Teaching the Elephant to Dance: Agility Meets Systems of Systems Engineering and Acquisition

Keynote, GSAW 2005
March 3, 2005
Outline

• Nature of systems of systems engineering and acquisition
  – Particularly, network-centric systems of systems (NCSOS)
  – Acquisition is more like doing C4ISR than buying fruitcake

• Agile methods and NCSOS: strengths and difficulties
  – Helpful, but not a silver bullet

• Integrating agile and plan-driven methods
  – Workshop results and integration framework

• Critical success factors
  – Evolutionary, risk-driven spiral framework and plan-driven builds
  – Compatible acquisition and contracting methods and skills
  – Knowing when not to system engineer

• Conclusions, references
The Need for NetCentric Systems of Systems (NC SOS)

- Lack of integration among stovepiped systems causes
  - Unacceptable delays in service
  - Uncoordinated and conflicting plans
  - Ineffective or dangerous decisions

- NC SOS can strongly boost performance of
  - National Defense
  - Supply Chain Management
  - National Air Traffic Control
  - Crisis Management
System Acquisition Trends

Traditional Acquisition

• Standalone systems
• Stable requirements
• Rqts. determine capabilities
• Control over evolution
• Enough time to keep stable
• Failures locally critical
• Reductionist systems
• Repeatability-oriented process, maturity models

Current/Future Trends

• Everything connected (maybe)
• Rapid requirements change
• COTS capabilities determine rqts.
• No control over COTS evolution
• Ever-decreasing cycle times
• Failures globally critical
• Complex, adaptive, emergent systems of systems
• Adaptive process models
NCSOS Acquisition is More Like Doing C4ISR - than buying fruitcake

• No detailed plan survives the first engagement
• Acquisition C4ISR via spiral OODA loops
  – Observe, Orient, Decide, Act
  – Vs. Requirements, Delay, Surprise
• Concurrent tasking, collaboration technology essential
  – Spanning deep chains of command
    • Customer, LSI, IPT’s (C4ISR), Decision Support, COP Refresh, Sensor Fusion, Sensors, Sensor components
• Common strategy essential; microplanning risky
• Competition, technology, marketplace ISR essential
• Rapid adaptability essential
Acquisition C4ISR Via Spiral OODA Loops
- Example: ARPANet/Internet Spiral

Observe new/updated objectives, constraints, alternatives
- Usage monitoring
- Competition, technology, marketplace ISR

Orient with respect to stakeholders priorities, feasibility, risks
- Risk/Opportunity analysis
- Business case/mission analysis
- Prototypes, models, simulations

Operate as current system
Accept new system

Act on plans, specifications
- Keep development stabilized
- Change impact analysis, preparation for next cycle (mini-OODA loop)

Decide on next-cycle capabilities, architecture upgrades, plans
- Stable specifications, COTS upgrades
- Development, integration, V&V, risk management plans
- Feasibility rationale

Life Cycle Architecture Milestone for Cycle
NC S O S Acquisition Practice Implications

• Need to stimulate agility during Observe, Orient, Decide sectors
  – With flexibility-oriented contract, award fee provisions

• Need to stimulate stability during Act sector
  – Current stability-oriented contract provisions a good match

• Risk-driven spiral process generator accommodates both

• Waterfall and V-models have their risk-driven place
  – Acquiring preceded systems in stable marketplace
  – Executing stable Act sector
Outline

• Nature of systems of systems engineering and acquisition
  – Particularly, network-centric systems of systems (NCSOS)
  – Acquisition is more like doing C4ISR than buying fruitcake

→ Agile methods and NCSOS: strengths and difficulties
  – Helpful, but not a silver bullet

• Integrating agile and plan-driven methods
  – Workshop results and integration framework

• Critical success factors
  – Evolutionary, risk-driven spiral framework and plan-driven builds
  – Compatible acquisition and contracting methods and skills
  – Knowing when not to system engineer

• Conclusions, references
The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
NCSSS-Relevant Agile Practices

• Short stabilized increments (+)
  – Prioritized feature backlog
• Continuous customer-developer participation (+)
• Early test; continuous integration (+)
• Tacit interpersonal vs. explicit documented knowledge (+)
• Welcome changing requirements (+)
• Simple design (-)
  – Just for current increment
  – Refactor to accommodate later capabilities
Agile and Plan-Driven Home Grounds: Five Critical Decision Factors

- Size, Criticality, Dynamism, Personnel, Culture

![Diagram showing the relationship between Size, Criticality, Dynamism, Personnel, and Culture](image)
NC SOS Agile/Plan-Driven Profile

Size
(# of personnel)

Personnel
(% Level 1B) (% Level 2&3)

Dynamism
(% Requirements -change/month)

Culture
(% thriving on chaos vs. order)

Criticality
(Loss due to impact of defects)
Conclusions So Far

• Large global enterprises need NCSOS
• NCSOS acquisition is more like doing C4ISR
  – Acquisition C4ISR via spiral OODA loops
  – Need more adaptive vs. build-to-spec acquisition practices
• Key agile practices help, but scalability is difficult
• NCSOS acquisition needs to balance agility and discipline
  – Integrating agile and plan-driven methods
Outline

• Nature of systems of systems engineering and acquisition
  – Particularly, network-centric systems of systems (NCSOS)
  – Acquisition is more like doing C4ISR than buying fruitcake

• Agile methods and NCSOS: strengths and difficulties
  – Helpful, but not a silver bullet

  Integrating agile and plan-driven methods
  – Workshop results and integration framework

• Critical success factors
  – Evolutionary, risk-driven spiral framework and plan-driven builds
  – Compatible acquisition and contracting methods and skills
  – Knowing when not to system engineer

• Conclusions, references
• Large companies having success with small agile pilot projects
  - ABB, Daimler Chrysler, IBM, LMCO, Motorola, Northrop Grumman, Raytheon, SAIC
  - Generally higher productivity, customer satisfaction, morale

• Some perceived agile problems were non-issues
  - Agile is monolithic, disorganized
  - No framework for quantitative management, quality assurance

• Some perceived agile problems were real issues
Large-Company Agile Assimilation Issues

• Scalability of agile methods
  – Tacit knowledge (propagation; personnel turnover; 5,000 requirements)
  – Multi-team coordination

• Avoiding agile stovepipes
  – Limitations on freedom of choice
    • COTS, interfaces, GUIs, legacy systems

• Traditional business practices
  – Contracting; earned value systems; timekeeping; waterfall/V-model standard, HR practices

• Inflexible maturity model interpretations

• Customer collocation, access

• Architecture suboptimization on early increments
  – Example: key performance parameters

• Predictable vs. unpredictable change
Spiral Integration of Agile and Plan-Driven Processes

Driven By:

- **Success-critical stakeholders’ win conditions**

1. **Identify System Objectives, Constraints & Priorities (OC&Ps); Alternative Solution Elements**
   - 1a. Identify Success-Critical Stakeholders
   - 1b. Stakeholders’ Review

2. **Evaluate Alternatives with respect to OC&Ps**
   - 2a. Evaluate Alternatives with respect to OC&Ps
   - 2b. Assess, Address Risks

3. **Elaborate Product and Process Definitions**

4. **Verify and Validate Product and Process Definitions**

Progress Through Steps:

1. **Stakeholders**
   - Build
   - Build
   - Build

3. **Risk Management**
   - 3. Elaborate Product and Process Definition

4. **Spiral anchor point milestones**
   - 4. Feasibility Rationale

Stakeholders’ Commitment

Stakeholders’ Review
Spiral Anchor Points Enable Concurrent Engineering

Disciplines
- Business Modeling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment
- Configuration & Change Mgmt
- Project Management
- Environment

Phases
- Inception
- Elaboration
- Construction
- Transition

Iterations
- Initial
- Elab #1
- Elab #2
- Const #1
- Const #2
- Const #N
- Tran #1
- Tran #2

3/3/2005 © USC - CSE
### Need Concurrently Engineered Milestone Reviews

**Life Cycle Objectives (LCO); Life Cycle Architecture Package (LCA)**

| Operational Concept | • Elaboration of system objectives and scope by increment  
|                     | • Elaboration of operational concept by increment  
| System Prototype(s) | • Exercise range of usage scenarios  
|                     | • Resolve major outstanding risks  
| System Requirements | • Elaboration of functions, interfaces, quality attributes, and prototypes by increment  
|                     |   - Identification of TBD’s (to be determined items)  
|                     | • Stakeholders’ concurrence on their priority concerns  
| System and Software Architecture | • Choice of architecture and elaboration by increment  
|                     |   - Physical and logical components, connectors, configurations, constraints  
|                     |   - COTS, reuse choices  
|                     |   - Domain architecture and architectural style choices  
|                     | • Architecture evolution parameters  

3/3/2005 © USC - CSE
### Need Concurrently Engineered Milestone Reviews

**Life Cycle Objectives (LCO); Life Cycle Architecture Package (LCA)**

| Life-Cycle Plan | • Elaboration of *WWW*HH* for Initial Operational Capability (IOC)  
                     • Partial elaboration, identification of key TBD’s for later increments |
|-----------------|--------------------------------------------------------------------------|
| Feasibility Rationale | • Assurance of consistency among elements above  
                           • All major risks resolved or covered by risk management plan. |

LCO (MS A) and LCA (MS B) Pass/Fail Criteria

- **A system built to the given architecture will**
  - Support the operational concept
  - Satisfy the requirements
  - Be faithful to the prototype(s)
  - Be buildable within the budgets and schedules in the plan
  - Show a viable business case
  - Establish key stakeholders' commitment to proceed

LCO: True for at least one architecture
LCA: True for the specific life cycle architecture;
   - All major risks resolved or covered by a risk management plan
The Cost of Hasty Fixed Requirements: 15-Month Architecture Rework Delay

- Original Architecture: Modified Client-Server
- Required Architecture: Custom; many cache processors

Original Cost vs. Response Time (sec):
- Original Spec
- After Prototyping

Costs:
- $100M
- $50M

15-Month Architecture Rework Delay
KPP Validation with Spiral Model

- **Attempt to validate 1-second KPP**
  - Architecture analysis: needs expensive custom solution
  - Prototype: 4-seconds OK 90% of the time

- **Negotiate KPP ranges**
  - 2 seconds desirable
  - 4 seconds acceptable with some 2-second special cases

- **Benchmark client-server to validate feasibility**

- **Present solution and feasibility rationale at anchor point milestone review**
  - Result: Acceptable solution with minimal delay
Key Points

- It’s not a requirement if you can’t afford it
  - Or fit it into your schedule
- Spiral approach avoids late rework
- Beware of sub-optimization on small-scale early iterations
Using Risk to Balance Discipline and Agility - Overview

**Step 1. Risk Analysis**
- Rate the project's environmental, agility-oriented and plan-driven risks.

**Step 2. Risk Comparison**
- Compare the agile and Plan-driven risks
- Agility risks dominate
- Plan-driven risks dominate
- Neither dominate

**Step 3. Architecture Analysis**
- Architect application to encapsulate agile parts

**Step 4. Tailor Life Cycle**
- Go Risk-based Agile
- Go Risk-based Plan-driven
- Go Risk-based Agile in agile parts; Go Risk-based Plan-driven elsewhere

**Step 5. Execute and Monitor**
- Deliver incremental capabilities according to strategy
- Monitor progress and risks/opportunities, readjust balance and process as appropriate
- Tailor life cycle process around risk patterns and anchor point commitment milestones

Note: Feedback loops present, but omitted for simplicity
## NCSOS Agile/Plan-Driven Strategy

- **CRACK**: collaborative, representative, authorized, committed, knowledgeable

<table>
<thead>
<tr>
<th></th>
<th>Startup</th>
<th>Teambuilding</th>
<th>Systems Architecting</th>
<th>Development</th>
</tr>
</thead>
</table>
| **Stakeholders** | Furnish CRACK representatives and alternates | • Develop shared vision  
• Negotiate top-level system objectives, architecture, plans, feasibility rationales. | • Prepare for/select developers  
• Formulate/negotiate definitive requirements, architecture, plans, feasibility rationales.  
• Encapsulate agile portions | • Ensure representative exercise of incremental capabilities  
• Monitor, adapt to new developments |
| **Project Leadership, Risk Management Teams** | • Staff and organize to cover major risk areas | • Develop shared vision  
• Negotiate top-level system objectives, architecture, plans, feasibility rationales. | | • Monitor and manage project progress, risk resolution, and new technology developments  
• Continuously integrate/test growing software infrastructure and components |
| **Agile, Plan Driven Developers** | | | • Develop compatible architectures, plans, feasibility rationales | • Develop system components |
Outline

• Nature of systems of systems engineering and acquisition
  – Particularly, network-centric systems of systems (NCSOS)
  – Acquisition is more like doing C4ISR than buying fruitcake

• Agile methods and NCSOS: strengths and difficulties
  – Helpful, but not a silver bullet

• Integrating agile and plan-driven methods
  – Workshop results and integration framework

  ❮ Critical success factors
  – Evolutionary, risk-driven spiral framework and plan-driven builds
  – Compatible acquisition and contracting methods and skills
  – Knowing when not to system engineer

• Conclusions, references
NCSOS Acquisition: Critical Success Factors

• **Risk-driven spiral processes and organizations**
  – Project manager’s risk/opportunity team

• **Stabilized evolutionary builds**
  – Concurrent plan-driven construction, agile rebaselining
  – Anchor point milestones and Feasibility Rationales

• **Rethinking supplier management**
  – Teambuilding and plans/architecture participation
  – Balanced agile/plan-driven contracts, award fees

• **Knowing when not to system engineer**
Agile Rebaselining Mini OODA Loop

• Many sources of next-build volatility
  – Supplier chain slippages, changes in current build
  – External interface volatility
    • COTS; interoperating systems
  – New threats, technology, policies
  – Organizational, top-management volatility

• Next build needs to hit the ground running

• Requires critical-mass budget, talent, tools for
  – Change impact analysis (observe, orient)
  – Solution rebaselining (decide, act)
    • Renegotiating future builds’ content, associated plans and resources
    • Integrated COTS refresh preparation
DoDI 5000.2 “Spiral Development”
Section 3.3.2.1

- Desired capability is identified
  - End-state requirements not initially known

- Requirements refined through demonstration and risk management
  - Continuous user feedback
  - Each increment provides user the best possible capability

- Requirements for future increments depend on feedback from users and technology maturation

NB: This section of 5000 is under revision and all reference to spiral may be removed
Knowing When Not to System Engineer
- A multi-platform NC SOS example

- Customer system-engineers an optimized product line architecture for platform functions
  - Estimates cost savings from reuse

- Customer solicits best-of-breed platform suppliers
  - Contracts with most cost-effective bidders

- Customer discovers that supplier bids are based on product line - incompatible components
  - Too expensive to refit to product line architecture

- Better to risk-manage degree of product line achievability
  - Involve potential suppliers in product line option exploration
Conclusions

• Large global enterprises need NCSOS
• NCSOS acquisition is more like doing C4ISR
• Critical success factors include
  - Risk-driven spiral processes and organizations
  - Concurrent plan-driven builds, agile rebaselining
    • And associated budgets, talent, tools
  - Rethinking supplier management
    • Balanced agile/plan-driven contracts, award fees
    • Teambuilding and plans/architecture participation
    • Knowing when not to system engineer
References


MBASE web site: sunset.usc.edu/research/MBASE

Agile workshops web site: www.cse.usc.edu/events/2004/arr

CrossTalk articles: www.stsc.hill.af.mil/crosstalk