The Incremental Commitment Model for System of Systems Development

----Tutorial----

Barry Boehm and Jo Ann Lane
University of Southern California
Center for Systems and Software Engineering

http://csse.usc.edu

Presented at
Systems and Software Technology Conference
18 June 2007
Abstract

• The ultimate **goal of system of systems (SoS) development** is to **deliver capabilities** that satisfy the needs of its operational stakeholders within satisfactory levels of the resources of its development stakeholders. Developing and delivering SoSs that simultaneously satisfy the success-critical stakeholders usually **requires managing a complex set of risks**. Many of these risks pose an additional risk to the delivery of an acceptable operational SoS **within the available budget and schedule**.

• The **goal of this tutorial** is to **describe a development process model** that **better enables integration of the human, hardware, and software aspects of a SoS’s development and evolution**. This model, called the Incremental Commitment Model (ICM), embodies the principles underlying the spiral model of system development, but organizes its process into multiple views (including a spiral view) that are more straightforward to apply and harder to misinterpret than the spiral model has been.
Objectives

• Describe the ICM and associated anchor point milestones for SoS development
• Explain the 5 underlying principles of the ICM
• Provide multiple views of the ICM based on example SoS development risks
• Provide participants with experience in developing an ICM for a sample SoS development program
What is the ICM?

• Process framework for developing systems
• Integrates the strengths of phased and risk-driven spiral process models
•Synthesizes together principles critical to successful system development
  – Stakeholder satisficing,
  – Incremental growth of system definition and stakeholder commitment
  – Iterative system definition and development
  – Concurrent system definition and development
  – Risk management through risk-driven activity levels and anchor point milestones
What is a “System of Systems”? 

• Very large systems developed by creating a framework or architecture to integrate component systems 

• SoS component systems independently developed and managed 
  – New or existing systems in various stages of development/evolution 
  – Have their own purpose 
  – Can dynamically come and go from SoS 

• SoS exhibits emergent behavior not otherwise achievable by component systems 

• Typical domains 
  – Business: Enterprise-wide and cross-enterprise integration to support core business enterprise operations across functional and geographical areas 
  – Military: Dynamic communications infrastructure to support operations in a constantly changing, sometimes adversarial, environment 

*Based on Mark Maier’s SoS definition [Maier, 1998]*
Tutorial Outline

• Incremental Commitment Model (ICM) background
• Overview of Incremental Commitment Process
• Overview of System of Systems (SoS) arena: definitions and how SoS Engineering (SoSE) differs from Traditional Systems Engineering (TSE)
• Applying ICM to the SoS arena
• SoS case study and examples
From the Spiral Model to the ICM

- Need for intermediate milestones
  - Anchor Point Milestones (1996)
- Avoid stakeholder success model clashes
  - WinWin Spiral Model (1998)
- Avoid model misinterpretations
  - Essentials and variants (2000-2005)
- Clarify usage in DoDI 5000.2
  - Initial phased version (2005)
- Explain SoS spiral usage to GAO
  - Underlying spiral principles (2006)
- Provide framework for human-systems integration
Future Process Challenges-I

- Multi-owner, multi-mission systems of systems
  - Integrated supply chain: strategic planning, marketing, merchandising, outsourcing, just-in-time manufacturing, logistics, finance, customer relations management
  - Over 50 separately evolving external systems
  - Need to satisfice among multiple stakeholders

- Rapid pace of change
  - In competition, mission priorities, technology, Commercial Off-the-Shelf (COTS), environment
  - Need incremental development to avoid obsolescence
  - Need concurrent vs. sequential processes
  - Need both prescience and rapid adaptability
    - Software important; humans more important
Priorities for Rapid Adaptation to Change

- Software is more adaptable than hardware
- People are generally more adaptable than software
- Need systems and software that better help people to adapt
  - Example: gentle-slope user interaction
The Need for Gentle Slope Systems: End-User Production Function

Value of results vs. Effort invested

- Gentle slope
- Lack of user programmability
- 3 weeks to “hello world”
Rapid Change: Ripple Effects of Changes
Breadth, Depth, and Length

Legend:
DOTMLPF  Doctrine, Organization,
          Training, Materiel, Leadership,
          Personnel, Facilities

Enterprise Control
Situation Assessment
Info Fusion
Sensor Data Management
Sensor Data Integration
Sensors
Sensor Components

Depth
Average Change Processing Time: 2 Systems of Systems

Average workdays to process changes

- Within Groups
- Across Groups
- Contract Mods
Future Process Challenges-II

- Emergence and human-intensiveness
  - Requirements not prespecifiable
  - Need for evolutionary growth
  - Need to manage uncertainty and risk

- Always-on, never-fail systems
  - Need well-controlled, high-assurance processes
  - Need to synchronize and stabilize concurrency
Process Model Principles

1. Success-critical stakeholder satisficing
2. Incremental growth of system definition and stakeholder commitment
3,4. Concurrent, iterative system definition and development cycles
   Cycles can be viewed as sequential concurrently-performed phases or spiral growth of system definition
5. Risk-based activity levels and anchor point commitment milestones
## Process Model Comparison with Respect to Process Principles

<table>
<thead>
<tr>
<th>Process Models</th>
<th>Principles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stakeholder Satisficing</td>
</tr>
<tr>
<td>Sequential Waterfall, V</td>
<td>Assumed via initial requirements; no specifics</td>
</tr>
<tr>
<td>Iterative, Risk-Driven Waterfall, V</td>
<td>Assumed via initial requirements; no specifics</td>
</tr>
<tr>
<td>Risk-Driven Evolutionary Development</td>
<td>Revisited for each iteration</td>
</tr>
<tr>
<td>Concurrent Engineering</td>
<td>Implicit; no specifics</td>
</tr>
<tr>
<td>Agile</td>
<td>Fix shortfalls in next phase</td>
</tr>
<tr>
<td>Spiral Process 2001</td>
<td>Driven by stakeholder commitment milestones</td>
</tr>
<tr>
<td>Incremental Commitment</td>
<td>Stakeholder-driven; stronger human factors support</td>
</tr>
</tbody>
</table>
Tutorial Outline

• ICM background

• **Overview of Incremental Commitment Process**
  • Overview of SoS arena: definitions and how SoSE differs from TSE
  • Applying ICM to the SoS arena
  • SoS case study and examples
Key Features of Incremental Commitment Process

- Builds on strengths of current processes
- Addresses challenges
- Incorporates principles
- Scalable and flexible to address different risk profiles
Scalable remotely controlled operations
Incremental Commitment in Gambling

- **Total Commitment: Roulette**
  - Put your chips on a number
    - Build 4:1 operator: UAS control system
  - Wait and see if you win or lose

- **Incremental Commitment: Poker, Blackjack**
  - Put some chips in
  - See your cards, some of others’ cards
  - Decide whether, how much to commit to proceed
Incremental Commitment In Systems and Life: Anchor Point Milestones

- Common System/Software stakeholder commitment points
  - Defined in concert with Government, industry organizations
  - Initially coordinated with Rational’s Unified Software Development Process
- Exploration Commitment Review (ECR)
  - Stakeholders’ commitment to support initial system scoping
  - Like dating
- Validation Commitment Review (VCR)
  - Stakeholders’ commitment to support system concept definition and investment analysis
  - Like going steady
Incremental Commitment In Systems and Life: Anchor Point Milestones (continued)

- **Architecting Commitment Review (ACR)**
  - Stakeholders’ commitment to support system architecting
  - Like getting engaged

- **Development Commitment Review (DCR)**
  - Stakeholders’ commitment to support system development
  - Like getting married

- **Incremental Operational Capabilities (OCs)**
  - Stakeholders’ commitment to support operations
  - Like having children
The Incremental Commitment Life Cycle Process: Overview

<table>
<thead>
<tr>
<th>Activities</th>
<th>Exploration Commitment Review</th>
<th>Valuation Commitment Review</th>
<th>Architecture Commitment Review</th>
<th>Development Commitment Review</th>
<th>Operations Commitment Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concurrent risk-and-opportunity-driven growth of system understanding and definition</td>
<td>Initial scoping</td>
<td>Concept definition investment analysis</td>
<td>System architecting</td>
<td>Increment 1 development</td>
<td>Increment 1 operations</td>
</tr>
<tr>
<td>Evaluation of evidence of feasibility to proceed</td>
<td>Feasibility rationales</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Stakeholder review and commitment</td>
<td>High, but addressable</td>
<td>Acceptable</td>
<td>Too high, unaddressable</td>
<td>Risk?</td>
<td>Risk?</td>
</tr>
</tbody>
</table>

Adjust scope, priorities, or discontinue.
Different Risk Patterns Yield Different Processes

ICM Lifecycle Phases

<table>
<thead>
<tr>
<th>General/DoD Milestones</th>
<th>Exploration</th>
<th>Valuation</th>
<th>Architecture</th>
<th>Development</th>
<th>Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM</td>
<td>ECR</td>
<td>VCR/CD</td>
<td>ACR/A</td>
<td>DCR/B</td>
<td>OCR/C</td>
</tr>
<tr>
<td></td>
<td>Exploration</td>
<td>Valuation</td>
<td>Architecture</td>
<td>Development</td>
<td>Operations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Activities

Example A: Simple Enterprise Resource Planning (ERP) based application
- High, but addressable
- Acceptable
- Too high, unaddressable
- Risk?
  - Risk?
  - Negligible
  - Negligible

Example B: Complex, but feasible product development
- Acceptable
- Acceptable
- Acceptable
- Acceptable
- Acceptable
- Risk?
  - Risk?
  - Risk?
  - Risk?
  - Risk?
  - Acceptable

Example C: Stakeholders agree that more convergence of objectives is necessary
- Acceptable
- Acceptable
- Acceptable
- Acceptable
- Acceptable
- Acceptable
- Risk?
  - Risk?
  - Risk?
  - Risk?
  - Risk?
  - Acceptable

Example D: A superior product enters the market
- Acceptable
- Acceptable
- Too high, unaddressable
- Risk?
  - Risk?
  - Discontinue
Pass/Fail Feasibility Rationales

• Evidence provided by developer and validated by independent experts that:
  If the system is built to the specified architecture, it will
  – Satisfy the requirements: capability, interfaces, level of service, and evolution
  – Support the operational concept
  – Be buildable within the budgets and schedules in the plan
  – Generate a viable return on investment
  – Generate satisfactory outcomes for all of the success-critical stakeholders

• All major risks resolved or covered by risk management plans

• Serves as basis for stakeholders’ commitment to proceed
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
# ICM HSI Levels of Activity for Complex Systems

<table>
<thead>
<tr>
<th>Activity category</th>
<th>Levels of activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td></td>
</tr>
<tr>
<td>Envisioning opportunities</td>
<td></td>
</tr>
<tr>
<td>System scoping</td>
<td></td>
</tr>
<tr>
<td>Understanding needs</td>
<td></td>
</tr>
<tr>
<td>Goals/objectives</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
<tr>
<td>Architecting and designing solutions</td>
<td></td>
</tr>
<tr>
<td>a. system</td>
<td></td>
</tr>
<tr>
<td>b. human</td>
<td></td>
</tr>
<tr>
<td>c. hardware</td>
<td></td>
</tr>
<tr>
<td>d. software</td>
<td></td>
</tr>
<tr>
<td>Life-cycle planning</td>
<td></td>
</tr>
<tr>
<td>Evaluation</td>
<td></td>
</tr>
<tr>
<td>Negotiating commitments</td>
<td></td>
</tr>
<tr>
<td>Development and evolution</td>
<td>OC1, OC2, OC3</td>
</tr>
<tr>
<td>Monitoring and control</td>
<td></td>
</tr>
<tr>
<td>Operations and retirement</td>
<td>Legacy, OC1, OC2</td>
</tr>
<tr>
<td>Organizational capability improvement</td>
<td></td>
</tr>
</tbody>
</table>
Primary Focus of HSI Activity Categories - I
– HSI activities often span multiple categories

<table>
<thead>
<tr>
<th>Activity Category</th>
<th>Primary Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Scoping</td>
<td>Identifying current system shortfalls and ways of addressing them via materiel or non-materiel solutions. Understanding needs and envisioning opportunities as below. Establishing initial system boundary that determines system scope.</td>
</tr>
<tr>
<td>Understanding Needs</td>
<td>Operations analysis, participatory analysis, ethnographic analysis, success-critical stakeholder identification, MEAD, social network analysis, competition analysis, market research, future needs analysis</td>
</tr>
<tr>
<td>Envisioning Opportunities</td>
<td>Scenarios, demonstrations, stories, shared visions, prototypes, models; change monitoring (technology, competition, marketplace, environment)</td>
</tr>
<tr>
<td>Architecting Solutions - Integration</td>
<td>Architecture frameworks, COTS/reuse evaluation, legacy transformation analysis, human/hardware/software allocation and quality attribute analysis and synthesis</td>
</tr>
<tr>
<td>- Human Elements</td>
<td>MEAD, participatory design, workflow/collaboration analysis and synthesis, task/usability analysis and synthesis, skill/career development planning</td>
</tr>
</tbody>
</table>
# ICM Anchor Point Milestone Content (1)
(Risk-driven level of detail for each element)

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Life Cycle Objectives (LCO/ACR)</th>
<th>Life Cycle Architecture (LCA/DCR)</th>
</tr>
</thead>
</table>
| **Definition of Operational Concept** | • Top-level system objectives and scope  
  – System boundary  
  – Environment parameters and assumptions  
  – Evolution parameters  
• Operational concept  
  – Operations and maintenance scenarios and parameters  
  – Organizational life-cycle responsibilities (stakeholders)  | • Elaboration of system objectives and scope of increment  
• Elaboration of operational concept by increment  |
| **System Prototype(s)**         | • Exercise key usage scenarios  
  • Resolve critical risks  | • Exercise range of usage scenarios  
• Resolve major outstanding risks  |
| **Definition of System Requirements** | • Top-level functions, interfaces, quality attribute levels, including  
  – Growth vectors and priorities  
  – Prototypes  
• Stakeholders’ concurrence on essentials  | • Elaboration of functions, interfaces, quality attributes, and prototypes by increment  
• Identification of TBD’s (to-be-determined items)  
• Stakeholders’ concurrence on their priority concerns  |
ICM Anchor Point Milestone Content (2)
(Risk-driven level of detail for each element)

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Life Cycle Objectives (LCO/ACR)</th>
<th>Life Cycle Architecture (LCA/DCR)</th>
</tr>
</thead>
</table>
| Definition of System and Software Architecture | • Top-level definition of at least one feasible architecture  
  – Physical and logical elements and relationships  
  – Choices of COTS and reusable software elements  
• Identification of infeasible architecture options | • 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 |
| Definition of Life-Cycle Plan            | • Identification of life-cycle stakeholders  
  – Users, customer, developers, maintainers, interoperators, general public, others  
• Identification of life-cycle process model  
  – Top-level stages, increments  
• Top-level WWWWWHH* by stage | • Elaboration of WWWWWHH* for Initial Operational Capability (IOC)  
  – Partial elaboration, identification of key TBD’s for later increments |
| Feasibility Rationale                    | • Assurance of consistency among elements above  
  – Via analysis, measurement, prototyping, simulation, etc.  
  – Business case analysis for requirements, feasible architectures | • Assurance of consistency among elements above  
  • All major risks resolved or covered by risk management plan |
# Incremental Commitment Model: Detailed

<table>
<thead>
<tr>
<th>ICM Lifecycle Phases</th>
<th>General/DoD Milestones</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploration</td>
<td>ECR</td>
<td>Concurrent risk-and-opportunity-driven growth of system understanding and definition</td>
</tr>
<tr>
<td>Valuation</td>
<td>VCR/CD</td>
<td>Definition &amp; analysis of scope &amp; solution alternatives</td>
</tr>
<tr>
<td>Architecting</td>
<td>ACR/A</td>
<td>Detailed mission scenarios, business work flows, macro-ergonomics aspects</td>
</tr>
<tr>
<td>Development</td>
<td>DCR/B</td>
<td>Use LCA, package for stabilized development, V&amp;V of increment 1</td>
</tr>
<tr>
<td></td>
<td>OCR/C, DCR/B</td>
<td>Concurrent agile change processing, rebasing of LCA\textsubscript{1}...LCA\textsubscript{n} packages</td>
</tr>
<tr>
<td></td>
<td>OCR/C, DCR/B</td>
<td>Operations and usage monitoring of increment 1</td>
</tr>
</tbody>
</table>

### Exploration
- Exploration and initial scoping of technical, economic, sociopolitical, & managerial aspects of the new initiative.
- Environment
- Competition
- Mission needs
- Stakeholder readiness
- Top level ops concept, requirements, architecture, life cycle plans

### Valuation
- Definition & analysis of scope & solution alternatives
- Human, hardware, software factors
- Mission analyses, business cases
- System/human/hardware/software build-to architecture

### Architecting
- Detailed ops concept, requirements, architecture, plans (System, increment Life Cycle Architecture packages)

### Development
- Use LCA, package for stabilized development, V&V of increment 1
- Concurrent agile change processing, rebasing of LCA\textsubscript{1}...LCA\textsubscript{n} packages

### Operation
- Operations and usage monitoring of increment 1

---

**Evaluation of evidence of feasibility to proceed**
- Ethnographic, operations analysis, models, simulations, prototypes
- Top-level feasibility rationale, trade studies, business case
- Detailed feasibility rationale, business case
- Increment, readiness for operations, LCA\textsubscript{1} feasibility rationale
- Increment, readiness for operations, LCA\textsubscript{1} feasibility rationale

**Stakeholder review and commitment**
- High, but addressable
- Acceptable
- Risk
- Too high, unaddressable
- Negligible

---

Adjust scope, priorities, or discontinue
Risk-Driven Scalable Spiral Model: Increment View

**Rapid Change**
- Short Development Increments
- Foreseeable Change (Plan)

**High Assurance**
- Stable Development Increments

**Increment N Baseline**

**Short, Stabilized Development of Increment N**

**Increment N Transition/O&M**
Risk-Driven Scalable Spiral Model: Increment View

- **Rapid Change**
  - Short Development Increments
  - Foreseeable Change (Plan)
  - Unforeseeable Change (Adapt)

- **High Assurance**
  - Stable Development Increments
  - Continuous V&V
  - Current V&V Resources

- **Increment N Baseline**
  - Agile Rebaselining for Future Increments
  - Short, Stabilized Development of Increment N
  - V&V of Increment N

- **Concerns**
  - Deferrals
  - Artifacts

- **Future Increment Baselines**

- **Increment N Transition/O&M**
  - Future V&V Resources
Risk-Driven Scalable Spiral Model: Life Cycle View

System Inception
System Elaboration

Agile DI₂ (OO&D) Rebaselining

Plan-Driven DI₁ Construction (A)
DI₁ V&V

Plan-Driven DI₂ Construction (A)
DI₂ V&V

LCA: Life Cycle Architecture
IOC: Initial Operational Capability
OO&D: Observe, Orient and Decide
V&V: Verification and Validation
DI: Development Increment
B/L: Baselined
Risk-Driven Scalable Spiral Model:
Life Cycle View

LCA: Life Cycle Architecture
IOC: Initial Operational Capability
OO&D: Observe, Orient and Decide
V&V: Verification and Validation
DI: Development Increment
B/L: Baselined
Acquisition Via Spiral OODA Loops

**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
Agile Change Processing and Rebaselining

Stabilized Increment-N Development Team

- Defer some Increment-N capabilities
- Recommend handling in current increment
- Negotiate change disposition
  - Accept changes
  - Handle Accepted Increment-N changes

Agile Future-Increment Rebaselining Team

- Assess Changes, Propose Handling
- Formulate, analyze options in context of other changes
- Handle in current rebaseline
- Recommend deferrals to future increments

Future Increment Managers

- Proposed changes
- Recommend no action, provide rationale
- Discuss, resolve deferrals to future increments
- Prepare for rebaselined future-increment development

Change Proposers

- Propose Changes
- Discuss, revise, defer, or drop

Development Team

- Discuss, revise, defer, or drop

Proposed changes

Prepare for rebaselined future-increment development
Stakeholder Commitment Ranges vs. Phase

Drivers
- System Size
- Length of Production Run
- System Criticality
- System Understanding Level
- Stakeholder Compatibility
- Personnel Capability
- Amount of New Modeling/Infrastructure
STAKEHOLDER COMMITMENT REVIEW POINTS:

Opportunities to proceed, skip phases backtrack, or terminate

Cumulative Level of Understanding, Cost, Time, Product, and Process Detail (Risk-Driven)

Concurrent Engineering of Products and Processes

Spiral View of Incremental Commitment Model

Exploration Commitment Review
Valuation Commitment Review
Architecture Commitment Review
Development Commitment Review
Operations, and Development Commitment Review
Operations, and Development Commitment Review
Annual *CrossTalk* Top-5 Projects

- Many candidate projects submitted annually
  - Providing evidence of success, key practices
- Evaluated by leading software experts
- Top-5 published in *CrossTalk*
  - DoD systems/software journal
  - 20 projects to date: 2002 – 2005
Top-5 Use of Key Process Principles

<table>
<thead>
<tr>
<th>Year</th>
<th>Concurrent Engineering</th>
<th>Risk-Driven</th>
<th>Evolutionary Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2003</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2004</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2005</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Total (of 20)</td>
<td>14</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>
## Process Principles in *CrossTalk*

### 2002 Top-5 Software Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Spiral Degree</th>
<th>Concurrent Requirements/Solution Development</th>
<th>Risk–Driven Activities</th>
<th>Evolutionary Increment Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>STARS Air Traffic Control</td>
<td>*</td>
<td>Yes</td>
<td>HCI, Safety</td>
<td>For multiple sites</td>
</tr>
<tr>
<td>Minuteman III Messaging (HAC/RMPE)</td>
<td>*</td>
<td>Yes</td>
<td>Safety</td>
<td>Yes; block upgrades</td>
</tr>
<tr>
<td>FA-18 Upgrades</td>
<td>*</td>
<td>Not described</td>
<td>Yes</td>
<td>Yes; block upgrades</td>
</tr>
<tr>
<td>Census Digital Imaging (DCS2000)</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>No; fixed delivery date</td>
</tr>
<tr>
<td>FBCB2 Army Tactical C3I</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Example ICM HCI Application:
Symbiq Medical Infusion Pump
Winner of 2006 HFES Best New Design Award
Symbiq IV Pump ICM Process - I

- **Exploration Phase**
  - Stakeholder needs interviews, field observations
  - Initial user interface prototypes
  - Competitive analysis, system scoping
  - Commitment to proceed

- **Valuation Phase**
  - Feature analysis and prioritization
  - Display vendor option prototyping and analysis
  - Top-level life cycle plan, business case analysis
  - Safety and business risk assessment
  - Commitment to proceed while addressing risks
Symbiq IV Pump ICM Process - II

• Architecting Phase
  – Modularity of pumping channels
  – Safety feature and alarms prototyping and iteration
  – Programmable therapy types, touchscreen analysis
  – Failure modes and effects analyses (FMEAs)
  – Prototype usage in teaching hospital
  – Commitment to proceed into development

• Development Phase
  – Extensive usability criteria and testing
  – Iterated FMEAs and safety analyses
  – Patient-simulator testing; adaptation to concerns
  – Commitment to production and business plans
ICM Summary

• Current processes not well matched to future challenges
  – Emergent, rapidly changing requirements
  – High assurance of scalable performance and qualities

• Incremental Commitment Model addresses challenges
  – Assurance via evidence-based milestone commitment reviews, stabilized incremental builds with concurrent V&V
    • Evidence shortfalls treated as risks
  – Adaptability via concurrent agile team handling change traffic and providing evidence-based rebaselining of next-increment specifications and plans
  – Use of critical success factor principles: stakeholder satisficing, incremental growth, concurrent engineering, iterative development, risk-based activities and milestones

• Major implications for funding, contracting, career paths
Implications for Funding, Contracting, Career Paths

• Incremental vs. total funding
  – Often with evidence-based competitive down-select

• No one-size-fits all contracting
  – Separate instruments for build-to-spec, agile rebaselining, V&V teams
    • With funding and award fees for collaboration, risk management
    • Compatible regulations, specifications, and standards
    • Compatible acquisition corps education and training
  – Generally, schedule/cost/quality as independent variable
    • Prioritized feature set as dependent variable

• Multiple career paths
  – For people good at build-to-spec, agile rebaselining, V&V
  – For people good at all three
    • Future program managers and chief engineers
Tutorial Outline

- ICM background
- Overview of Incremental Commitment Process
- **Overview of SoS arena: definitions and how SoSE differs from TSE**
  - Applying ICM to the SoS arena
  - SoS case study and examples
What is a “System of Systems”?

- Very large systems developed by creating a framework or architecture to integrate component systems
- SoS component systems independently developed and managed
  - New or existing systems in various stages of development/evolution
  - Have their own purpose
  - Can dynamically come and go from SoS
- SoS exhibits emergent behavior not otherwise achievable by component systems
- Typical domains
  - Business: Enterprise-wide and cross-enterprise integration to support core business enterprise operations across functional and geographical areas
  - Military: Dynamic communications infrastructure to support operations in a constantly changing, sometimes adversarial, environment

*Based on Mark Maier’s SoS definition [Maier, 1998]*
What is a “Lead System Integrator”?

- Organization (or set of organizations) selected to accomplish the definition and acquisition of SoS components, and the continuing integration, test, and evolution of the components and SoS
- Typical activities
  - Lead concurrent engineering of requirements, architecture, and plans
  - Identify and evaluate technologies to be integrated
  - Conduct source selection
  - Coordinate supplier activities and validate SoS architecture feasibility
  - Integrate and test SoS-level capabilities
  - Manage changes at the SoS level and across the SoS-related IPTs
  - Manage evolving interfaces to external systems
- Typically do not develop system components to be integrated (possible exception: SoS infrastructure)
What is SoSE


  The process of planning, analyzing, organizing, and integrating the capabilities of a mix of existing and new systems into a SoS capability greater than the sum of the capabilities of the constituent parts.

  **DAG:** Places emphasis on the life cycle activities to identify SoS capabilities, allocate capabilities to a set of interdependent systems, and coordination of the development and integration activities

  **SoS SE Guide:** Similar to the DAG, focuses on the delivery of capabilities through the combination of multiple collaborative, autonomous, yet interacting systems

  **USAF SAB Report:** Emphasizes the process of discovering, developing, and implementing standards that promote interoperability among systems developed via different sponsorship, management, and primary acquisition processes
What is SoSE \textit{(continued)}

- **National Centers for Systems of Systems Engineering (NCOSOSE):**
  
The design, deployment, operation, and transformation of metasystems that must function as an integrated complex system to produce desirable results. These metasystems are themselves comprised of multiple autonomous embedded complex systems that can be diverse in technology, context, operation, geography, and conceptual frame.  
(http://www.eng.odu.edu/ncsose/what_is_SOSE.shtml)
SoSE Compared to Traditional SE Activities: Reported Differences

• Architecting
  – Architecting composability vs. decomposition (Meilich 2006)
  – Net-friendly vs. hierarchical (Meilich 2006)

• Prototypes/experimentation/tradeoffs
  – Early tradeoffs/evaluations of alternatives (Finley 2006)
  – Intense concept phase analysis followed by continuous anticipation; aided by ongoing experimentation (USAF 2005)
  – Modeling and simulation, in particular to better understand “emergent behaviors” (Finley 2006)
  – First order tradeoffs above the component systems level (e.g., optimization at the SoS level, instead of at the component system level) (Garber 2006)
  – Discovery and application of convergence protocols (USAF 2005)
SoSE Compared to Traditional SE Activities: Reported Differences (continued)

• Scope and performance
  – Added “ilities” such as flexibility, adaptability, composability (USAF 2005)
  – Organizational scope defined at runtime instead of at system development time (Meilich 2006)
  – Dynamic reconfiguration of architecture as needs change (Meilich 2006)

• Maintenance and evolution
  – Component systems separately acquired and continue to be managed as independent systems (USAF 2005)
SoSE Compared to Traditional SE Activities: Key Challenges for DoD SoSE

- Business model and incentives to encourage working together at the SoS level (Garber 2006)
- Doing the necessary tradeoffs at the SoS level (Garber 2006)
- Human-system integration (Siel 2006, Meilich 2006)
- Commonality of data, architecture, and business strategies at the SoS level (Pair 2006)
- Removing multiple decision making layers (Pair 2006)
- Requiring accountability at the enterprise level (Pair 2006)
- Evolution management (Meilich 2006)
- Maturity of technology (Finley 2006)

For the most part, SoSE appears to be SE+
Particular “SE+” SoSE Concerns

- Scoping emergent SoS boundary region
  - Avoiding authority/responsibility mismatch
- Integrating multiple classes of partners
  - Subcontractors, vendors, strategic partners
  - Managing incompatible products and processes
- Combining systems engineering and acquisition management
  - Contracting and incentives for both adaptability and time-certain development
- Rapid, inclusive, validated change management
Avoiding Authority/Responsibility Mismatch

Responsibility

Authority

Infeasible

Leadership target area

Stronger Leaders

Weaker Leaders

Traditional Methods
Tutorial Outline

• ICM background
• Overview of Incremental Commitment Process
• Overview of SoS arena: definitions and how SoSE differs from TSE

• Applying ICM to the SoS arena
• SoS case study and examples
Applying ICM to the SoS Arena

- Key SoSE features driving process model
  - Potential emphasis on source selection/contract negotiations to identify SoS component systems that can provide desired functionality in desired timeframe
  - Tradeoffs at the SoS level which may not be optimal for component systems
  - Concurrent engineering at multiple levels
    - SoS-level activities
    - Activities of multiple suppliers
    - Integration of processes, risks, and schedules
- Results in multiple process models with key synchronization points
Generic View of SoS ICM – Initial Increment

1. **Source Selection**
   - Candidate Supplier/Strategic Partner 1
   - Candidate Supplier/Strategic Partner n

2. **SoS-Level**
   - Exploration
   - Valuation
   - Architecting
   - Develop
   - Operation

3. **Supplier m**
   - Exploration
   - Valuation
   - Architecting
   - Develop
   - Operation

4. **Strategic Partner C**
   - Exploration
   - Valuation
   - Architecting
   - Develop
   - Operation

5. **Supplier B**
   - Exploration
   - Valuation
   - Architecting
   - Develop
   - Operation

6. **Supplier A**
   - Exploration
   - Valuation
   - Architecting
   - Develop
   - Operation

7. **Valuation**
   - ECR
   - VCR
   - ACR/LCO
   - DCR/LCA
   - OCR

8. **Exploration**
   - Rebaseline/Adjustment LCO
   - LCO-type Proposal & Feasibility Info

9. **Develop**
   - OCR

10. **Operation**
    - OCR

11. **Rebaseline/Adjustment LCO**
Key Points for SoS ICM

• Providers for component systems may be suppliers, vendors, or strategic partners
• Suppliers that are under contract may be easier to coordinate and negotiate with, but contracts need to incorporate flexibility and incentives
• Traditional “build-to-spec” contracts are often problematic in the SoS arena, but can be used for incremental stabilized, plan-driven development activities
• Vendors and strategic partners may be more difficult to control and coordinate
• Some of the suppliers may be the SoS infrastructure supplier that needs to provide initial capability early to support integration of other component systems
  – SoS infrastructure such as net-centric standards and conventions, middleware
  – Process infrastructure such as collaboration websites, configuration management systems, and integration labs
• Desired component system providers that cannot prove first increment core capabilities on schedule will be deferred to future increments
Top-10 Risks for SISOS Development
(CrossTalk, May 2004)

1. Acquisition management and staffing
2. Requirements/architecture feasibility
3. Achievable software schedules
4. Supplier integration
5. Adaptation to rapid change
6. Quality factor achievability and tradeoffs
7. Product integration and electronic upgrade
8. Software COTS and reuse feasibility
9. External interoperability
10. Technology readiness
Focus Areas for SISOS Top-10 Risks

• Risk at the SoS level
  – Feasibility and performance of the SoS infrastructure
    • Overall security
    • Overall safety
    • Overall performance
    • Overall evolvability
  – Continuing viability of the component systems
    • Availability
    • Compatibility
    • Functionality
    • Performance
    • Cost
    • Schedule

• Component system risks
  – General product evolution risks
  – Plus concerns that SoS needs will adversely affect system’s performance, purpose, future marketability, etc.

• Constant monitoring and evaluation of risk opportunities at multiple levels is required for SoS programs

• An added option at the SoS level is to replace non-viable, poor-performing, or obsolete component systems with alternatives
SoS HSI Issues

• SoS strategic stakeholders have identified HSI as a key challenge for SoSs
  – Data/information management so that users are not overwhelmed with too much
  – Integration of new capabilities into existing platforms/systems with limited available resources, e.g.:
    • Computer utilization
    • Data storage/retrieval
    • Local network bandwidth
    • Component Graphical User Interface (GUI) compatibilities
    • Multiple views for multiple missions
    • Footprint/”real estate”
    • Number of people required to operate platform/system
    • Frequency of updates and retraining
    • Platform weight (e.g. aircraft parameter)
• Issues occur at both the SoS and the component system levels
SoS-Related Feasibility Rationales

• Plan and track progress towards generating feasibility evidence
  – Technical
    • Understand infrastructure performance and capabilities
    • Provide evidence that new SoS functionality does not significantly degrade existing component system performance
    • Provide evidence that reuse expectations are reasonable from a technical perspective
    • Challenge unrealistic capabilities/performance targets
  – Cost
    • Ensure reuse/COTS expectations are reasonable from cost perspective (e.g., tailoring, maintenance licenses, transition to new releases)
    • Challenge expensive capabilities/performance targets
  – Schedule
    • Understand component system schedules and other commitments
• Identify unvalidated targets as risks
• Monitor horizon for future significant impacts in all areas at both SoS and component system levels
ICM Anchor Point Milestone Content for SoS Development

- General ICM anchor point milestone content still applicable at SoS and component system levels
- At SoS level, major emphasis on
  - Commonality of data
  - Architecture
  - Compatibility of system component interfaces
  - Safety and security strategies
  - Business processes/strategies
  - Usability
  - Maturity of proposed newer technologies
  - Overall SoS performance against critical mission scenarios
  - Analysis of potential emergent behaviors at SoS level
- At component system level, additional emphasis on
  - New interface impacts
  - Performance impacts of other SoS component systems
  - Safety/security impacts of other SoS component systems
SoS Incremental View of Agile, Plan-Driven, and V&V Team Relationships

*May be subcontractors, vendors, or strategic partners*
SoS Changes, Change Processing, and Rebaselining

- Plan for continual change and the development of future baselines
- Most SoS changes are typically across groups and may also require contract modifications to flow down changes to multiple suppliers and vendors
- Must also negotiate changes with strategic partners
- Need to minimize impacts to increment currently under development
- Need to continually monitor evolution (changes in) the component systems for potential SoS impacts
Tutorial Outline

• ICM background
• Overview of Incremental Commitment Process
• Overview of SoS arena: definitions and how SoSE differs from TSE
• Applying ICM to the SoS arena
• SoS case study and examples
Case Study: Regional Area Crisis Response System (RACRS)
RACRS Key Features

• Goals:
  – Minimize impacts of area crises
  – Contain potential losses

• Ability to coordinate responses to regional area crises
  – Classify type of crisis
  – Alert appropriate organizations
  – Alert/evacuate public
  – Identify and manage needed resources
    • Fire trucks
    • Airplanes
    • Helicopters
    • Robots/remotely controlled vehicles
    • Medical supplies/special treatment or isolation facilities

• Request and coordinate support from other agencies: state, Federal, or other regional areas

• Strategic partnership with local news stations for live video feeds

• Support crisis management activities in other regions
RACRS Issues and Risks

• Incompatible interfaces between existing systems
• COTS products available to support interconnectivity, but have not been used at this level (potential scalability issues)
• Police and fire departments currently have on-going projects to integrate the police, fire department, and 911 systems
• Limited local budgets to modify other existing systems
• Little or no modifications expected for related State and Federal systems but expectations that these will evolve
  – Potential impacts with interfaces to other regional area systems
• Federal funds available if system implemented within the next 5 years
• Both County Board of Supervisors and City Council need to approve plans and budgets
• Citizen privacy and security issues
• Potential overlapping authorities during crises: local, state, and Federal
RACRS Desired Characteristics

• Integrate existing legacy systems together using a net-centric architecture that includes wireless, mobile networks for mobile units and existing networks for fixed Control Center connectivity
• Must work across coastal plain, intermediate mountain range, and low-lying desert area on far side of mountain range
• As part of this effort, the city and county planning and land use organizations would like to replace their location tracking systems with a new system that is based on city/county records and not the more general purpose map programs/databases typically provide by Geographic Information System (GIS) vendors
• No other new system components planned for the early versions of this SoS
• Build on existing connectivity
  – Some sort of connectivity exists between
    • City police, sheriff’s, 911, and ambulance systems
    • Jail information system and state and Federal agencies
  – Most other system components are relatively closed, independent systems
Discussion Topics to Set the Stage

• Key mission scenarios
  – Earthquake in Southern California
  – Hazardous material spill on freeway during rush hour
• Feature, service, or crisis priorities—how to define early increments
• Candidate architecture(s) and increment definitions: What can be defined as “independent projects”? How does this impact cost and schedule?
• What elements require early simulation/prototyping/evaluation?
• Risk management: What key risks should be addressed first?
• Where to be agile? Where to be plan-driven?
• Types of oversight for various types of component system providers
  – Strategic partners
  – Vendors
  – Suppliers
  – Developers
• Any assumptions?
Exercises

1. Develop ICM model for RACRS (as a group)
   - Initial increments
   - Planned prototypes, simulations, and models
   - Reviews

2. Identify elements to review in initial feasibility assessments (integrate lists from tutorial participants)

3. List candidate key risks to monitor in early increments (integrate lists from tutorial participants)
General References


SoSE-Related References


DiMario, Mike (2006); “System of Systems Characteristics and Interoperability in Joint Command Control”, Proceedings of the 2nd Annual System of Systems Engineering Conference

Electronic Industries Alliance (1999); EIA Standard 632: Processes for Engineering a System

Finley, James (2006); “Keynote Address”, Proceedings of the 2nd Annual System of Systems Engineering Conference

Garber, Vitalij (2006); “Keynote Presentation”, Proceedings of the 2nd Annual System of Systems Engineering Conference


Krygiel, A. (1999); Behind the Wizard’s Curtain; CCRP Publication Series, July, 1999, p. 33


Meilich, Abe (2006); “System of Systems Engineering (SoSE) and Architecture Challenges in a Net Centric Environment”, Proceedings of the 2nd Annual System of Systems Engineering Conference

Pair, Major General Carlos (2006); “Keynote Presentation”, Proceedings of the 2nd Annual System of Systems Engineering Conference

Proceedings of AFOSR SoSE Workshop, Sponsored by Purdue University, 17-18 May 2006


Siel, Carl (2006); “Keynote Presentation”, Proceedings of the 2nd Annual System of Systems Engineering Conference

Backup Charts
## Spiral Aspects of CrossTalk 2003 Top-5 Software Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Spiral Degree</th>
<th>Concurrent Requirements/Solution Development</th>
<th>Risk – Driven Activities</th>
<th>Evolutionary Increment Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defense Civilian Pay (DCPS)</td>
<td></td>
<td>No; waterfall</td>
<td>Yes</td>
<td>For multiple organizations</td>
</tr>
<tr>
<td>Tactical Data Radio (EPLRS)</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Joint Helmet-Mounted Cueing (JHMCS)</td>
<td>*</td>
<td>Yes; IPT-based</td>
<td>Not described</td>
<td>For multiple aircraft</td>
</tr>
<tr>
<td>Kwajalein Radar (KMAR)</td>
<td>*</td>
<td>Yes; IPT-based</td>
<td>Not described</td>
<td>For multiple radars</td>
</tr>
<tr>
<td>One SAF Simulation Testbed (OTB)</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Spiral Aspects of *CrossTalk* 2004 Top-5 Software Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Spiral Degree</th>
<th>Concurrent Requirements/Solution Development</th>
<th>Risk – Driven Activities</th>
<th>Evolutionary Increment Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Field Artillery (AFATDS)</td>
<td></td>
<td>Initially waterfall</td>
<td>Not described</td>
<td>Yes; block upgrades</td>
</tr>
<tr>
<td>Defense Medical Logistics (DMLSS)</td>
<td></td>
<td>Initially waterfall</td>
<td>Not described</td>
<td>Yes; block upgrades</td>
</tr>
<tr>
<td>F-18 HOL (H1E SCS)</td>
<td></td>
<td>Legacy requirements-driven</td>
<td>COTS, display</td>
<td>No</td>
</tr>
<tr>
<td>One SAF Objectives System (OOS)</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Patriot Excalibur (PEX)</td>
<td>**</td>
<td>Yes; agile</td>
<td>Not described</td>
<td>Yes</td>
</tr>
</tbody>
</table>
## Spiral Aspects of *CrossTalk* 2005
### Top-5 Software Projects

<table>
<thead>
<tr>
<th>Project</th>
<th>Spiral Degree</th>
<th>Concurrent Requirements/Solution Development</th>
<th>Risk – Driven Activities</th>
<th>Evolutionary Increment Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lightweight Handheld Fire Control</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Marines Integrated Pay (MCTFS)</td>
<td></td>
<td>Initially waterfall</td>
<td>Not described</td>
<td>Yes; block upgrades</td>
</tr>
<tr>
<td>Near Imaging Field Towers (NIFTI)</td>
<td>**</td>
<td>Yes; RUP based</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Smart Cam Virtual Cockpit (SC3DF)</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>WARSIM Army Training</td>
<td>**</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Primary Focus of HSI Activity Categories - II
– HSI activities often span multiple categories

<table>
<thead>
<tr>
<th>Activity Category</th>
<th>Primary Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecting Solutions</td>
<td>Facility/vehicle architecting, equipment design, component evaluation and selection, supplies/logistics planning, construction/maintenance planning</td>
</tr>
<tr>
<td>- Hardware Elements</td>
<td>Architectural style determination, component evaluation and selection, physical/logical/connector/behavioral design, evolvability design</td>
</tr>
<tr>
<td>Evaluation</td>
<td>Evidence of fitness to proceed: Solution Feasibility (usability, functionality, quality attributes, cost/schedule risk, business case/mission analysis, stakeholder commitment)</td>
</tr>
<tr>
<td></td>
<td>- Results of exercises, prototypes, simulations, models, benchmarks, analyses</td>
</tr>
<tr>
<td>Negotiating Commitments</td>
<td>Understanding stakeholder value propositions/utility functions/win conditions; dependency/compatibility/tradeoff analysis; expectations management/prioritization; option exploration; option preservation; increment sequencing</td>
</tr>
</tbody>
</table>
## Primary Focus of HSI Activity Categories - III

HSI activities often span multiple categories

<table>
<thead>
<tr>
<th>Activity Category</th>
<th>Primary Focus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development &amp; Evolution</td>
<td>Materiel/operational solution analysis; make or buy analysis; acquisition planning; source selection; contracting/incentivization; human/hardware/software element development and integration; legacy transformation preparation; incremental installation and training</td>
</tr>
<tr>
<td>Monitoring &amp; Control</td>
<td>Progress monitoring vs. plans; corrective action; adaptation of plans to change monitoring</td>
</tr>
<tr>
<td>Operations</td>
<td>Planned operations; OODA (observe, orient, decide, act) operations; adaptation of operations to change monitoring</td>
</tr>
<tr>
<td>Organizational Capability Improvement</td>
<td>Identifying, analyzing, and addressing shortfalls in current process and assessing new opportunities for improvement. Planning, executing, monitoring, and controlling capability improvement initiatives.</td>
</tr>
</tbody>
</table>
Waterfall Model Risk Profile

The most critical risks are architectural:
- Performance (size, time, accuracy)
- Interface integrity
- System qualities (adaptability, portability, etc.)
Conventional Software Process

Problem: Late Tangible Design Assessment

Standard sequence of events:
- Early and successful design review milestones
- Late and severe integration trauma
- Schedule slippage
Sequential Engineering Neglects Risk

Arch. A: Custom many cache processors

Arch. B: Modified Client-Server

$100M

$50M

Response Time (sec)
KPP Validation with ICM

- **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
Acquisition Management Implications - I

• 20th century build-to-spec contracting practices usable in part
  – Good fit for stabilized-increments team
  – But not for rebaselining, V&V teams
    • Time & materials or equivalent
    • Award fee based on cost/effectiveness
• These apply all the way down the supplier chain
• Need top-level award fee for cost-effective team balancing
  – No stable distribution of effort
Acquisition Management Implications - II

• Don’t skimp on system definition phases
  – But avoid analysis-paralysis
  – Use Feasibility evidence generation as progress metric

• Use more evidence-based source-selection processes
  – Competitive exercise as proof of capability
  – Preceded by multistage downselect

• Use Schedule/Cost as Independent Variable processes
  – Prioritized features as dependent variable

• Top priority: transformational empowerment of acquisition corps
  – Education, mentoring, tools, techniques
Failure Mode I: Build-to-Spec Deliverables—Purchasing agent metaphor

- Rapid change: heavy spec change traffic, slow contract changes
- Plus deep supplier chain: slowdowns multiply, changes interact
- Plus emergent requirements: many initial specs wrong; more changes
- Plus build-to-spec award fee: supplier inertia
- Bottom line: late rework, overruns, mission shortfalls
Failure Mode II: Sequential Document-Driven Milestones—Waterfall, V-model, MIL-STD-1521B

- Requirements emergence, COTS: freeze requirements too early
- Plus document-completion progress metrics: hasty point solutions, undiscovered risks
- Plus rapid change: problems with Failure Mode I
- Bottom line: more late rework, overruns, mission shortfalls
Failure Mode III: Hasty Best-of-Breed Source Selection

• Complexity, emergence: incomplete, unvalidated system architecture, solicitation SOWs

• Deep, wide supplier chains: incompatible legacy solutions, COTS; critical-path modeling and simulation needs

• Rapid change: rapid COTS evaluation, version obsolescence
  – GSAW: 8-11 months/release; 3 supported releases

• Bottom line: serious integration problems, overruns, mission shortfalls

Over-ambitious startup schedule...
Failure Mode IV: Incremental Document-Driven Development

- High assurance of –ilities: deferral to later increments
- Complex SISOS: early-increment architecture inadequate for later-increment –ilities.
- Rapid change: destabilization of ambitious increment schedules; increment completion delays; next increment destabilized
- Bottom line: serious security, safety, scalability problems; destabilized development; more rework, overruns, shortfalls

*Risk-insensitive “spirals”; ambitious increment schedules*
Evolution of Estimation Methods to Support SISOS Acquisition Activities

- Estimation approaches need to more closely reflect development approaches
  - Processes, both agile and traditional
  - Architecture
  - Factors for
    - Security
    - Reliability
    - Building for reuse
    - COTS incompatibilities
  - People factors
    - Politics
    - Timeliness of key decisions
    - Vendor compatibility
    - Experience levels
    - Resource availability

- Need to integrate existing “stove-pipe” cost models to better capture relationships between various teams and activities
  - Integrate SE, COTS, software development, hardware development/manufacturing
- Need to add in budgets for activities not covered in current cost models
  - LSI
  - Post development life cycle activities (e.g., installation/deployment, operation, maintenance)

*Estimates evolve over time, starting with upfront “knowns”, then expanding to include tasks identified as part of architecture decisions*
**SISOS Schedule Estimation: An Exploratory Composite Approach**

### Inception
- **Schedule** = Effort/Staff
  - Try to model ideal staff size

### Elaboration
- **Source Selection**
  - RFP, SOW, Evaluations, Contracting
  - Effort/Staff
- **SoS Architecting**
  - Assess compatibility, short-falls
  - COSOSIMO-like
  - Effort/staff at all levels

### Increment 1
- **Assess sources of change; Negotiate rebaselined LCA\(_1\) package at all levels**
- **Rework LCO \rightarrow LCA\(_1\) Packages at all levels**
- **COSOSIMO-like**
- **Effort/staff at all levels**

### Increments 2,..., n
- **Similar, with added change traffic from users...**

### Degree of Completeness
- Proposal Feasibility

### Risks
- Customers, Users
- LSI – Agile
- LSI IPTs – Agile
- Suppliers – Agile
- Suppliers – PD – V&V
- LSI – Integrators
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACR</td>
<td>Architecting Commitment Review</td>
</tr>
<tr>
<td>B/L</td>
<td>Baselined</td>
</tr>
<tr>
<td>CCD</td>
<td>Core Capability Drive-Through</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-the-Shelf</td>
</tr>
<tr>
<td>DCR</td>
<td>Development Commitment Review</td>
</tr>
<tr>
<td>DI</td>
<td>Development Increment</td>
</tr>
<tr>
<td>DOTMLPF</td>
<td>Doctrine, Organization, Training, Materiel, Leadership, Personnel, Facilities</td>
</tr>
<tr>
<td>ECR</td>
<td>Exploration Commitment Review</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure Modes and Effects Analysis</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>HSI</td>
<td>Human-System Interface</td>
</tr>
<tr>
<td>ICM</td>
<td>Incremental Commitment Model</td>
</tr>
<tr>
<td>IOC</td>
<td>Initial Operational Capability</td>
</tr>
<tr>
<td>IRR</td>
<td>Inception Readiness Review</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>LCA</td>
<td>Life Cycle Architecture</td>
</tr>
<tr>
<td>LCO</td>
<td>Life Cycle Objectives</td>
</tr>
<tr>
<td>OC</td>
<td>Operational Capability</td>
</tr>
<tr>
<td>OCR</td>
<td>Operations Commitment Review</td>
</tr>
<tr>
<td>OO&amp;D</td>
<td>Observe, Orient and Decide</td>
</tr>
<tr>
<td>OODA</td>
<td>Observe, Orient, Decide, Act</td>
</tr>
<tr>
<td>O&amp;M</td>
<td>Operations and Maintenance</td>
</tr>
<tr>
<td>PRR</td>
<td>Product Release Review</td>
</tr>
<tr>
<td>RACRS</td>
<td>Regional Area Crisis Response System</td>
</tr>
<tr>
<td>SoS</td>
<td>System of Systems</td>
</tr>
<tr>
<td>SoSE</td>
<td>System of Systems Engineering</td>
</tr>
<tr>
<td>TSE</td>
<td>Traditional Systems Engineering</td>
</tr>
<tr>
<td>VCR</td>
<td>Validation Commitment Review</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Verification and Validation</td>
</tr>
</tbody>
</table>