

# Agile Acquisition of Agile C2

Software Engineering Institute  
Carnegie Mellon University  
Pittsburgh, PA 15213

Dr. Paul Nielsen  
June 20, 2012

SEI Proprietary; Distribution: Director's Office Permission Required



# Introduction

- Commanders are increasingly more engaged in day-to-day activities
- There is a rapid pace of software creation, adoption, and demand
- The U.S. Government is in support of agile adoption
  - 2010 National Defense Authorization Act
  - DoD CIO 10 Point Plan to reform DoD IT



SEI Proprietary; Distribution: Director's Office Permission Required

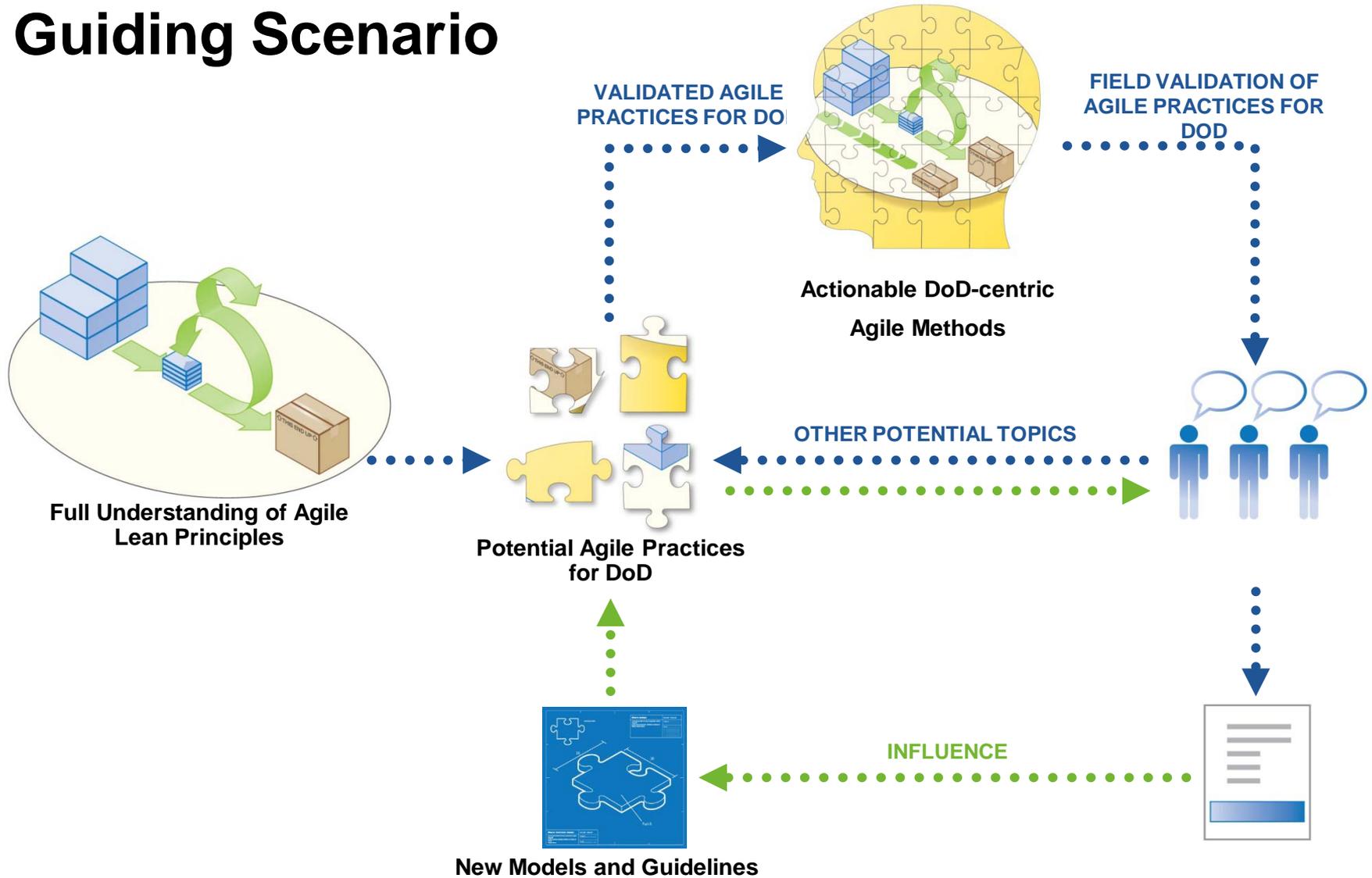


# Agile DoD Compared to Traditional DoD

Element	Agile DoD	Traditional DoD
<b>Organizational Structure</b>	<ul style="list-style-type: none"> <li>• Flexible and adaptive structures;</li> <li>• Self organizing teams,</li> <li>• Co located teams or strong communication mechanisms when teams are distributed</li> </ul>	<ul style="list-style-type: none"> <li>• Command and control structures that are difficult to change</li> <li>• Hierarchical, command and control-based teams</li> </ul>
<b>Rewards System</b>	<ul style="list-style-type: none"> <li>• Team is focus of rewards</li> <li>• Sometimes team itself recognizes individuals</li> </ul>	<ul style="list-style-type: none"> <li>• Individual is focus of the reward system</li> </ul>
<b>Communications &amp; Decision Making</b>	<ul style="list-style-type: none"> <li>• Daily stand up meetings,</li> <li>• Frequent retrospectives,</li> <li>• Information radiators to communicate critical project information;</li> <li>• Evocative documents to feed conversation;</li> <li>• “Just enough” documentation.</li> <li>• Control and discipline comes from the Agile team itself.</li> </ul>	<ul style="list-style-type: none"> <li>• Top down communication; External regulations, policies and procedures tend to drive the work. Activities and processes documented;</li> <li>• Traditional, representational documents used by the PMO throughout the development life cycle to oversee the progress and discipline of the developer through formal and informal reviews.</li> </ul>
<b>Staffing Model</b>	<ul style="list-style-type: none"> <li>• Cross functional teams including all roles across the life cycle throughout the lifespan of the project;</li> <li>• Agile advocate or coach</li> <li>• End-user representative</li> </ul>	<ul style="list-style-type: none"> <li>• Uses traditional waterfall model with separate teams, particularly for development and testing</li> <li>• Different roles (e.g. developer, tester) are active at different defined points in the life cycle and are not substantively involved except at those times</li> </ul>



# Guiding Scenario



SEI Proprietary; Distribution: Director's Office Permission Required



# Features of Agility in Acquisition

- Adopting an agile culture
- Incremental delivery
- Self organizing teams
- Flexibility
- End-user participation
- Risk management



SEI Proprietary; Distribution: Director's Office Permission Required



# Benefits of Agile Software Development

Better software can improve operational command and control by

- Ability to adjust quickly
- Ability to be responsive to changing customer needs
- “Uncertainty is inherent in the process of software development,” (Atkinson)
- Earlier insight into development problems
- More personal commitment to project

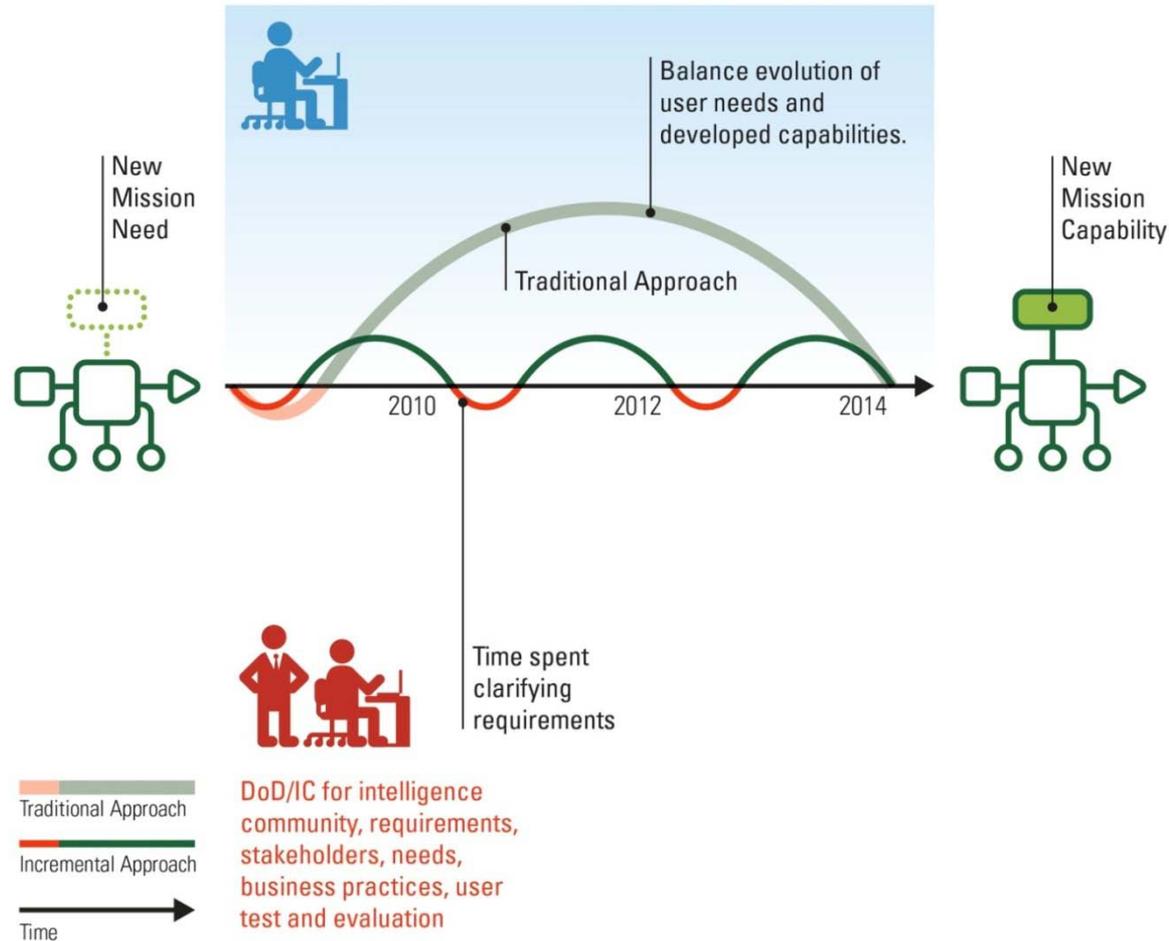


SEI Proprietary; Distribution: Director's Office Permission Required



# Innovation and Agility in Acquisition

Systems and Software Engineering  
Expertise and Framework



SEI Proprietary; Distribution: Director's Office Permission Required



# Obstacles to Agility in Acquisition

- Long DoD timelines
- Traditional contract constraints
- Culture shifts
- Rigid DoD requirements and oversight
- The element of the unknown
- Information overload
- Communication gaps
- Scaling and Architecture



SEI Proprietary; Distribution: Director's Office Permission Required



# How to Implement Agility in Acquisition

- DoD Oversight
  - Examine the formal and informal acquisition process
  - Examine the requirement process
  - Require cultural shift
  - Retool contracts for agile acquisition
  - Research how to scale agility
- PEOs and PMs
  - Push back when needed
  - Understand requirements and intent of requirements
  - Stay connected with users

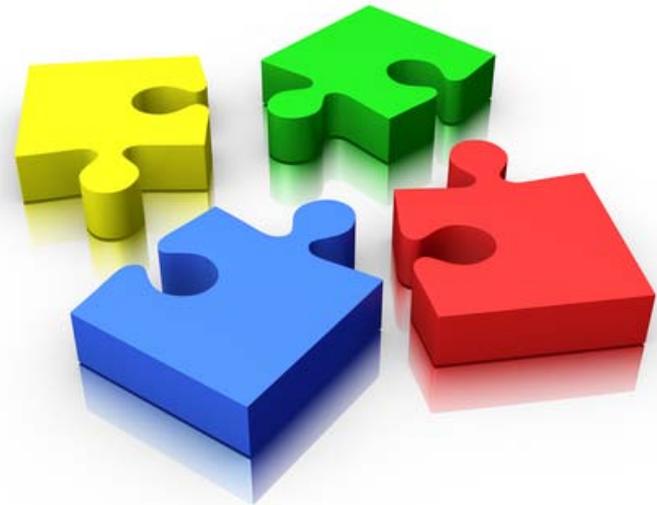
---

SEI Proprietary; Distribution: Director's Office Permission Required



# How to Implement Agility in Acquisition

- Engineers and Developers
  - Demand more authority in setting schedule, resources
  - Commit to plan
  - Produce to schedule
  - Learn from each iteration, improve ability to plan and produce
  - Where possible live in the users shoes



SEI Proprietary; Distribution: Director's Office Permission Required



# Conclusion



SEI Proprietary; Distribution: Director's Office Permission Required



# Contact Information

**Dr. Paul Nielsen**  
Director and CEO

## Web

[www.sei.cmu.edu](http://www.sei.cmu.edu)

[www.sei.cmu.edu/contact.cfm](http://www.sei.cmu.edu/contact.cfm)

## U.S. Mail

Software Engineering Institute  
Customer Relations  
4500 Fifth Avenue  
Pittsburgh, PA 15213-2612  
USA

## Customer Relations

Email: [info@sei.cmu.edu](mailto:info@sei.cmu.edu)

Telephone: +1 412-268-5800

SEI Phone: +1 412-268-5800

SEI Fax: +1 412-268-6257

---

SEI Proprietary; Distribution: Director's Office Permission Required



Copyright 2012 Carnegie Mellon University.

This material is based upon work 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.

This material has been approved for public release and unlimited distribution except as restricted below.

Internal use:\* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

External use:\* 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 external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at [permission@sei.cmu.edu](mailto:permission@sei.cmu.edu).

\*These restrictions do not apply to U.S. government entities.

---

SEI Proprietary; Distribution: Director's Office Permission Required

