Panel: Experiences with Agile for Systems Engineering in the Defense Industry

Moderator: Mary Ann Lapham, PMP, CSM
Principal Engineer
Software Engineering Institute
Carnegie Mellon University
Agile and Systems Engineering?
Agile & Systems Engineering: Software Drivers for Systems Engineering
Agile Teams/Systems Engineering
Communication Challenges

A WISH LIST:
You just don't understand me
-SE needs to better understand Agile

© copyright Carnegie Mellon University 2013. All rights reserved.
Panel Discussion

Diverse set of panelists – Government, Industry, academia, and FFRDCs

Diverse experience and views of Agile within the government

Touch on challenges, successes, what works and what doesn’t
Introductions

Brian Gallagher – CACI

Eileen Wrubel – SEI

Carmen Graver – USMC

Rich Turner – Stevens Institute

Peter Christensen – Mitre
Experiences with Agile for Systems Engineering in the Defense Industry

Brian Gallagher
Senior Vice President, Operational Excellence
October 30, 2013
One Example

- **Wanted to increase delivered capability over time**

- **Productivity Numbers:**
  - Moved from Iterative Development to Agile

<table>
<thead>
<tr>
<th>Period</th>
<th>PRs/ CRs Delivered</th>
<th>Months</th>
<th>PRs/ CRs per Month</th>
</tr>
</thead>
<tbody>
<tr>
<td>09/01/2008 – 01/19/2011</td>
<td>532</td>
<td>28 months</td>
<td>532/28 = 19/Month</td>
</tr>
<tr>
<td>01/20/2011 – 12/09/2011</td>
<td>710</td>
<td>12 months</td>
<td>710/12 = 59/month</td>
</tr>
</tbody>
</table>

- **Additional Benefits:**
  - Reduced Wasted Development Effort
    - Did not build things the customer did not need
  - Reduced Rework
    - Got it right the first time more often
    - Lower bug rates
  - Customer Satisfaction Increased
  - Development Team Satisfaction Increased
The Trouble with Terminology

- “Agile Systems Engineering” or “Engineering Agile Systems”?
- Do we want “Agile” or “agility”?
- When we stick the term “Agile” in front of everything, it means nothing:
  - Agile Systems Engineering
  - Agile Earned Value
  - Agile Project Management
  - Agile Architecture

“I have only made this letter longer because I have not had the time to make it shorter.”

- Blaise Pascal, The Provincial Letters, 1657
Disciplined Agility

- **Discipline**
  - Slow to change or react
  - Repeatably late
  - Inability to change quickly
  - Heavy reliance on CMMI

- **Agility**
  - Quick to React and Replan
  - Architecture-centric
  - Predict Outcomes
  - Highly Collaborative

- **High (H)**
  - Process Impedes Mission and Performance Balance
  - Poor Results
  - Lack of Speed and Urgency
  - Activity over Progress
  - Lack of Architectural Focus
  - Lethargic Progress
  - Complexity Stifles Creativity
  - Ignorance of Best Practice
  - High Cost of Change
  - Staff Burn-out
  - Methodology over Performance

- **Low (L)**
  - Process and Performance Balance
  - High Results
  - Speed and Urgency
  - Progress
  - Architectural Focus
  - Dynamic Progress
  - Simplicity Promotes Creativity
  - Awareness of Best Practice
  - Low Cost of Change
  - Staff Engagement
  - Performance over Methodology
Which Approach is Better?

Traditional Approach

- System Context
- Architecture & Design
- Implementation

Evolutionary Approaches

- System Context
- Simultaneous Definition and Tradeoffs
- Architecture & Design

- requirements
- cost
- schedule
- business processes
- operational procedures, etc.

- COTS products
- NDI

Known Rqmnts

Buy, Re-use, Build, Integrate, Refresh
Choosing the Systems Engineering Approach

- “Dial in” the appropriate amount of agility based on program characteristics, customer, team capabilities, and risk

- Example:
  - Quick reaction capability drops in an evolving operational environment versus deploying a highly reliable infrastructure
What Does Agility Mean in this Context?

Is it enough to require developers to “Go Agile”?
Software and Systems Engineering for DoD using Agile

Eileen Wrubel
Senior Engineer
Software Engineering Institute
Carnegie Mellon University
Agile software/Systems engineering in the DoD

Lots of historical tension between these disciplines even when you don’t have Agile in the mix

Our research effort: how are Agile software teams interacting with systems engineering teams on DoD programs?

What works?
What crashes & burns?
What environments are set up for success?

Tech Note “Agile Software Teams: How They Engage with Systems Engineering on DoD Acquisition Programs” (working title) in internal review at SEI.
Interaction cases we envisioned
Thinking about systems engineering & Agile

COMMUNICATING & COORDINATING
AGILE ←→ SYS ENG

I need a report!
Nudge, nudge,
wink, wink,
Know what I mean?

TEAM

Don't get too hung up on words

TEAM

TEAM

TEAM

TRANSFORMING ARTIFACTS

SYSTEMS ENGINEERING AS A SERVICE
Do contract vehicles matter?

Contracting Officers will stay on the "safe" path unless forced off.
Training and awareness at the PMO matters. A LOT.
United States Marine Corps

Agile and Systems Engineering
“Marriage”

Carmen Graver

Version: 13 May 08

Unclassified
MC-Agile Supporting Documentation

SPRINT ‘0’
- SEP
- Agile Annex “SEMP”
- DOORS Database
- TEMP
- Agile Annex

SPRINT ‘1’
- Sprint 1 Design
- Sprint 1 Test Plan
- Program RTM

SPRINT ‘2’
- Sprint 2 Design
- Sprint 2 Test Plan

SPRINT ‘3’
- Sprint 3 Design
- Sprint 3 Test Plan

SPRINT ‘N’
- Update SEMP 1-3 pages
- “Retrospective” SEMP
- “Release” Design
- “Comprehensive” Test Report/Plan
- “Visibility” thru Forge.Mil

Continuous Feedback and Traceability

Unclassified

Version: XX YYY 08
• Training at all levels is **absolutely vital**
• It’s okay to **raise issues**
• It’s okay to **change your process**
• Involve **all** disciplines up front and early
• Develop a **common** program vision
• “Technical reviews” need to be **collaborative conversations**
• All documentation must be **value added**
• Ensure that **all** team members have access to collaborative tools and environments
• **Share** lessons across programs
Lean-Agile Systems Engineering in Defense Panel

Richard Turner
Stevens Institute rturner@stevens.edu

NDIA SE Conference
October 30, 2013
Alexandria, VA

www.SERCuarc.org
The World is Changed

- Traditional systems engineering assumptions
  - Requirements are predefined and generally stable
  - Resources and technologies are predictable and stable
  - Values remain stable
  - There is sufficient time to complete the work
  - Reductionism is the best way to approach large problems

- Some results of these assumptions
  - The V model and its (apparent) waterfall-like once-through approach
  - Addiction to plans and schedules rather than value and solutions
  - Focus on precision and coherence (if not accuracy) with requirements
  - Change is seen as the enemy (along with the customer who wants it)
  - Deep specializations in engineering
  - Local optimization in processes and designs
The World as Changed

• That was so 20th Century! In the 21st Century:
  – System contexts have multiplied, and change in customer needs and developer solution technologies has accelerated.
  – Requirements are less tangible, more evolving, and sometimes emergent
  – Systems are both complex and constantly adapting

• Given the actual terrain has changed, we need some new mapping tools and techniques

• Need proof of the change?
  – The venerable PMI has “adapted” (finally)
  – V5 of the Guide to the PMBOK provides for both predictive (plan-driven) and adaptive (agile) project lifecycles!
A (Very) Few Lean and Agile Pioneers

- Fighter pilots
- Designers
- Psychologists
- Manufacturers
- Researchers
- Software developers
- Product developers and entrepreneurs
- Systems engineers (?)
“Fundamental things apply”

- Stakeholder **Value-based** Evolution
- **Incremental** Commitment and Accountability
- Concurrent Multi-discipline Engineering
- Evidence- and Risk-based Decisions

Adapted from *The Incremental Commitment Spiral Model* by Boehm, Lane, Koolumanjwong, and Turner

- Value, Flow
- Agility, Response-ability
Sticking Points

• Large-scale budgeting and estimation
• Long-lead items
• Operational systems of independently evolving systems
• Highly regulated domains (e.g. defense, financial, health)
• Command and control environments (low trust, bureaucratic)
• There is still no Silver Bullet
• ICSM principles
• Service orientation is promising
• Trust is a key ingredient and often difficult to find
• “Maybe...” is better than “Hell, No!”
• Patience, but not abdication
• Creativity and Collaboration can be better than Command & Control
• Santayana was half right – it’s only the mistakes that you don’t want to repeat, not the successes
Questions?