Carnegie Mellon University
Software Engineering Institute

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution]

Agile in Government: Go for Insight, Not Just Oversight

July 2020

Software Engineering Institute
Continuous Deployment of Capability Directorate

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.

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.

Although the rights granted by contract do not require course attendance to use this material for U.S. Government purposes, the SEI recommends attendance to ensure proper understanding.

DM20-0409

Topics in this presentation include:

- **A Notional Manifesto for Agile Acquisition**
- **Traditional Things We Think About Differently when Going for Insight**

Bottom Line Up Front

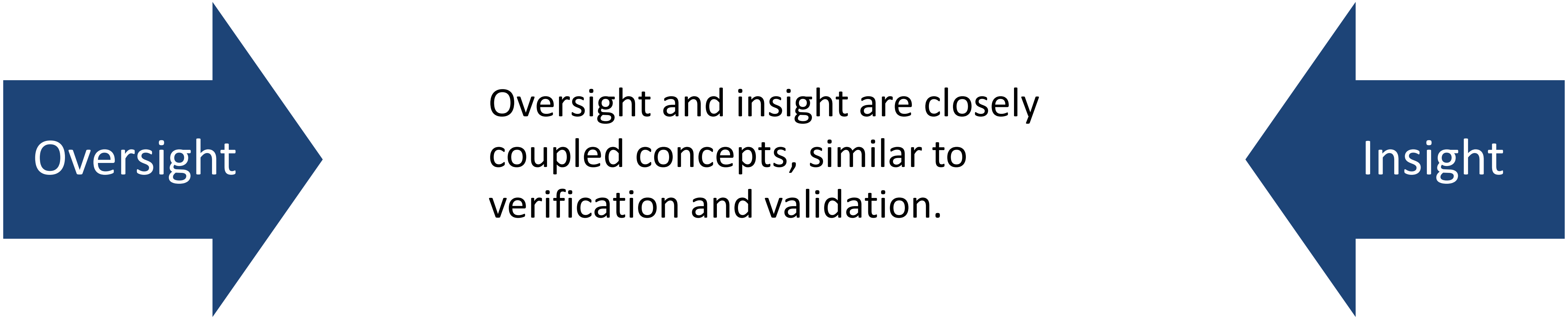
There is a great deal of **OVERSIGHT** activity that is required by the traditional DoD acquisition eco system to progress a program, software or otherwise

- Many of the mechanisms used for acquisition oversight could be seen as substitutes for the communication that naturally occurs in a trust-based relationship typical of Agile settings
- Regardless of the informal communication on the program, required oversight has to be accomplished
  - Recent developments like Adaptive Acquisition Pathways change some of those requirements

The other goal for contract monitoring is to achieve **INSIGHT** into the program

- Acquisition CDRLS and required events are not always the best way to achieve insight
- Agile development settings, in particular, promote transparency and have built in mechanisms for achieving ongoing insight
  - These mechanisms, however, require proactive participation from the acquirer to be effective

Oversight/Insight

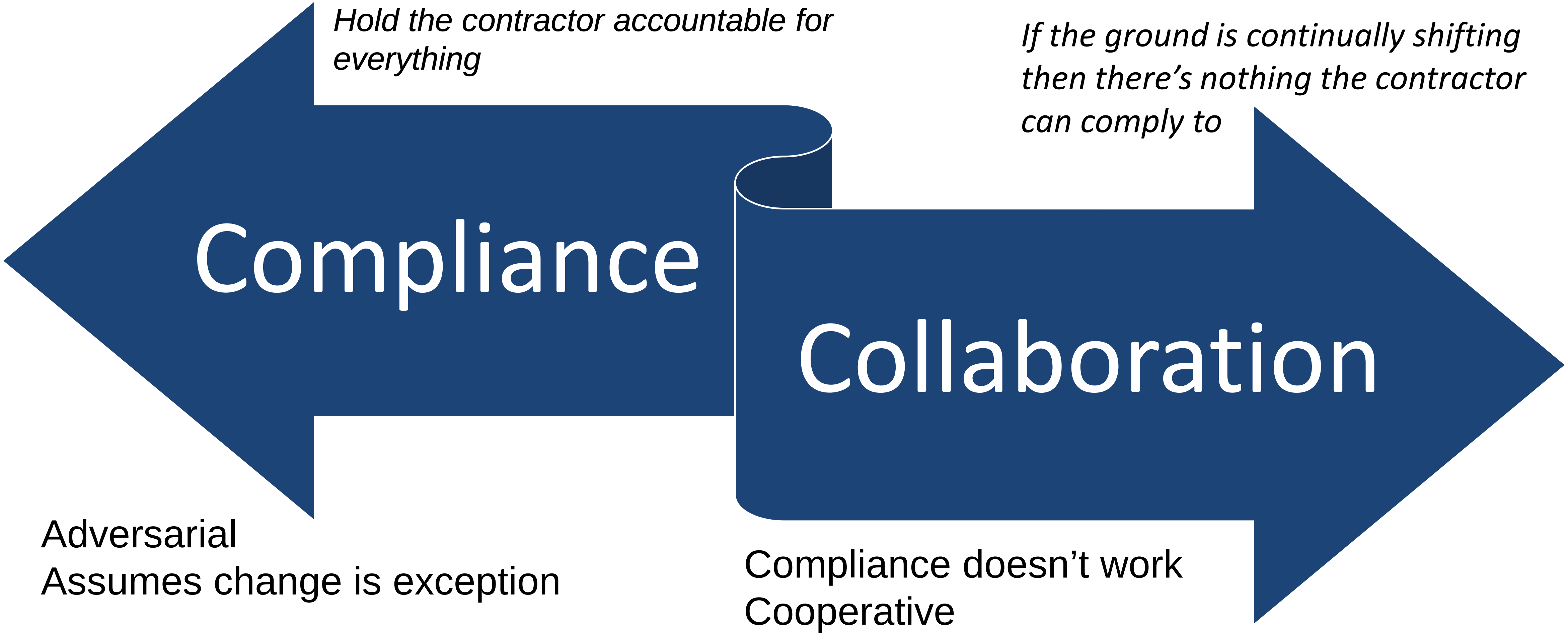


Oversight and insight are closely coupled concepts, similar to verification and validation.

- 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



Hold the contractor accountable for everything

If the ground is continually shifting then there's nothing the contractor can comply to

Adversarial
Assumes change is exception

Compliance doesn't work
Cooperative
Assumes change is constant

Back to the Principles…

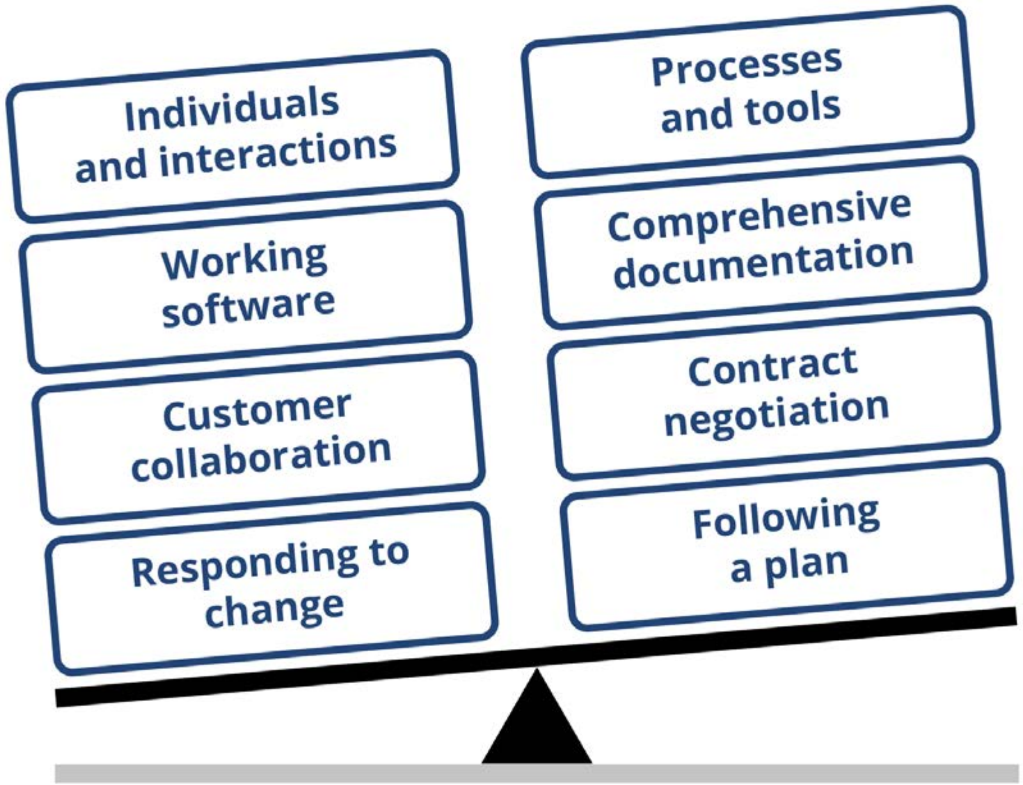
Agile and Lean

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?

Through this work we have come to value:



Individuals and interactions

Working software

Customer collaboration

Responding to change

Processes and tools

Comprehensive documentation

Contract negotiation

Following a plan

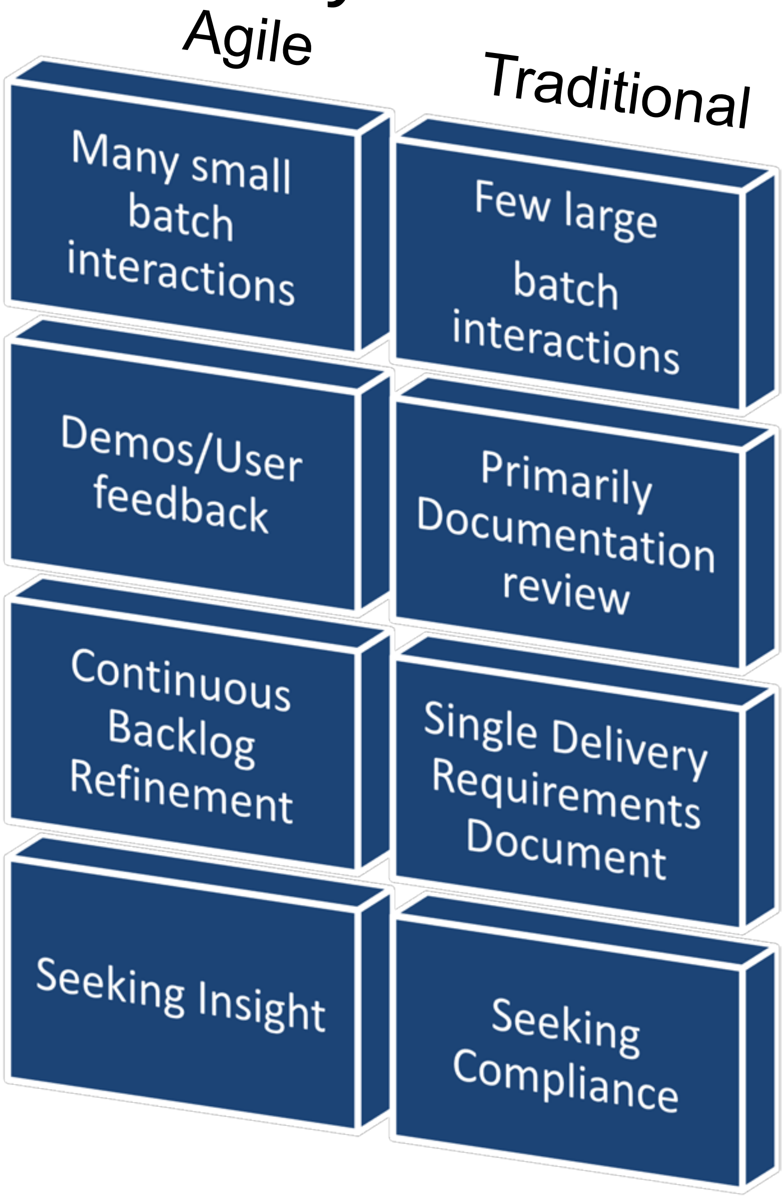
That is, while there is value in the items on the right, we value the items on the left more.

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

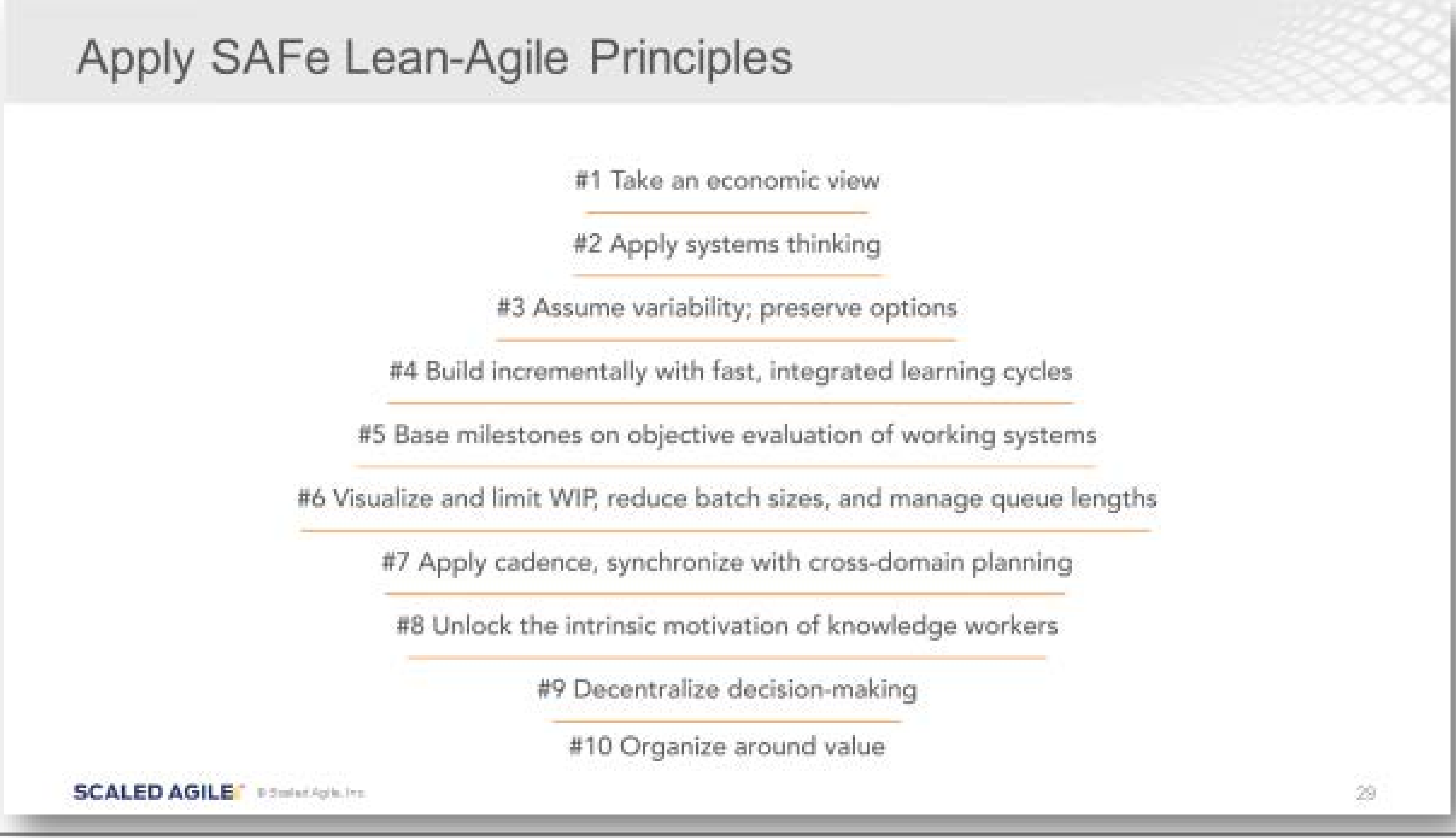

Agile
Traditional
Many small batch interactions
Few large batch interactions
Demos/User feedback
Primarily Documentation review
Continuous Backlog Refinement
Single Delivery Requirements Document
Seeking Insight
Seeking Compliance

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

SCALED AGILE® © Scaled Agile, Inc.

29

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:

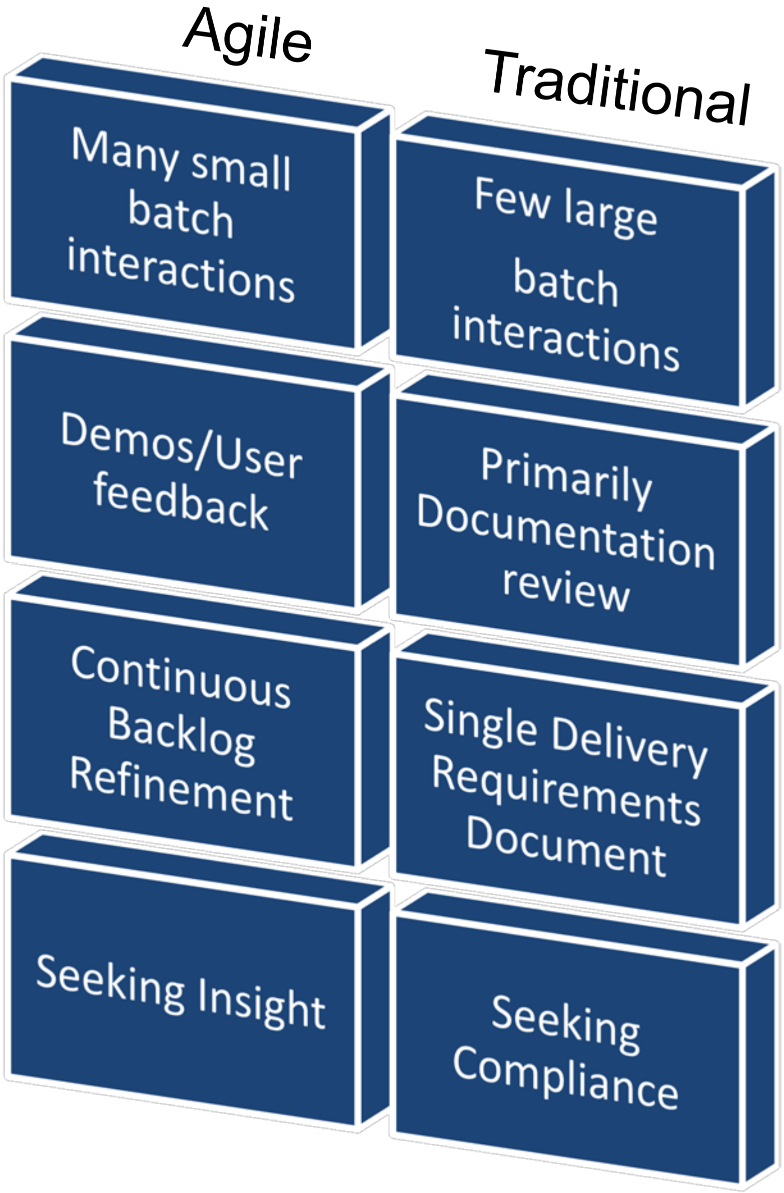
- Deadlines that don't have clear relationship to product evolution goals
- Silo'ed relationships
  - Independence=Isolation
  - Formal vs informal handoffs of information
- Agile events must conform to traditional Program events
  - Large batch SETR events
  - PMRs are disconnected from development 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…

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



Agile	Traditional
Many small batch interactions	Few large batch interactions
Demos/User feedback	Primarily Documentation review
Continuous Backlog Refinement	Single Delivery Requirements Document
Seeking Insight	Seeking Compliance

Traditional Things We Still Need to Think About…but Differently

BLUF for each of these addressed briefly:

- Technical Reviews
- Requirements
- Systems Engineering
- Contracting
- Measures
- Testing
- Certification

Note: The “one page summaries” that follow are expanded into 90 minute modules as part of SEI’s Virtual Schoolhouse offering, see here:

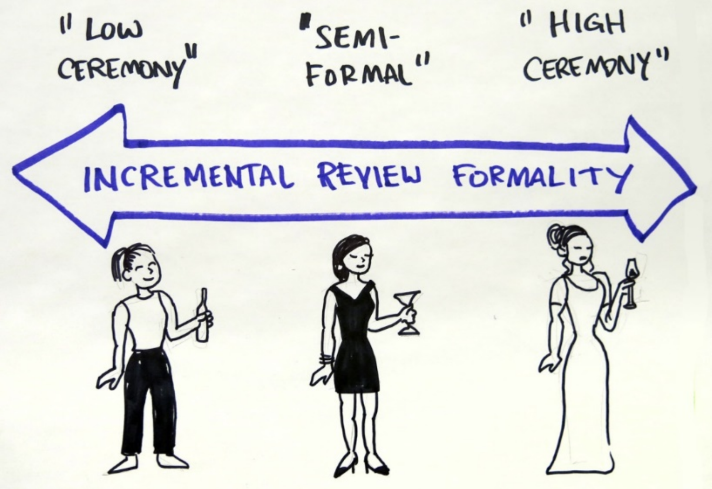
https://www.sei.cmu.edu/education-outreach/courses/course.cfm?courseCode=P145

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



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

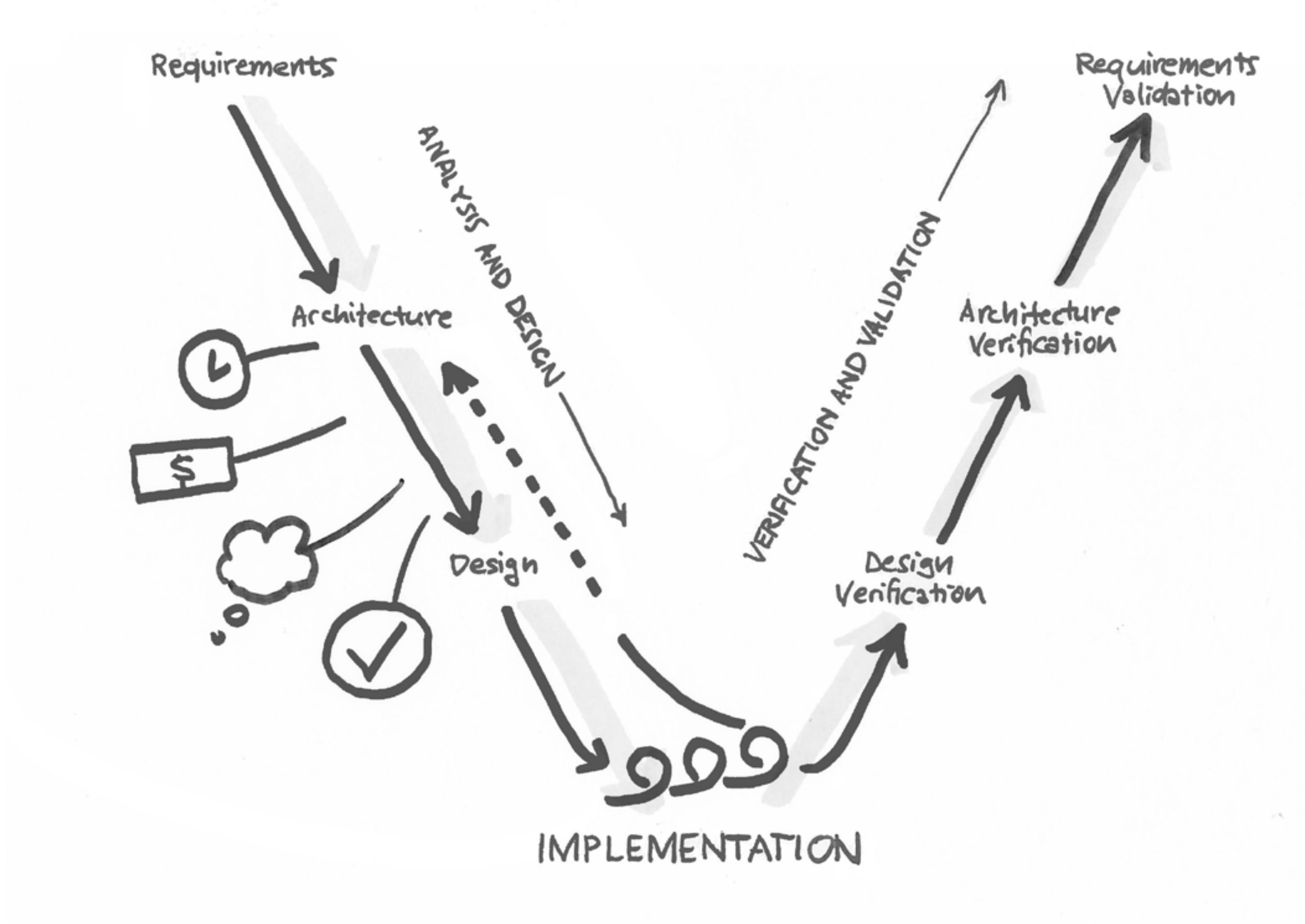
“Variable” intent assumes learning and evolution of the environment

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

If it’s IN the backlog, it MIGHT get done…
If it ISN’T in the backlog, it WON’T get done!

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

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

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

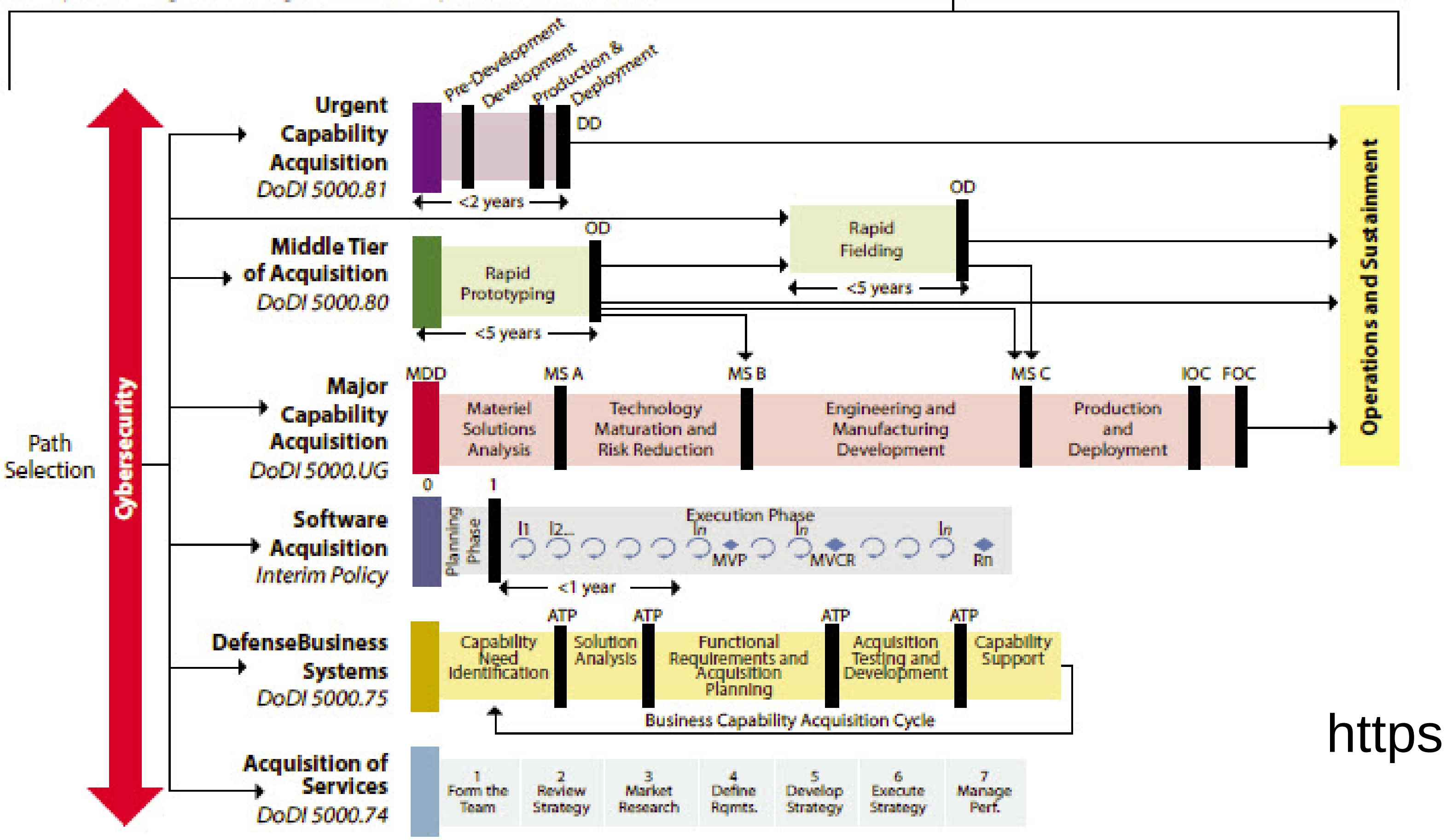
This is an Area Changing More than some Others

Figure 3. Adaptive Acquisition Framework

Tenets of the Defense Acquisition System

1. Simplify Acquisition Policy
2. Tailor Acquisition Approaches
3. Empower Program Managers
4. Conduct Data Driven Analysis
5. Actively Manage Risk
6. Emphasize Sustainment

DoDD 5000.01: The Defense Acquisition System
DoDI 5000.02: Operation of the Adaptive Acquisition Framework



https://aaf.dau.edu/

Measuring Progress in Agile Government Settings

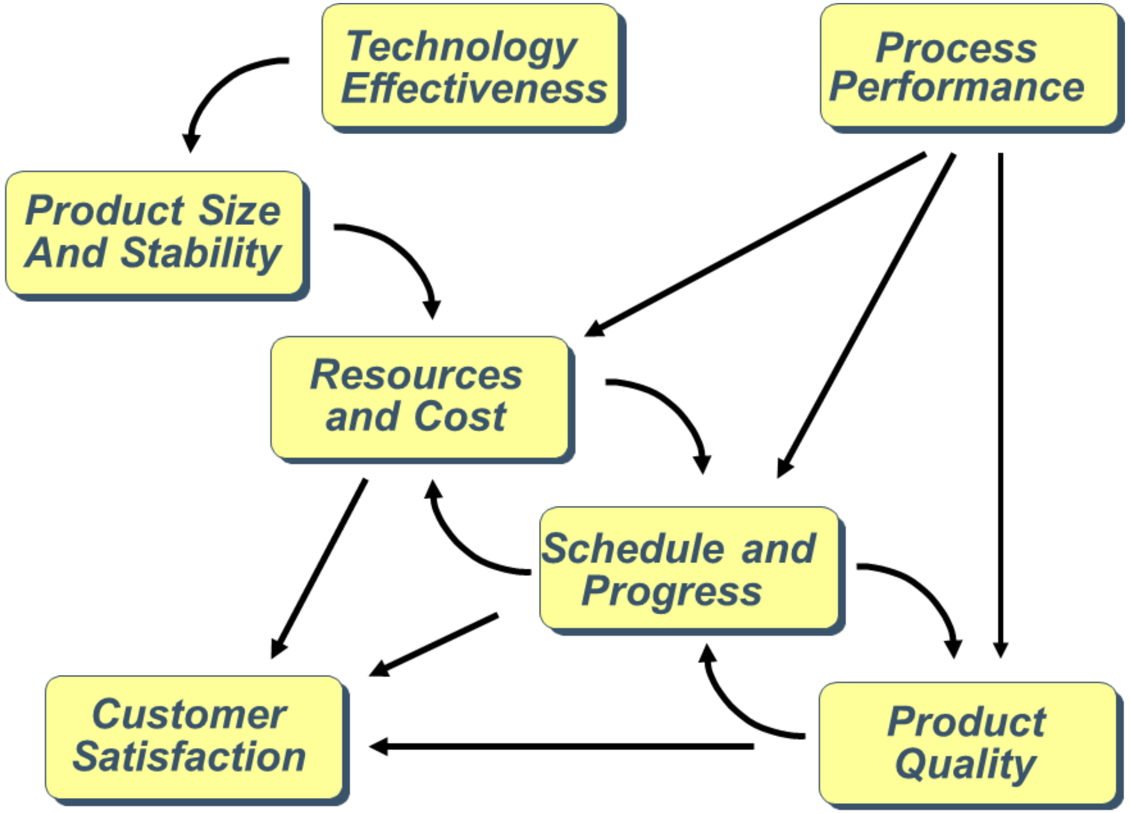
The Consistent Measurement Challenge is the Multi-Variable Nature of Causality

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:

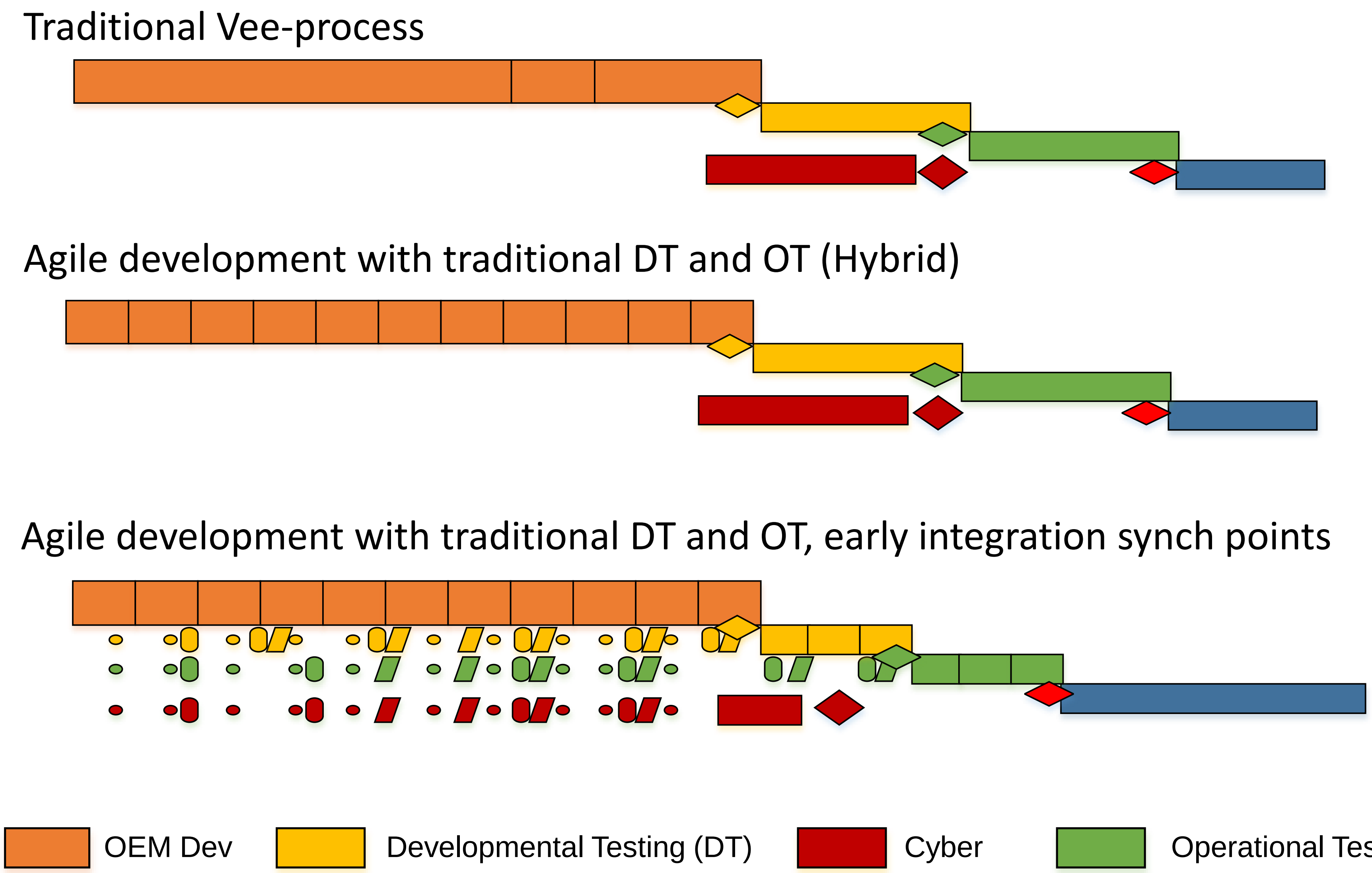
- Flow vs utilization
- Value
- Concept to Capability



Graphic Adapted from: *Practical Software & Systems Measurement*
http://www.psmsc.com/

Testing in Agile Government Settings

Left-shift with Agile Testing

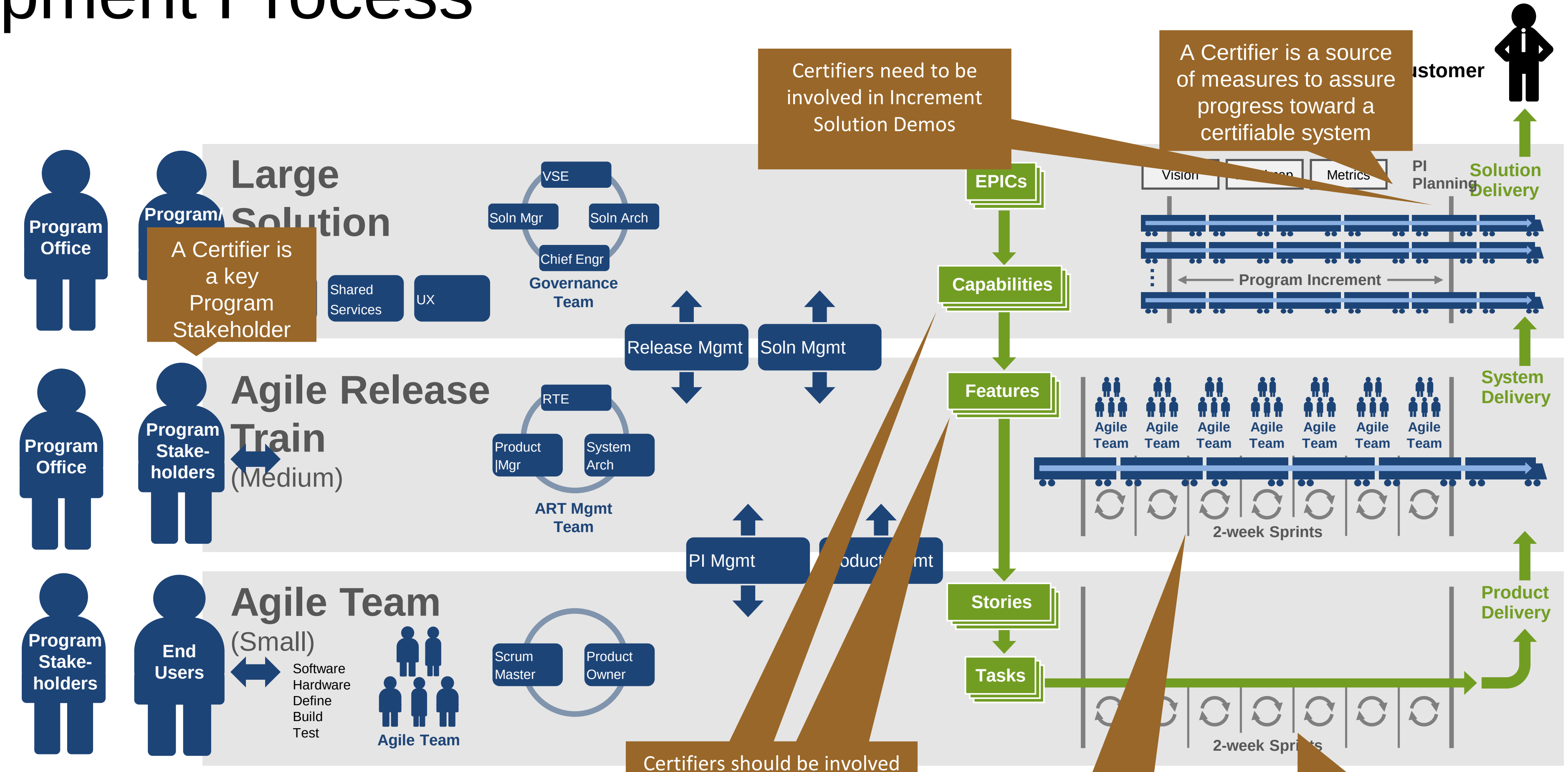

Traditional Vee-process
Agile development with traditional DT and OT (Hybrid)
Agile development with traditional DT and OT, early integration synch points
🟧 OEM Dev
🟨 Developmental Testing (DT)
🟥 Cyber
🟩 Operational Testing (OT)
🟦 Deployment

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

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

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



A Certifier is a key Program Stakeholder

Certifiers need to be involved in Increment Solution Demos

A Certifier is a source of measures to assure progress toward a certifiable system

Certifiers should be involved in Backlog prioritization, as well as “Defn of Done” for Enabler Capabilities and Features related to Certification interests

Certifiers need to be involved in System Demos of integrated system elements

A Certifier might be involved in Sprint Demos where certification aspects are in play

Final Thoughts

Moving from Oversight to Insight is a Big Shift for Many Program Offices

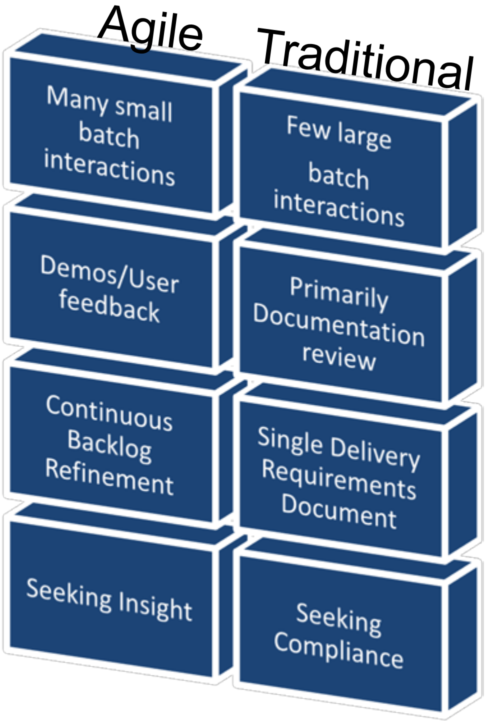
Changes in skill profiles

Changes in staffing curves

Changes in character of interactions with contractors and stakeholders

Changes in batch size

….



Agile	Traditional
Many small batch interactions	Few large batch interactions
Demos/User feedback	Primarily Documentation review
Continuous Backlog Refinement	Single Delivery Requirements Document
Seeking Insight	Seeking Compliance

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

THANK YOU!!!!

Contact Information

Suzanne Miller
Principal Researcher
SSD/CDC
Email: smg@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