The Road to Agile Systems Engineering

Presented by:
Rob Simons and Philip Matuzic

Contributors:
Dick Carlson and Don O’Connell
**The Road to Agile Systems Engineering**

Presented at the 23rd Systems and Software Technology Conference (SSTC), 16-19 May 2011, Salt Lake City, UT. Sponsored in part by the USAF. U.S. Government or Federal Rights License

---

**ABSTRACT**

""""Systems Engineering for large, high-technology, aerospace programs is scope historically complex and broad in scope. We present a logical and practical path toward leaner and more agile systems engineering beginning with a high-level overview of typical systems engineering processes. Emphasis is on activities and processes amenable to improvement by adapting Agile practices. This presentation is limited to a small subset of early SE activities; This presentation shows example candidate areas that can benefit from the application of Agile practices, and identifies efforts currently underway to apply these practices to new and ongoing development."""

---

**SUPPLEMENTARY NOTES**

Approved for public release; distribution unlimited
Abstract

Boeing Defense, Space & Security | Lean-Agile Software

- Systems Engineering for large, high-technology, aerospace programs is historically complex and broad in scope.

- We present a logical and practical path toward leaner and more agile systems engineering beginning with a high-level overview of typical systems engineering processes. Emphasis is on activities and processes amenable to improvement by adapting Agile practices.

- This presentation is limited to a small subset of early SE activities

- This presentation shows example candidate areas that can benefit from the application of Agile practices, and identifies efforts currently underway to apply these practices to new and ongoing development.

“Agility is the ability to both create and respond to change in order to profit in a turbulent business environment.” – Jim Highsmith
Drivers for Agile Systems Engineering

Leverage “lighter weight” tailored Systems Engineering process for urgent needs, rapid fielding and technology insertion....accelerate delivery of technical capabilities to win the current fight by using agile SE to incrementally / quickly evaluate changing requirements
The SE ‘Engine’ from an Ops Analysis Perspective

- SE is process-based and often influenced

Related Terms:
- Customer
- Primary Functions
- Systems Elements
- Organizations responsible for Primary Functions
- Development, Production/Construction, Verification, Deployment, Operations, Support, Training, Disposal
- Hardware, Software, Personnel, Facilities, Data, Material, Services, Techniques


The Takeaway: This is complex stuff!
Where to Start? An Aerospace Big Picture View

Systems Engineering views SE involvement across a product life cycle

Boeing Defense, Space & Security | Lean-Agile Software

Source: http://www.thefullwiki.org/V-Model
If you accept that:

- Agile has been successful with software because it is ‘soft’* and lends itself to the qualities of Agile and Scrum

Then you could accept that:

- Some Systems Engineering activities produce ‘soft’ artifacts that support HW activities (where Agile is more difficult to instantiate) and could be candidates for ‘agilizing’

Where can you find many examples of ‘soft’ Systems Engineering activities?

- In the early ‘Front End’ Conceptual Phase

---

*Generally describes elements and activities that produce artifacts with a high degree of flexibility and latitude in its definition, execution, and outputs, such as studies and analyses.
The percentages along the time line represent the actual life-cycle cost (LCC) accrued over time based on a statistical analysis performed on projects in the U.S. Department of Defense (DOD) as reported by the Defense Acquisition University.

As shown, the Concept Stage of a new system averages 8% of the total LCC.

The curve for committed costs represents the amount of LCC committed by project decisions.

INCOSE SE Handbook, pg. 15: “(This) figure also demonstrates the consequences of making early decisions without the benefit of good information and analysis. SE extends the effort performed in concept exploration and design to exceed the percentages shown in the cumulative committed cost curve and reduce the risk of hasty commitments...”

Source: INCOSE Systems Engineering Handbook v. 3.2, January 2010; INCOSE; Defense Acquisition University, 1993
What does SE do in the Front End?

Source: http://www.thefullwiki.org/V-Model
Conceptual System Design Phase Activity

Examples:

- Early Problem Definition & Need Identification
- Advanced System Planning
- Evaluation & Trade Studies
- System Feasibility & Requirements Analysis
- Measures of Effectiveness/Technical Performance Measures (MOE / TPM) Investigation
- Output Supports Modeling and Simulation activities
- Culminates in a Conceptual Design Review
Trade Studies at early program stages are often oriented towards determining technical feasibility and focusing on reducing the solution space.

Activities include articulating the purpose, identifying scope and constraints, etc.

Trade Studies are applicable to each level of the system architecture.

Application examples include technologies, suppliers, ‘boxes’, etc.
Applying Agile to the Trade Study Downselect Alternatives Process

- Identify evaluation criteria & weighting factors iteratively
  - Define purpose, scope, constraints, assumptions
  - Identify initial list of alternatives
  - Apply evaluation method for analyzing initial list of alternatives

- Downselect alternatives iteratively
  - Apply evaluation method for analyzing initial list of alternatives

- Perform detailed analysis iteratively
  - Apply evaluation method for analyzing initial list of alternatives

- Identify recommended alternatives
General Steps of a Trade Study:

- Define the **purpose** of the trade study
- Plan the trade study
- Describe the **assumptions** for the trade study
- Define the **evaluation criteria** and assign **weighting factors**
- Identify and develop **alternatives**
- **Evaluate** alternatives against the evaluation criteria
- **Recommend** a decision based on the evaluation of the alternatives
- **Document** recommendations and rationale in a report
- Review and **accept/reject** the trades study recommendations and report

Suggested AgileSE Trade Study

- Implements most of the Agile practices and principles
- Applies the Scrum framework where roles, activities, and artifacts are essentially the same
- At the end of each iteration, the product backlog “baseline” has evolved into a clearer and more complete set of requirements
- Acceptance tests are written for each backlog item (dependent upon the type of TS)
- Attributes, constraints, and dependencies are identified
- All data is managed and controlled

Source: INCOSE Systems Engineering Handbook v. 3.2, January 2010; INCOSE; Defense Acquisition University, 1993
Agile Systems Engineering will Challenge the Paradigm

Boeing Defense, Space & Security| Lean-Agile Software

Recommendations:

- Find a Champion
- Look for SE opportunity areas early in the Lifecycle
- Look for discrete job chunks that can be easily defined & are amenable to implementing Scrum practices
- Consider selecting a bounded pilot project executing a specific activity (such as a TS)

Suggest developing Agile Checklists that reflects your AgileSE concerns:
- Level of Uncertainty & Agility Needed
- Cost, Risk (Assessment of Risk is key)
- Metrics to gauge progress, improvement, and satisfaction

- Recognize that you will want to deliver the product needed at the end and not necessarily what was requested at the beginning
Challenges Going Forward

Boeing Defense, Space & Security | Lean-Agile Software

- Cultural acceptance of Agile within Systems Engineering
- Acceptance of AgileSE from Acquisition & Contractual organizations
  - Oversight, Documentation, Detailed Planning issues
- Expanding Agile to successfully address mid-late product life cycle issues
  - Applicability of Agile methods and practices to Prototyping, Manufacturing, Production, etc.
- Educating SE Leadership and Teams on implementing, practicing, and sharing successful AgileSE implementations
Experience in 2007-2009: Agile Software Requirements

- System development that required modifying a large enterprise COTS application across several engineering domains

- A more efficient approach was needed

- A release plan was established aligned to the scheduled mandate

- The Agile Scrum approach was selected to manage the project

- Teams were staffed with system engineers to develop software requirements through a series of 4-week iterations
The model below illustrates a much tighter collaboration between Agile system engineering teams and Agile software teams.

Source: Dick Carlson
The direction now is to define specific Agile practices that can be used to enable more efficient activities for the larger system engineering effort.

- Develop several Agile templates of typical SE artifacts:
  - SEMP
  - Trade Study
  - IV&V
  - Etc.

- Organize and prioritize SE activities and develop SE artifacts by implementing Agile practices:
  - Active customer participation
  - Short daily stand-up meetings
  - Planning and estimating
  - Frequent deliveries
  - Short iterations
  - Prioritized requirements
  - Artifact reviews
  - Self-organized teams
  - Frequent delivery
  - Simplicity
  - Sustainable pace
Questions?
Backup
Agile is a philosophy defined by principles and practices that drive quick delivery of quality software and encourage user feedback.

**Principles:**
- Customer Satisfaction
- Frequent Delivery/Deployment
- Motivated Team
- Working Software
- Technical Excellence
- Emergent Design
- Embrace Change
- Collaboration
- High Bandwidth
- Sustainable Pace
- Simplicity
- Continuous Improvement

**Practices:**
- Close customer collaboration
- Daily stand-up meetings
- Continuous integration
- Automated testing
- Planning and estimating
- Short iterations
- Test-driven development
- Prioritized requirements
- Product demonstrations
- Self-organized teams

Source: Agile Alliance (www.agilealliance.org)
Agile Practices Often Used

- Continuous planning
- Short iterations (2 to 4 weeks)
- On-site customer or proxy
- Self-organized teams
- Pair programming
- Small releases
- Prioritized requirements
- Automated testing
- Refactoring
- Continuous integration and CM
- Test first
- Simple, robust design

Source: Agile Alliance (www.agilealliance.org)
Some Lessons Learned

Boeing Defense, Space & Security | Lean-Agile Software

- Focused, face-to-face iteration sessions improve team synergy

- Appoint someone to maintain specified time boxed activities

- First estimating round should focus on perceived size, complexity, or difficulty (in nebulous units)

- Scrum Master should work with management to resolve unplanned organizational impediments that affect work tasks

- Avoid side discussions that disrupt team momentum

- Attend all daily stand-ups (best way to know health and progress of team)

- Working as a team helps everyone understand tasks and how to formulate stories

- Team consultations with domain SMEs help clarify misunderstandings

- Ensure requisite architecture, infrastructures, technologies, and tools are in place before starting team activities
Common Agile Terms

- Agile
- Burndown Chart
- Daily Scrum
- Iteration or Sprint Backlog
- Iteration Planning
- Product Backlog
- Product Owner
- Scrum
- ScrumMaster
- Story
- Sprint
- Task Board
- Team