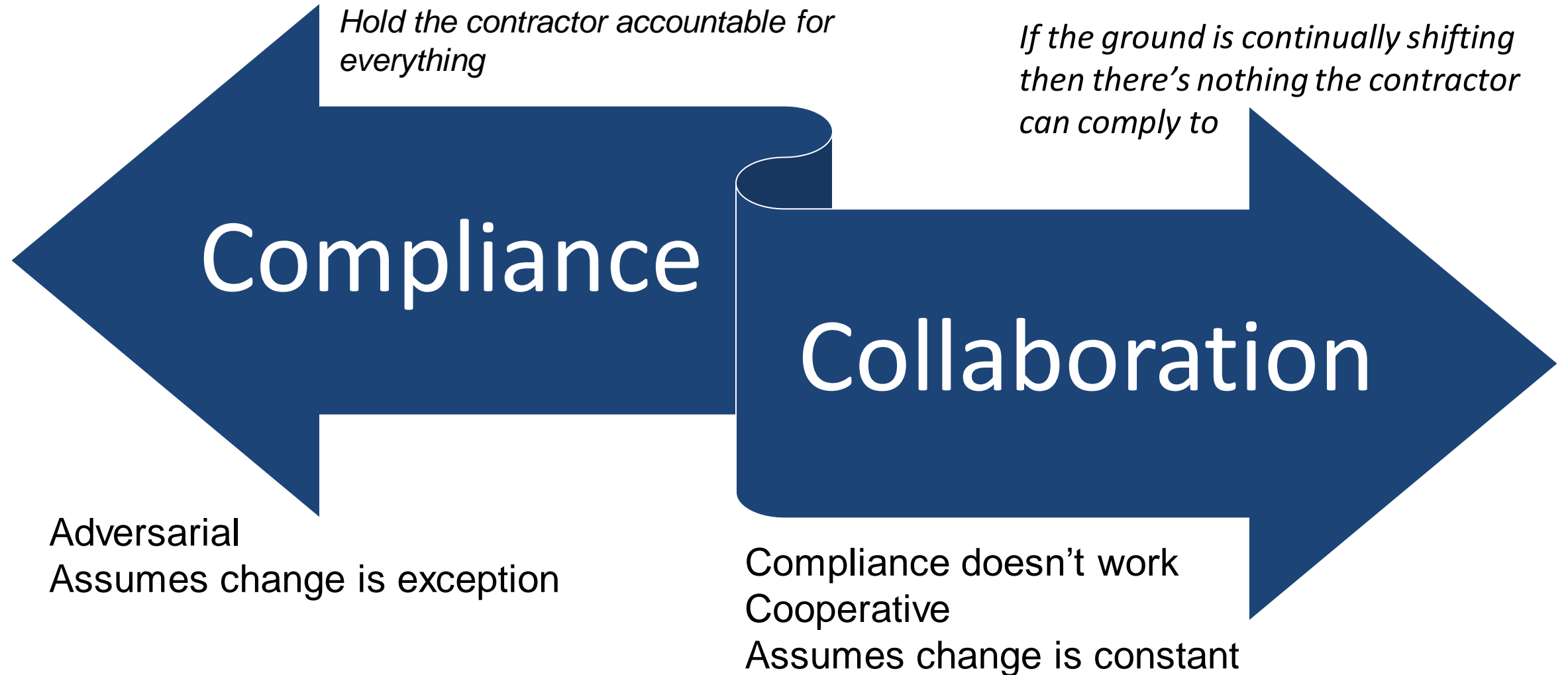
- Continually assure that the project (internal or external) is:

- following an appropriate process to the work at hand
- providing sufficient evidence that the process, as executing, can feasibly deliver target capabilities within available resources
- providing evidence that the appropriate standards are being met
- managing the capability requirements and that they remain needed and feasible

- Continually assure that the project (internal or external) is:

- correctly understanding the environment's requirements
- providing adequate user participation to support validation activities and prevent rework
- providing valid feasibility evidence for oversight
- executing an appropriate process
- periodically reviewing its process to evolve as necessary
- managing the relationships, communications and interoperability within the project and with other projects

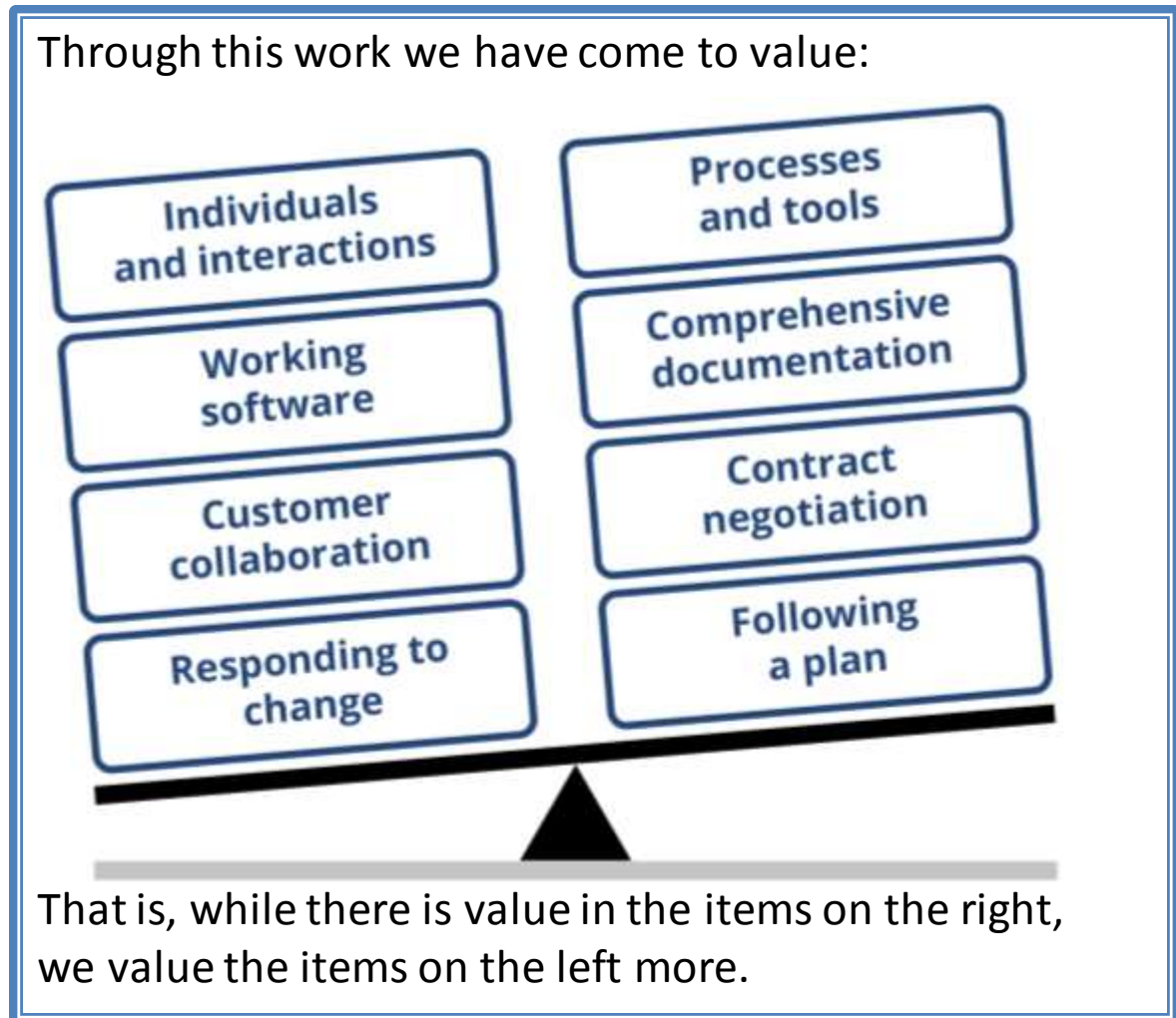
World View Determines How We See the Need for Insight



Reorienting the Manifesto for Agile *Software Development* Toward *System Acquisition*

We know the Agile Manifesto for Software Development works well to orient software development teams toward Agile....

What needs to change to orient system acquisition teams towards Lean and Agile methods?



<https://agilemanifesto.org/history.html>

Agile Acquisition Notional Manifesto for System Acquisition

These four areas are key to shifting from oversight to insight:

- Batch size
- Feedback approach
- Requirements expression and management
- Compliance/Insight mindset



The Oversight/Insight Challenge for the Lean Program Office

*How can the Program Office Gain
Insight into Contractor Expression
Of the Agile & Lean Principles?*

We are just hitting the tip of the iceberg on these concepts today!

From Large Batch to Small Batch

Typical Large Batch Behaviors/Mindsets:

- “Nothing is done until everything is done”
- More Work in Progress is good
- 100% utilization of resources is a goal
- Tendency to hide bad news
- -False/optimistic reporting of progress in order to justify incentive/progress payments
- Integration events are riddled with defects and are pushed out “until we think we have it right”
 - Increases number of potential defects that affect multiple areas of the system
 - Reduces confidence in system robustness
 - Harder for engineers to find sources of defects
- Tendency toward “test quality in”

Typical Small Batch Behaviors/Mindsets:

- We can learn from even small pieces being implemented/done
- “Stop starting, start finishing”
- Work in Progress is limited to enhance flow through the system
- 100% utilization of resources is recognized as limiting flow, flexibility, and work accomplishment
- Short time between when a defect is found and when it was created
 - Easier for engineering/developer to find source of defect
- LOTS of integration happening across entire system, building confidence
- Tendency to “build quality in”

From Primarily Documentation Review to Demos and other Mechanisms for User Feedback

Typical “Primarily Document Reviews”

Behaviors/Mindsets:

- Preference for larger, more infrequent demos
- Spotty participation in demos
- Requirements documents seen as “ground truth” for user needs, even when known to be superseded
- Few opportunities for feedback
- Incomplete, rushed feedback on documents
- More emphasis on “to be” documents than “as built” documents
 - Using documents to “lock down” design

Typical “Demos/Other Feedback Mechanisms”

Behaviors/Mindsets:

- Recognition that demo doesn’t EQUAL test, but INFORMS it
- Active participation in demos of small pieces of functionality
- Open, continuous feedback about both the fact of and the meaning of progress or lack thereof
- Info from demos is fed forward to testing and certification staff to ensure alignment
- Uses Defn of Done that includes certification criteria (cyber, DT/OT, ATC, ATO, etc.)
- Participation on continuous integration team by govt staff seen as a high priority

From Single Delivery of Requirements Document to Continuous Backlog Refinement

Typical “Single Delivery”

Behaviors/Mindsets:

- Long lead time to get to the requirements document delivery reduces motivation to allow for refinement after delivery
- Task-switching from one large batch review to another
 - Hard to take in the large requirements set
 - Demotivates “digging in” on the need behind the requirements
- Get as far as we can with review in time available, but not expecting complete understanding in time allowed

Typical Continuous Backlog Refinement

Behaviors/Mindsets:

- Mix of “push” and “pull” communication across govt/contractor interface on evolving refinements to requirements
 - Facilitated by workflow mgmt. tools like Jira, but both sides need to be on the same platform
- Frequent face to face/high bandwidth meetings to keep the relationship going, not just to do the refinement tasks
- Transparency among stakeholders that builds trust
- Frequent small batch prioritizations build a solid base of understanding of current state and progress

From Seeking Compliance to Seeking Insight

Typical “Seeking Compliance” Behaviors/Mindsets:

- Deadlines that don’t have clear relationship to product evolution goals
- Silo’ed relationships
 - Independence=Isolation
 - Formal vs informal handoffs of information
- Agile events must conform to traditional Program events
 - Large batch SETR events
 - PMRs are disconnected from development cadence and cause value-based work to stop-start
- No/ too little sharing of test assets – “if you know the criteria, you’ll develop to it; I lose my independence!”
- Measures collected but not used for process improvement
- “Gotcha” mindset
- Work designed to pass the audit more than deliver value

Typical “Seeking Insight” Behaviors/Mindsets:

- Don’t immediately react negatively to “bad news” – treat it as information that is meant to help make a different decision
- Informal handoffs of information where feasible/allowed
- Agile events allowed to preserve their cadence
- Lots of sharing of test/certification assets – “if you know the criteria, you’ll develop to it; that’s the goal!!!”
- Measures carefully selected and visibly used to solve problems in the process, not punish the contractor
- Collaborative mindset

This is the Beginning of a Conversation...

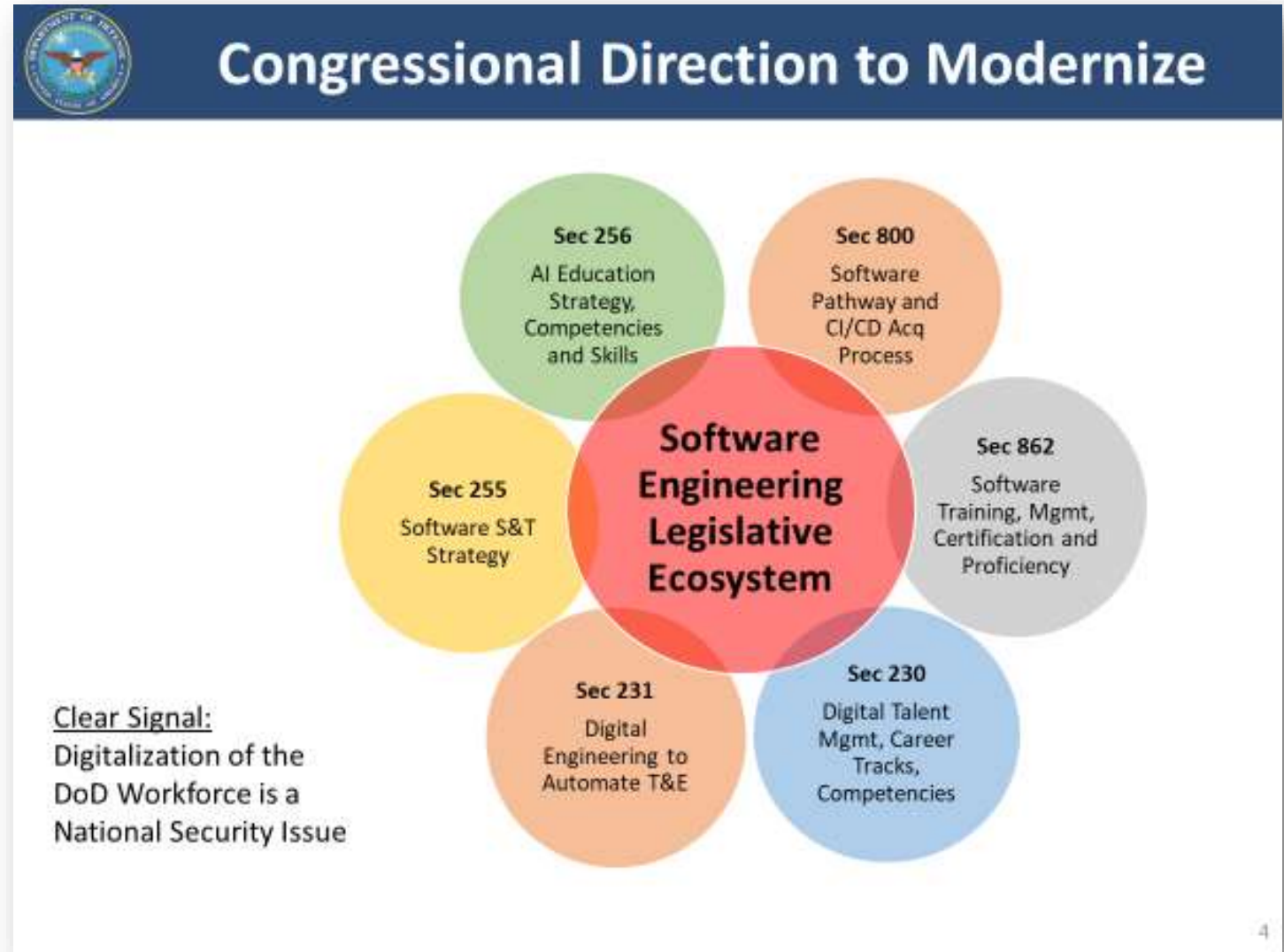
Agile

Traditional

As we move in the direction of Agile System Acquisition, we expect that a set of principles relevant to this setting can be derived, related to, but not exactly the same as those for Agile and Lean product development

Deeper Dive into Differences

Multiple Sources of SW Engineering Legislative Direction



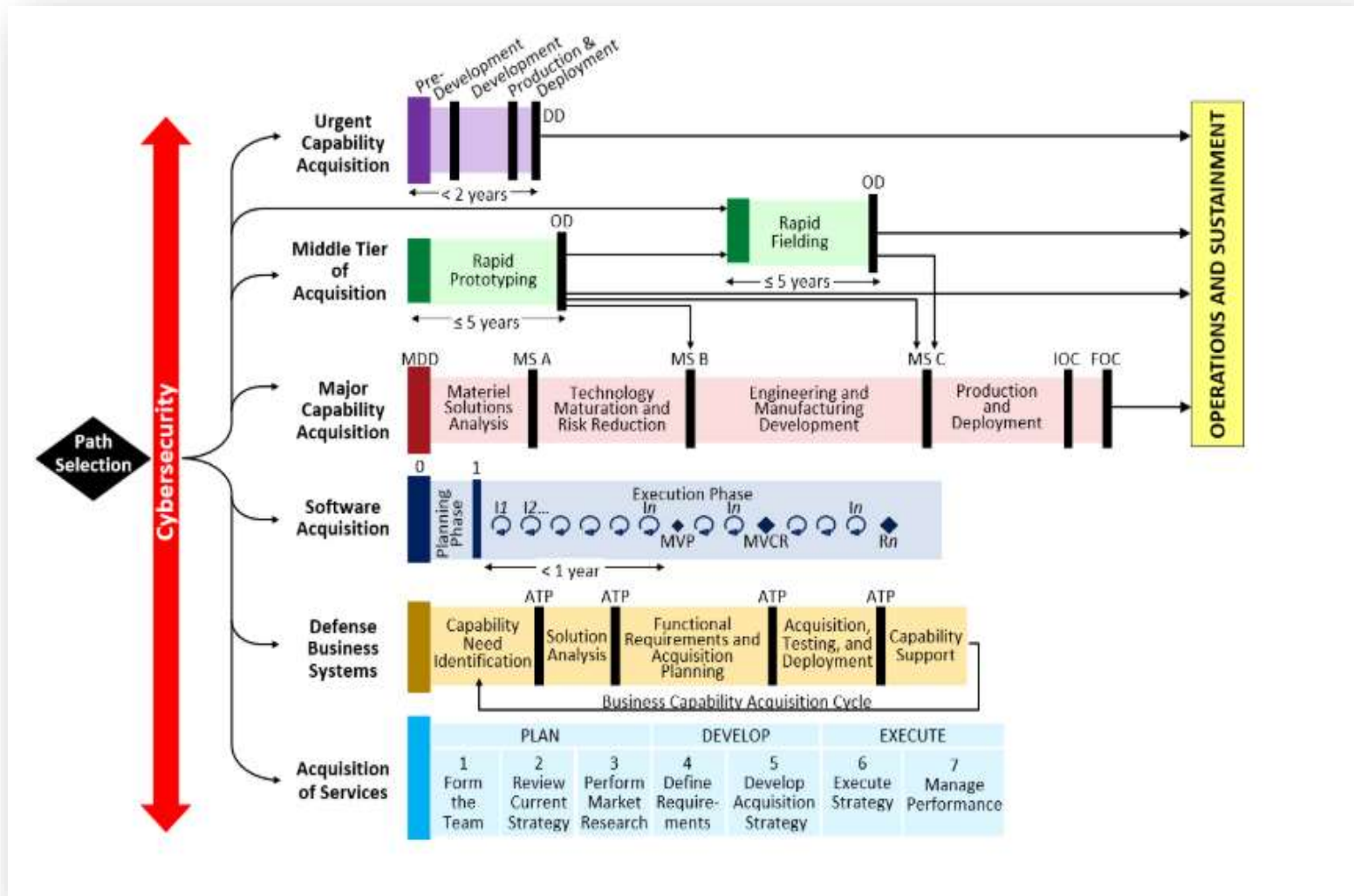
Source: Brady, Sean. *How Software Acquisition & DevSecOps Increase the Lethality of the DoD*, DSO Days, Oct 2020.

Adaptive Acquisition Pathways

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)

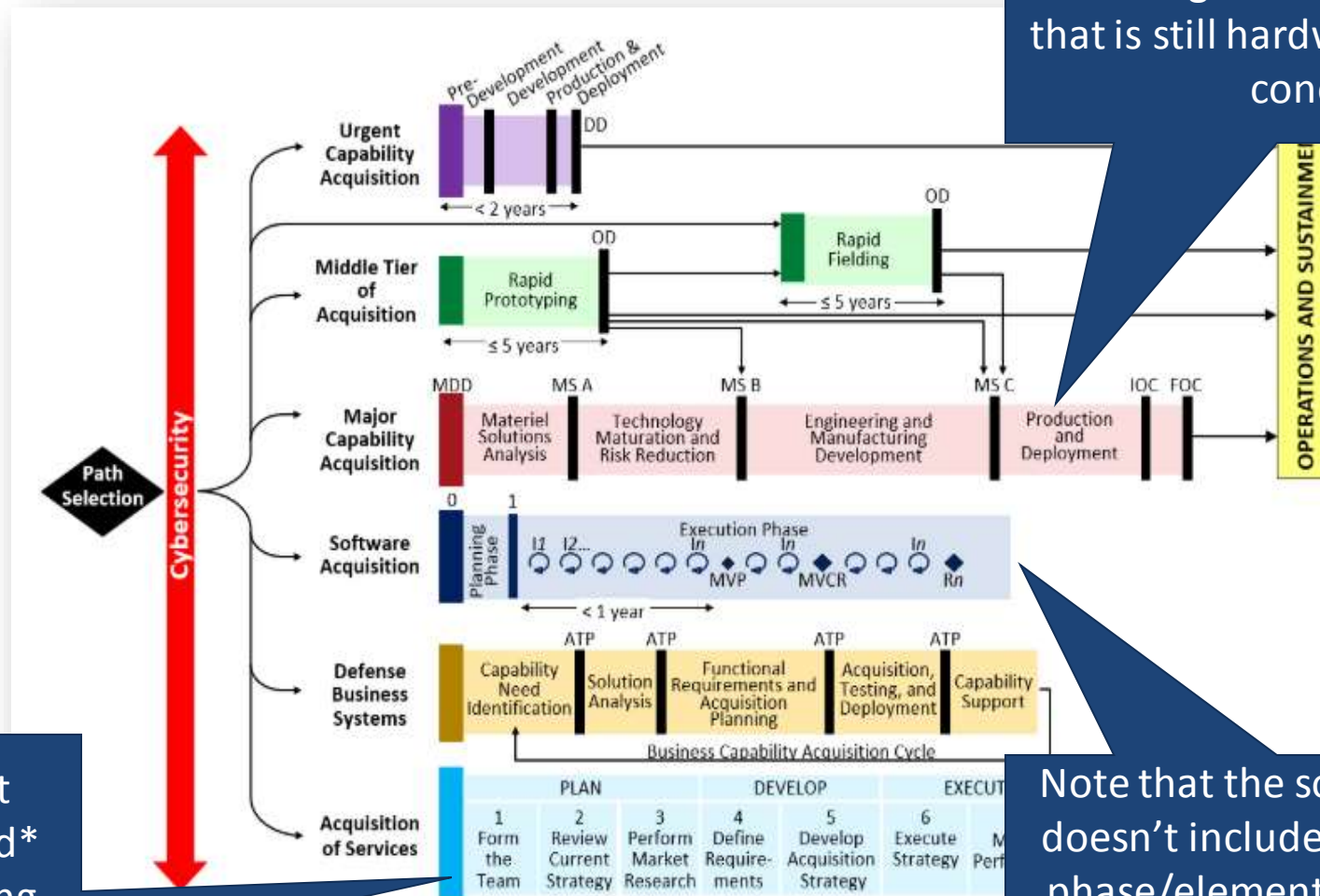
<https://aaf.dau.edu/>

You may not be using these yet, but it's likely that you will be using one or more in the future!



Selected Things to Note

A valid acquisition strategy under the pathways is to use different acquisition pathways for different contracts within the same program



Complex cyberphysical systems are still governed by a pathway that is still hardware-centric in its concepts

Not explicitly called out, but software development *could* be acquired as a service using this pathway

Note that the software pathway doesn't include an "operations" phase/element—assumption is that software evolves continuously

CDRL (Contract Data Requirements List) Items Potentially Affected by Agile in Acquisition

The “Baker’s Dozen” acquisition artifacts to review for Agile Implications

Acquisition Strategy -where you would specify using multiple pathways etc

Request for Proposal Statement of Work – where most people focus their communication about Agile

Request for Proposal Technical Evaluation

Criteria – these need to reflect what is in the SOW

Request for Proposal CDRL- the list of required artifacts contractor must deliver; usually needs to be modified

Systems Engineering Plan (SEP) –where overall system development process is SOW Agile intent

Test & Evaluation Master Plan (TEMP)- if TEMP isn’t modified to reflect Agile strategy, there will be a large bottleneck of items waiting for “end of dev” testing

CDRL DIDs (1423s) – descriptions of what the CDRL items look like and how they should be delivered; usually need modification

Technical Review Guidance – often embedded in the SEP, need to make sure that incremental reviews are enabled

Measurement Specification – traditional measurement spec likely conflicts with Agile approach

Life Cycle Sustainment Plan – especially when govt “Own the Technical Baseline” is in play

Incentive/Award Fee Plan – if incentives are misaligned, behavior will not be Agile

Integrated Management Plan/Schedule – if the IMP/IMS don’t reflect a capability basis, Agile will be difficult

Not Exactly an Acquisition Document...

Government staffing model –

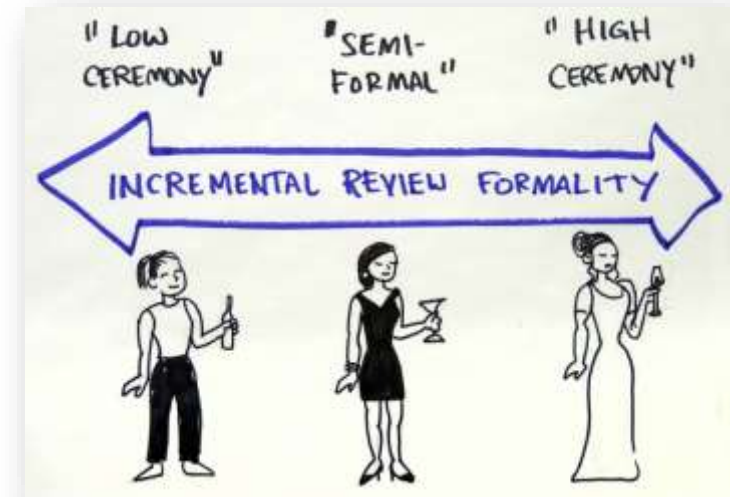
- Moving from large batch to small batch interactions has staffing implications for govt
 - Training
 - Accounting for staff that are working both Agile and Traditional acquisitions
 - Scheduling

Technical Reviews in Agile Government Settings

Agile View of Formal Technical Reviews

Agile in Government settings – have to recognize that changing the acquisition life cycle expectations doesn't happen overnight. Some accommodations have been successfully used

- Small batch events (around 12 weeks of work, typically) that reduce risk by producing working software/product, as well as required documentation
- Exit criteria for formal large-batch reviews incorporated into the small batch events
- Depending on system context, traditional System Requirements Review and even System Preliminary Design Review may occur more traditionally



Reference:

<https://apps.dtic.mil/sti/pdfs/ADA609878.pdf>

Focus of gaining understanding by learning from as built software vs projective documents

Requirements in Agile Government Settings

From Single Large Batch Requirements Document to Roadmaps and Variable Detail Backlogs

“Fixed” intent assume stability

- Higher level of requirements baseline needed to allow learning to occur
- Roadmaps guide *what* needs to be specified in more detail
- Roadmaps guide *when* higher level requirements need to be detailed

“Variable” intent assumes learning and evolution of the environment

- The trade space where requirements backlogs are reprioritized continually
- Allows focusing on the highest risk, most valued items earlier

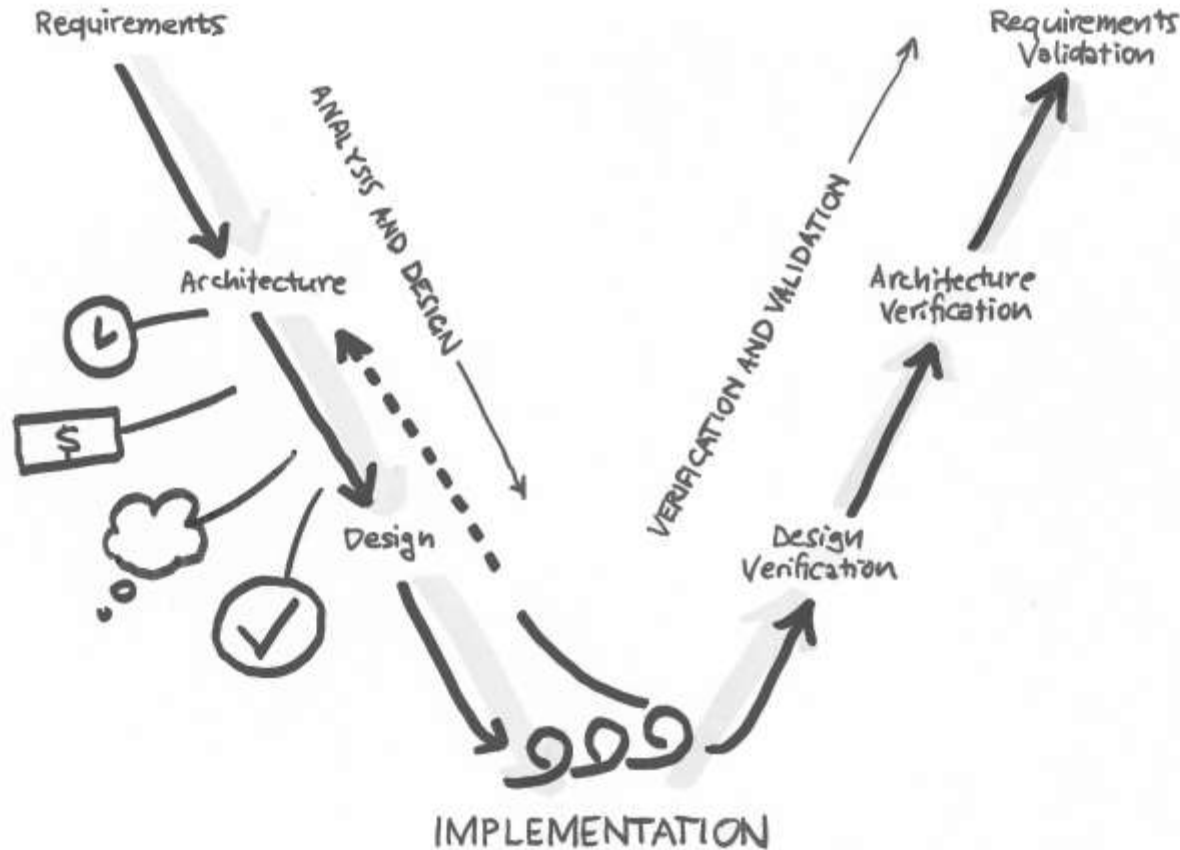
Ref:

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=89158>



Systems Engineering in Agile Government Settings

What Happens When “Large Batch” Systems Engineering Meets “Small Batch” Agile SW Development?



Traditional systems engineering creates too much “false positive feasibility” sentiment if system implementation is lagging

Systems Engineering V turns into more of a “spinning V” or a set of its own iterative cycles that interact with implementation and validation much more frequently than traditional

Agile at the Bottom of the V Video:

<https://www.youtube.com/watch?v=ZdSodMDDVh8>

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=427580>

Contracting in Agile Government Settings

Properties of Successful Agile Project Teams

1. Frequent delivery
2. Reflective improvement
3. Close/Osmotic Communication
4. Personal Safety (free to speak without fear of reprisal)
5. Focus (knowing what to work on, having time to work on it)
6. Easy access to expert users
7. Sunshine/Visibility (no dark places in the project) (courtesy Sam Person, Overstock.com)
8. Technical environment

Agile Minute video on RFPs:

https://www.youtube.com/watch?v=Wdpk_o5fYBo

[Agile Contracting TN:](#)

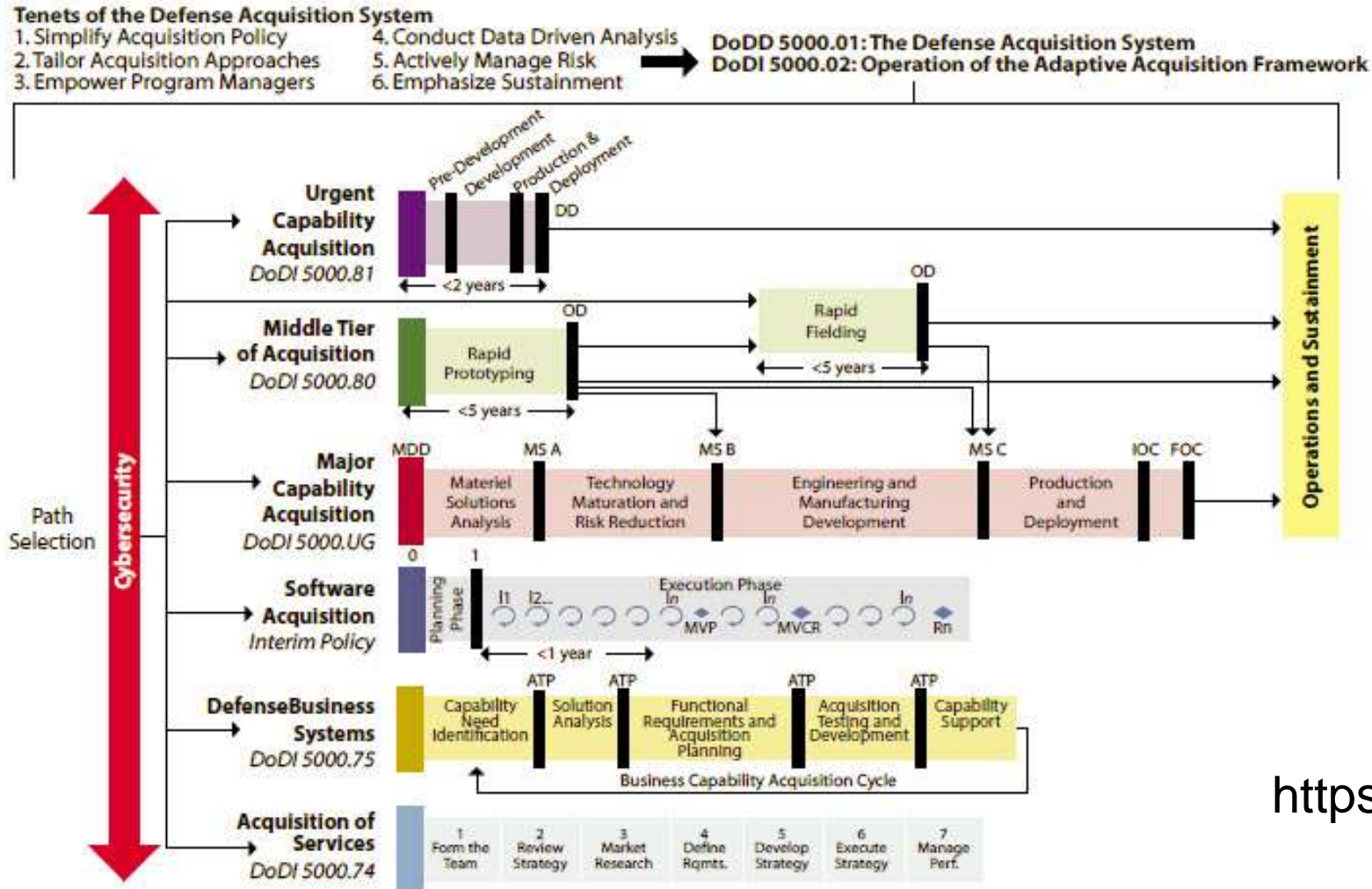
<http://docshare01.docshare.tips/files/27506/275060628.pdf>

Crystal Clear: A Human-Powered Methodology for Small Teams, Alistair Cockburn, 2005 Pearson Ed cc 19 - 37

The contracting challenge is how to be aware of, and incentivize these and other Agile/Lean attributes we've discussed

This is an Area Changing More than some Others

Figure 3. Adaptive Acquisition Framework



<https://aaf.dau.edu/>

Measuring Progress in Agile Government Settings

The Consistent Measurement Challenge is the Multi-Variable Nature of Causality

Understanding cause-and-effect in a multi-variant universe is challenging

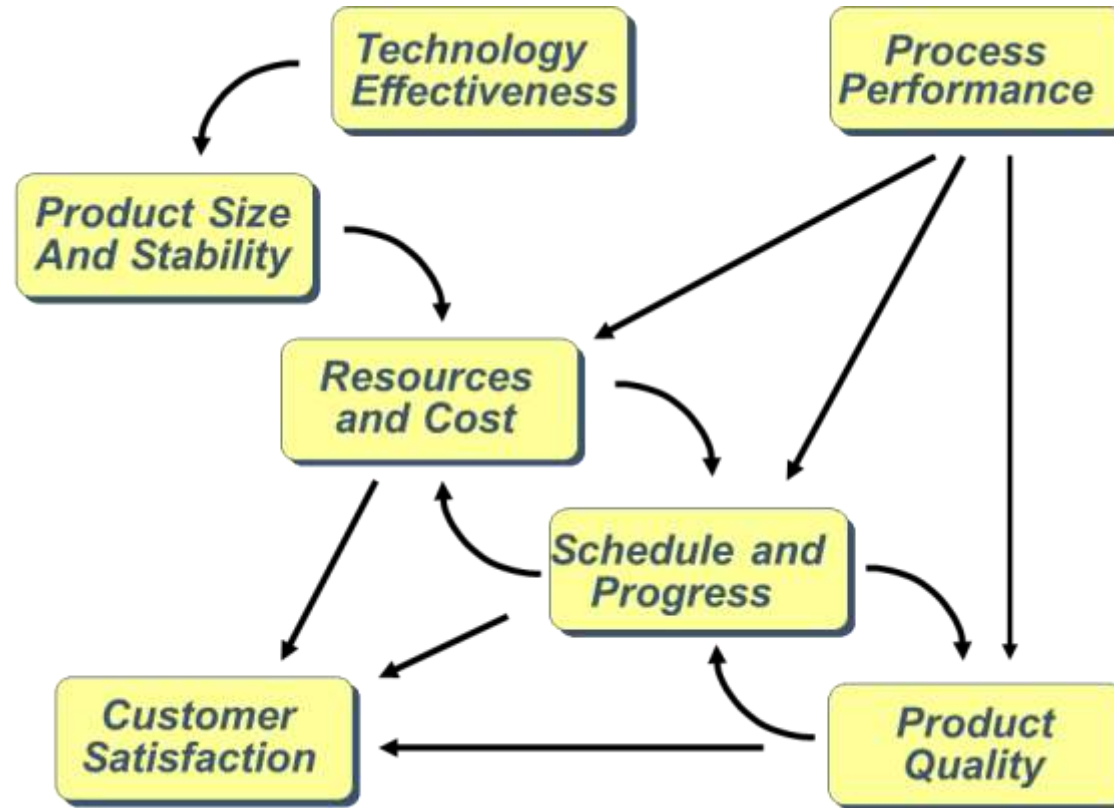
- A pre-requisite to acting on progress measures.

Agile & Lean thinking drives to new areas of focus:

- Flow vs utilization
- Value
- Concept to Capability

[Will Hayes Webinar on Agile Metrics in DoD:](https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503751)

<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503751>



Graphic Adapted from: *Practical Software & Systems Measurement*

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

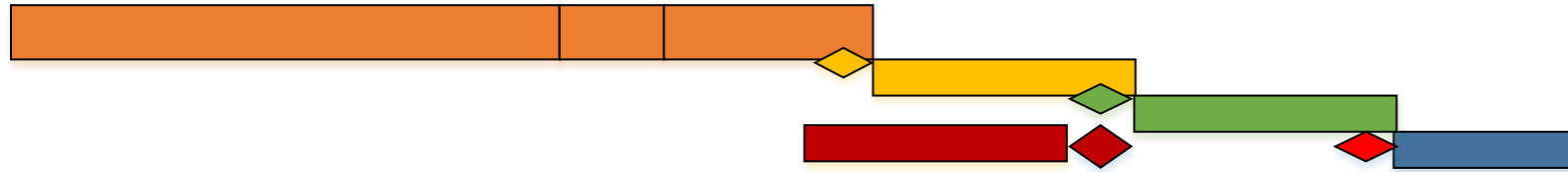
[Agile Metrics TN:](https://www.researchgate.net/profile/Timothy_Chick/publication/266049888_Agile_Metrics_Progress_Monitoring_of_Agile_Contractors/links/542416830cf26120b7a71853/Agile-Metrics-Progress-Monitoring-of-Agile-Contractors.pdf)

https://www.researchgate.net/profile/Timothy_Chick/publication/266049888_Agile_Metrics_Progress_Monitoring_of_Agile_Contractors/links/542416830cf26120b7a71853/Agile-Metrics-Progress-Monitoring-of-Agile-Contractors.pdf

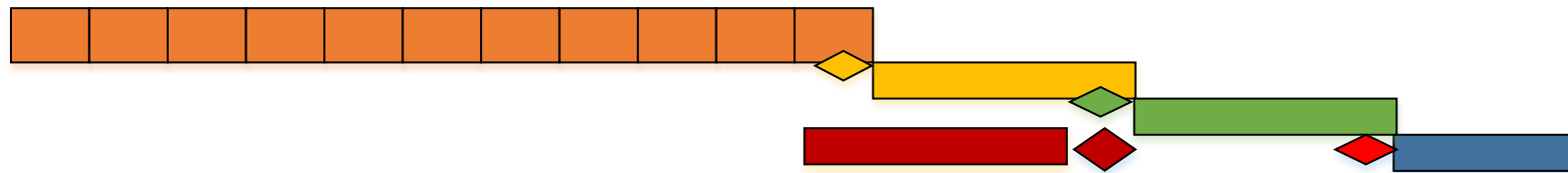
Testing in Agile Government Settings

Left-shift with Agile Testing

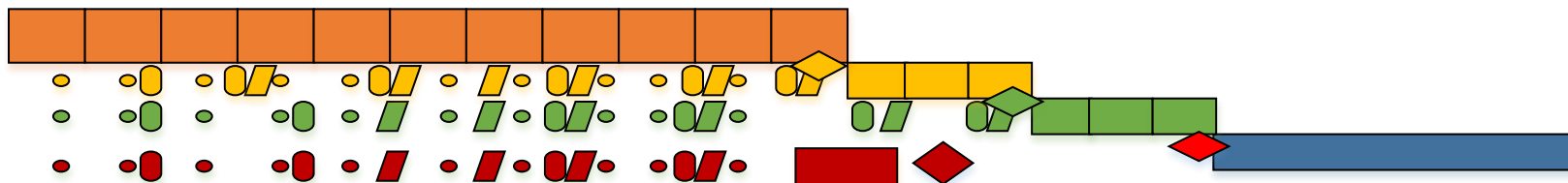
Traditional Vee-process



Agile development with traditional DT and OT (Hybrid)



Agile development with traditional DT and OT, early integration synch points



Moving from phased and siloed testing to Agile testing is the “Big Deal”

Integrating Agile cadence with DT/OT and certification is a key challenge

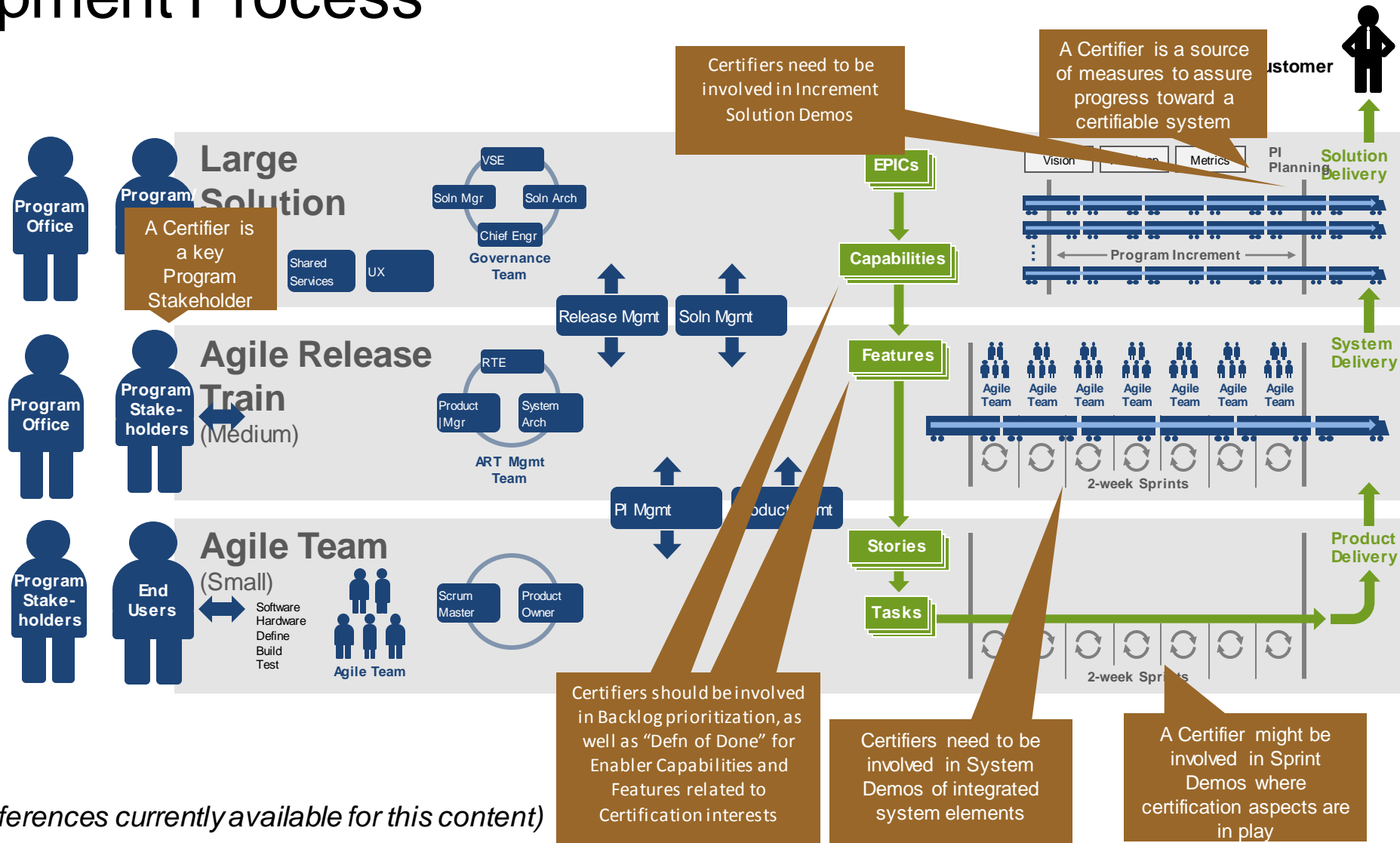
[Binder/Miller webinar on automated test:](https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503507)
<https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=503507>

Shift Left Testing the Agile Way article, ITEA journal
<https://www.researchgate.net/publication/315756100>



Certification (Cyber, Airworthiness, Nuclear Surety...) in Agile Government Settings

Certifiers Need to be Involved Throughout an Agile/Lean Development Process



(no Technical Note references currently available for this content)

Ex 2: Challenges of Agile Acquisition

On the flip chart, add sticky notes with your answers to the question:

What do you think is the most challenging aspect of what's different between Agile and traditional acquisition?



Summary/What else do you want to learn?

Moving from Traditional Acquisition to Agile-based Acquisition is a Big Change for Most Program Offices

Agile *Traditional*

Changes in skill profiles

Changes in staffing curves

Changes in character of interactions
with contractors and stakeholders

Changes in batch size

....

The payoff: faster delivery of certified, high value solutions to warfighters and our other stakeholders.

What Else Do You Want to Learn?



Exercise: Add sticky notes to the flip chart answering:

What topics that we **did** address do you want more information on? What topics that we did **not** address do you want more information on?

THANK

YOU!!!!

Contact Information

Suzanne Miller

Principal Researcher

SSD/CDC/ATT

Email: smg@sei.cmu.edu

Bart Hackemack

Senior Member, Technical Staff

SSD/CDC/ATT

Email: blhackemack@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

Telephone: +1 412-268-5800

Fax: +1 412-268-6257

Backup slides

Principles from the Agile Manifesto for Developers

1. Highest priority is satisfy the customer through early and continuous delivery of software.
2. Welcome changing requirements, even late in development...
3. Deliver working software frequently, from a couple of weeks to a couple of months...
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Provide environment and support they need...
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...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.

Remember these principles are small team (6-10 developers) focused!

Don't Forget the Lean Principles – For the Entire (Govt + Contractor + Stakeholder) Enterprise

Apply SAFe Lean-Agile 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

#10 Organize around value