Improved Acquisition Processes Through Incremental Commitments

----Tutorial----

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

Presented at
NDIA 2010

Additional ICSM Information: Tech Report 2009-500 at http://csse.usc.edu
Goals of Tutorial

- Participants to understand
  - Future system acquisition challenges and needs for improvement
  - Relevant traditional acquisition failure modes and associated lessons learned
  - Ways to enable successful acquisition of desired capabilities using incremental commitments
  - How to use the Incremental Commitment Spiral Model (ICSM) in analyzing acquisition decision issues and determining appropriate acquisition and development processes
- Facilitated through
  - Case studies
  - Group exercise

Tutorial Abstract:

The wide variety of software-intensive systems needed to support the new horizons of evolving technology, system and software complexity, high dependability, global interoperability, emergent requirements, and adaptability to rapid change make traditional and current one-size-fits-all process models infeasible. This tutorial presents the process framework, principles, practices, and case studies for a new model developed and being used to address these challenges. It has a series of risk-driven decision points that enable projects to converge on whatever combination of agile, plan-driven, formal, legacy-oriented, reuse-oriented, or adaptive processes that best fit a project’s situation. The tutorial discusses the decision table for common special cases; exit ramps for terminating non-viable projects; support of concurrent engineering of requirements, solutions and plans; and evidence-based commitment milestones for synchronizing the concurrent engineering. The tutorial will include case studies and exercises for participants’ practice and discussion.
Tutorial Outline

• Current DoD acquisition challenges
  • Overview of ICSM
  • Risk-based balance of agility and assurance
  • ICSM process decision table
  • Guidance and examples for using the ICSM
  • ICSM tailoring exercise

Current DoD Acquisition Challenges

• Rapid change, emergent requirements, net-centric systems of systems make traditional DoD requirements-first, one-step acquisition infeasible
• Recent policy recognizes this
  – WSARA 2009, Section 201 (a) (2) (B)
    • “The process for developing requirements is structured to enable incremental, evolutionary, or spiral acquisition approaches ....”
  – DoDI 5000.02, Enclosure 2.a
    • “Evolutionary acquisition is the preferred strategy for rapid acquisition of mature technology for the user.”
• But DoD system acquisition is still dominated by obsolete contracting processes and instruments
This chart shows data from two complex systems of systems that illustrates the challenge of executing tight OODA loops that involve coordinating changes across closely coupled systems of systems. The average time in workdays (longer than calendar days) was 27 workdays to close a change request within an individual platform or capability group; 48 workdays if the change required coordination across multiple platform or capability groups; and 141 workdays if the change involved a change in performers’ contracts. These were typical high-content U.S. build-to-specification contracts, which [Hall, 1976] found averaged 10 times as long as high-context French contracts.

Corroborative data was provided at a workshop by [Schroeder, 2007], who found that the length of such changes was proportional to the number of contracts requiring changes. Other experienced sources have reported even longer contract change turnaround times for large complex systems. Clearly, improvements are needed in change-facilitating architectures, processes, and contracts if DoD is to stand any chance of performing evolutionary acquisition in ways that keep its change-adapting OODA loops within those of its adversaries.
Most DoD systems and the acquisition processes that developed them were organized to function in set-piece warfare situations, and have not been well-suited to function in current and future asymmetric, irregular, and hybrid warfare situations. Recent warfare experiences in Iraq and Afghanistan, and large-system response-time data shown on the next chart, illustrate this challenge to DoD SE and EvA.

In the early stages of the Iraq war, DoD forces were phenomenally successful. They could pick the time and place of attacks, and their highly superior command, control, communications, computing, intelligence, surveillance, and reconnaissance (C4ISR) capabilities enabled them to perform observe-orient-decide-act (OODA) loops well inside those of their conventional Iraqi army adversaries.

However, in the later stages of the Iraq war, and in Afghanistan, DoD faced much more serious challenges that are representative of many future conflict situations. Their adversaries were diffuse. They could pick the time and place of an attack. They had little to lose, and could use very agile, lightweight systems and processes, while being able to use powerful but hazardous hardware and software at relatively little risk.

On the other hand, DoD forces must be ready for anything at any time, and must be able to defend much more valuable assets. This requires more heavyweight systems and processes in order to develop and operate high-assurance systems, and restricts the use of untrusted components. Even for individual systems, this causes significant challenges in turning OODA loops inside those of one’s adversaries, but as seen next, the challenges will be even higher for complex systems of systems.
Need for Evolution-Compatible Acquisition Capabilities

<table>
<thead>
<tr>
<th>Traditional Metaphor: Purchasing Agent</th>
<th>Needed Metaphor: C2ISR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete, consistent, testable requirements before design</td>
<td>Concurrent engineering of requirements and solutions</td>
</tr>
<tr>
<td>Single-step development</td>
<td>Evolutionary, incremental system definition and development</td>
</tr>
<tr>
<td>One-size-fits-all acquisition instruments</td>
<td>Selectable, tailorable acquisition instruments</td>
</tr>
<tr>
<td>Tailorable down from monolithic base</td>
<td>Tailorable up via risk-driven checklist</td>
</tr>
<tr>
<td>Premium on ambitious performance</td>
<td>Premium on acquisition speed, system flexibility, assurance, total ownership cost</td>
</tr>
</tbody>
</table>

Current System Acquisition Methods
Too easy to misinterpret as one-size-fits-all

- **V-Model**
- **Spiral Model**

High level guidance assumes that acquirers have extensive acquisition experience... Without experience, too easy to misinterpret and auger in with disastrous results...
Tutorial Outline

- Current DoD acquisition challenges
- **Overview of ICSM**
- Risk-based balance of agility and assurance
- ICSM process decision table
- Guidance and examples for using the ICSM
- ICSM tailoring exercise
What is the ICSM?

- Risk-driven framework for determining and evolving best-fit system life-cycle process
- Integrates the strengths of phased and risk-driven spiral process models
- Synthesizes together principles critical to successful system development
  - Stakeholder value-based system definition and evolution
  - Incremental commitment and accountability
  - Concurrent system definition and development
  - Evidence and risk-driven decisionmaking

Principles used by 60-80% of CrossTalk Top-5 projects, 2002-2005

ICSM Nature and Origins

- Integrates hardware, software, and human factors elements of systems engineering
  - Concurrent exploration of needs and opportunities
  - Concurrent engineering of hardware, software, human aspects
  - Concurrency stabilized via anchor point milestones
- Developed in response to a variety of issues
  - Clarify “spiral development” usage
    - Initial phased version (2005)
    - Provide framework for human-systems integration
- Integrates strengths of current process models
  - But not their weaknesses
- Improves teaching of software project courses
Incremental Commitment in Gambling

A simple metaphor to help understand the ICSM is to compare ICSM to gambling games such as poker and blackjack, to single-commitment gambling games such as Roulette. Many system development contracts operate like Roulette, in which a full set of requirements is specified up front, the full set of resources is committed to an essentially fixed-price contract, and one waits to see if the bet was a good one or not. With the ICSM, one places a smaller bet to see whether the prospects of a win are good or not, and decides to increase the bet based on better information about the prospects of success.
Scalable remotely controlled operations – ICSM Case Study

An example to illustrate ICSM benefits is the Unmanned Aerial Vehicle (UAV) (or Remotely Piloted Vehicles (RPV)) system enhancement discussed in Chapter 5 of the NRC HSI report [Pew and Mavor, 2007]. The RPVs are airplanes or helicopters operated remotely by humans. These systems are designed to keep humans out of harm’s way. However, the current system is human-intensive, requiring two people to operate a single vehicle. If there is a strong desire to modify the 2:1 (2 people to one vehicle) ratio to allow for a single operator and 4 aircraft (e.g., a 1:4 ratio), based on a proof-of-principle agent-based prototype demo showing 1:4 performance of some RPV tasks, how should one proceed?
Total vs. Incremental Commitment – 4:1 RPV

Total Commitment
- Agent technology demo and PR: Can do 4:1 for $1B
- Winning bidder: $800M; PDR in 120 days; 4:1 capability in 40 months
- PDR: many outstanding risks, undefined interfaces
- $800M, 40 months: “halfway” through integration and test
- 1:1 IOC after $3B, 80 months

Incremental Commitment [with a number of competing teams]
- $25M, 6 mo. to VCR [4]: may beat 1:2 with agent technology, but not 4:1
- $75M, 8 mo. to FCR [3]: agent technology may do 1:1; some risks
- $225M, 10 mo. to DCR [2]: validated architecture, high-risk elements
- $675M, 18 mo. to IOC [1]: viable 1:1 capability
- 1:1 IOC after $1B, 42 months

This slide outlines two approaches to the RPV question: total commitment and incremental commitment. While this is a hypothetical case for developing a solution to the RPV manning problem, it shows how a premature total commitment without significant modeling, analysis, and feasibility assessment will often lead to large overruns in costs and schedule, and a manning ratio that is considerably less than initially desired. However, by “buying information” early and validating high-risk elements, the more technologically viable option is identified much earlier and can be provided for a much lower cost and much closer to the desired date. The ICSM approach leads to the same improved manning ratio as the total commitment approach, but sooner and at a much reduced cost.

The ICSM approach also employs a competitive downselect strategy, which both reduces risk and enables a buildup of trust among the acquirers, developers, and users.
The Incremental Commitment Life Cycle Process: Overview

This slide shows how the ICSM spans the life cycle process from concept exploration to operations. Each phase culminates with an anchor point milestone review. At each anchor point, there are 4 options, based on the assessed risk of the proposed system. Some options involve go-backs. These options result in many possible process paths.

The life cycle is divided into two stages: Stage I of the ICSM (Definition) has 3 decision nodes with 4 options/node, culminating with incremental development in Stage II (Development and Operations). Stage II has an additional 2 decision nodes, again with 4 options/node.

One can use ICSM risk patterns to generate frequently-used processes with confidence that they fit the situation. Initial risk patterns can generally be determined in the Exploration phase. One then proceeds with development as a proposed plan with risk-based evidence at the VCR milestone, adjusting in later phases as necessary. For complex systems, a result of the Exploration phase would be the Prototyping and Competition Plan discussed above.

Risks associated with the system drive the life cycle process. Information about the risk(s) (feasibility assessments) supports the decision to proceed, adjust scope or priorities, or cancel the program.
ICSM HSI Levels of Activity for Complex Systems

As mentioned earlier, with the ICSM, a number of system aspects are being concurrently engineered at an increasing level of understanding, definition, and development. The most significant of these aspects are shown in this slide, an extension of a similar view of concurrently engineered software projects developed as part of the RUP (shown in a backup slide).

As with the RUP version, it should be emphasized that the magnitude and shape of the levels of effort will be risk-driven and likely to vary from project to project. In particular, they are likely to have mini risk/opportunity-driven peaks and valleys, rather than the smooth curves shown for simplicity in this slide. The main intent of this view is to emphasize the necessary concurrency of the primary success-critical activities shown as rows. Thus, in interpreting the Exploration column, although system scoping is the primary objective of the Exploration phase, doing it well involves a considerable amount of activity in understanding needs, envisioning opportunities, identifying and reconciling stakeholder goals and objectives, architecting solutions, life cycle planning, evaluation of alternatives, and negotiation of stakeholder commitments.
Anchor Point Feasibility Evidence Descriptions

- **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

*Can be used to strengthen current schedule- or event-based reviews*

Anchor Point Feasibility Rationales

To make ICSM concurrency work, the anchor point milestone reviews are the mechanism by which the many concurrent activities are synchronized, stabilized, and risk-assessed at the end of each phase. Each of these anchor point milestone reviews is focused on developer-produced evidence, documented in a Feasibility Evidence Description (FED), to help the key stakeholders determine the next level of commitment. At each program milestone/anchor point, feasibility assessments and the associated evidence are reviewed and serve as the basis for the stakeholders’ commitment to proceed.

The FED is not just a document, a set of PowerPoint charts, or Unified Modeling Language (UML) diagrams. It is based on evidence from simulations, models, or experiments with planned technologies and detailed analysis of development approaches and projected productivity rates. The detailed analysis is often based on historical data showing reuse realizations, software size estimation accuracy, and actual developer productivity rates.

It is often not possible to fully resolve all risks at a given point in the development cycle, but known, unresolved risks need to be identified and covered by risk management plans.
The Incremental Commitment Life Cycle Process: More on the Overview

Stage II of the Incremental Commitment Life Cycle provides a framework for concurrent engineering and development of multiple increments. More on this concurrency follows on the next slides.

Note: The term “concurrent engineering” fell into disfavor when behind-schedule developers applied it to the practice of proceeding into development while the designers worked on finishing the design. Not surprisingly, the developers encountered a high rework penalty for going into development with weak architecture and risk resolution.

“Concurrent engineering” as applied in the ICSM is much different. It is focused on doing a cost-effective job of architecture and risk resolution in Stage I; and on performing stabilized development, verification, and validation of the current system increment while concurrently handling the systems change traffic and preparing a feasibility-validated architecture and set of plans for the next increment in Stage II.
Agile Change Processing and Rebaselining

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)

- **Foundations Commitment Review (FCR)**
  - 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

### ICSM Anchor Point Milestone Content (1)
(Risk-driven level of detail for each element)

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Foundations Commitment Review</th>
<th>Development Commitment Review</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition of Operational Concept</td>
<td>- Top-level system objectives and scope</td>
<td>- Elaboration of system objectives and scope of increment</td>
</tr>
<tr>
<td></td>
<td>- System boundary</td>
<td>- Elaboration of operational concept by increment</td>
</tr>
<tr>
<td></td>
<td>- Environment parameters and assumptions</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Evolution parameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Operational concept</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Operations and maintenance scenarios and parameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Organizational life-cycle responsibilities (stakeholders)</td>
<td></td>
</tr>
<tr>
<td>System Prototype(s)</td>
<td>- Exercise key usage scenarios</td>
<td>- Exercise range of usage scenarios</td>
</tr>
<tr>
<td></td>
<td>- Resolve critical risks</td>
<td>- Resolve major outstanding risks</td>
</tr>
<tr>
<td>Definition of System Requirements</td>
<td>- Top-level functions, interfaces, quality attribute levels,</td>
<td>- Elaboration of functions, interface, quality attributes, and</td>
</tr>
<tr>
<td></td>
<td>including growth vectors and priorities</td>
<td>prototypes by increment</td>
</tr>
<tr>
<td></td>
<td>- Prototypes</td>
<td>Identification of TBD’s (to-be-determined items)</td>
</tr>
<tr>
<td></td>
<td>- Stakeholders’ concurrence on essence</td>
<td>Stakeholders’ concurrence on their priority concerns</td>
</tr>
</tbody>
</table>
# ICSM Anchor Point Milestone Content (2)

## Risk-driven level of detail for each element

<table>
<thead>
<tr>
<th>Milestone Element</th>
<th>Foundations Commitment Review</th>
<th>Development Commitment Review</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 WWWWWW+ by stage |  - Elaboration of WWWWWW+ for Initial Operational Capability (IOC)  
  - Partial elaboration, identification of key TDDs for later increments |
| Feasibility Evidence             |  - 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 |

Example ICSM Commercial Application: Symbiq Medical Infusion Pump
Winner of 2006 HFES Best New Design Award
Described in NRC HSI Report, Chapter 5

Example ICSM HCI Application: Symbiq Medical Infusion Pump

This next-generation infusion pump is a general-purpose intravenous infusion pump (IV pump) designed primarily for hospital use with secondary, limited-feature use by patients at home. The device is intended to deliver liquid medications, nutrients, blood, and other solutions at programmed flow rates, volumes, and time intervals via intravenous and other routes to a patient. The marketed name is the Symbiq IV Pump. The device offers medication management features, including medication management safety software through a programmable drug library. The infuser also has sufficient memory to support extensive tracking logs and the ability to communicate and integrate with hospital information systems. The infuser is available as either a single-channel pump or a dual-channel pump. The two configurations can be linked together to form a 3- or 4-channel pump. The infuser includes a large touchscreen color display and can be powered by either A/C power or rechargeable batteries.
(adapted from NRC HSI Report, Chapter 5)
Symbiq IV Pump ICSM 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 ICSM 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
ICSM Summary

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

- ICSM 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 value-based, incremental commitment and accountability, concurrent system definition and development, evidence and risk-driven decisionmaking

- Major implications for funding, contracting, career paths

October 2010

Copyright © U.S.C. © SE

Tutorial Outline

- Current DoD acquisition challenges
- Overview of ICSM
- **Risk-based balance of agility and assurance**
- ICSM process decision table
- Guidance and examples for using the ICSM
- ICSM tailoring exercise
Using Risk to Balance Assurance and Agility - Overview

Risks Used in Risk Comparison

- **Environmental risks**
  - Technology uncertainties
  - Many stakeholders
  - Complex system-of-systems
  - Legacy compatibility

- **Agility risks**
  - Scalability
  - Use of simple design
  - Personnel turnover
  - Too-frequent releases
  - Not enough agile-capable people

- **Plan-driven risks**
  - Rapid change
  - Need for rapid results
  - Emergent requirements
  - Not enough discipline-capable people
Tutorial Outline

• Current DoD acquisition challenges
• Overview of ICSM
• Risk-based balance of agility and assurance
• **ICSM process decision table**
• Guidance and examples for using the ICSM
• ICSM tailoring exercise
The Incremental Commitment Life Cycle Process: Overview

This slide shows how the ICSM spans the life cycle process from concept exploration to operations. Each phase culminates with an anchor point milestone review. At each anchor point, there are 4 options, based on the assessed risk of the proposed system. Some options involve go-backs. These options result in many possible process paths.

The life cycle is divided into two stages: Stage I of the ICSM (Definition) has 3 decision nodes with 4 options/node, culminating with incremental development in Stage II (Development and Operations). Stage II has an additional 2 decision nodes, again with 4 options/node.

One can use ICSM risk patterns to generate frequently-used processes with confidence that they fit the situation. Initial risk patterns can generally be determined in the Exploration phase. One then proceeds with development as a proposed plan with risk-based evidence at the VCR milestone, adjusting in later phases as necessary. For complex systems, a result of the Exploration phase would be the Prototyping and Competition Plan discussed above.

Risks associated with the system drive the life cycle process. Information about the risk(s) (feasibility assessments) supports the decision to proceed, adjust scope or priorities, or cancel the program.
The ICSM as Risk-Driven Process Generator

- Stage I of the ICSM has 3 decision nodes with 4 options/node
  - Culminating with incremental development in Stage II
  - Some options involve go-backs
  - Results in many possible process paths
- Can use ICSM risk patterns to generate frequently-used processes
  - With confidence that they fit the situation
- Can generally determine this in the Exploration phase
  - Develop as proposed plan with risk-based evidence at VCR milestone
  - Adjustable in later phases
Different Risk Patterns Yield Different Processes

As illustrated in the four example paths through the Incremental Commitment Model in this slide, the ICSM is not a single monolithic one-size-fits-all process model. As with the spiral model, it is a risk-driven process model generator, but the ICSM makes it easier to visualize how different risks create different processes.

In Example A, a simple business application based on an appropriately-selected Enterprise Resource Planning (ERP) package, there is no need for a Valuation or Foundations activity if there is no risk that the ERP package and its architecture will not cost-effectively support the application. Thus, one could go directly into the Development phase, using an agile method such as a Scrum/Extreme Programming combination would be a good fit. There is no need for Big Design Up Front (BDUF) activities or artifacts because an appropriate architecture is already present in the ERP package. Nor is there a need for heavyweight waterfall or V-model specifications and document reviews. The fact that the risk at the end of the Exploration phase is negligible implies that sufficient risk resolution of the ERP package’s human interface has been done.

Example B involves the upgrade of several incompatible legacy applications into a service-oriented web-based system. Here, one could use a sequential waterfall or V-model if the upgrade requirements were stable, and its risks were low. However, if for example the legacy applications’ user interfaces were incompatible with each other and with web-based operations, a concurrent risk-driven spiral, waterfall, or V-model that develops and exercise extensive user interface prototypes and generates a Feasibility Evidence Description (described on chart 12) would be preferable.
In Example C, the stakeholders may have found during the Valuation phase that their original assumptions about the stakeholders having a clear, shared vision and compatible goals with respect to the proposed new system’s concept of operation and its operational roles and responsibilities were optimistic. In such a case, it is better to go back and assure stakeholder value proposition compatibility and feasibility before proceeding, as indicated by the arrow back into the valuation phase.

In Example D, it is discovered before entering the Development phase that a superior product has already entered the marketplace, leaving the current product with an infeasible business case. Here, unless a viable business case can be made by adjusting the project’s scope, it is best to discontinue it. It is worth pointing out that it is not necessary to proceed to the next major milestone before terminating a clearly non-viable project, although stakeholder concurrence in termination is essential.

---

**The ICSM Process Decision Table:**

**Key Decision Inputs**

- Product and project size and complexity
- Requirements volatility
- Mission criticality
- Nature of Non-Developmental/COTS item support
  - Commercial, open-source, reused components
- Organizational and Personnel Capability
The ICSM Process Decision Table:
Key Decision Outputs

- Key Stage I activities: incremental definition
- Key Stage II activities: incremental development and operations
- Suggested calendar time per build, per deliverable increment

Common Risk-Driven Special Cases of the ICSM (Cases 1-4)

<table>
<thead>
<tr>
<th>Case 1: Use NDI</th>
<th>Case 2: Agile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Small accounting system</td>
<td><strong>Example:</strong> Exercises</td>
</tr>
<tr>
<td>Size, Complexity: Low, complexity low</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: Negligible</td>
<td></td>
</tr>
<tr>
<td>Criticality: Low to medium</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Complete</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: NDI-oriented (medium)</td>
<td></td>
</tr>
<tr>
<td>Key Stage II Activities: Incremental Development/Operation(s): Use NDI</td>
<td></td>
</tr>
<tr>
<td>Time Build: n/a</td>
<td></td>
</tr>
<tr>
<td>Time Increment: Vendor-driven</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3: Architected Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Business data processing</td>
<td><strong>Example:</strong> Security protocol: Safety-critical UAT chip</td>
</tr>
<tr>
<td>Size, Complexity: Medium</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 1-10%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Medium to high</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, most in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium to high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage I Activities: Incremental Development: Incremental Development/Operation(s): Architect妄想ism of System(s)</td>
<td></td>
</tr>
<tr>
<td>Time Build: 2-4 weeks</td>
<td></td>
</tr>
<tr>
<td>Time Increment: 2-6 months</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 2: Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Experiments</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Low</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 10-50%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Low to medium</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium-high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage II Activities: Incremental Development/Operation(s): Use NDI</td>
<td></td>
</tr>
<tr>
<td>Time Build: n/a</td>
<td></td>
</tr>
<tr>
<td>Time Increment: Vendor-driven</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3: Architected Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Business data processing</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Medium</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 1-10%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Medium to high</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, most in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium to high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage I Activities: Incremental Development: Incremental Development/Operation(s): Architect妄想ism of System(s)</td>
<td></td>
</tr>
<tr>
<td>Time Build: 2-4 weeks</td>
<td></td>
</tr>
<tr>
<td>Time Increment: 2-6 months</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 2: Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Experiments</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Low</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 10-50%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Low to medium</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium-high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage II Activities: Incremental Development/Operation(s): Use NDI</td>
<td></td>
</tr>
<tr>
<td>Time Build: n/a</td>
<td></td>
</tr>
<tr>
<td>Time Increment: Vendor-driven</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3: Architected Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Business data processing</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Medium</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 1-10%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Medium to high</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, most in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium to high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage I Activities: Incremental Development: Incremental Development/Operation(s): Architect妄想ism of System(s)</td>
<td></td>
</tr>
<tr>
<td>Time Build: 2-4 weeks</td>
<td></td>
</tr>
<tr>
<td>Time Increment: 2-6 months</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 2: Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Experiments</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Low</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 10-50%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Low to medium</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium-high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage II Activities: Incremental Development/Operation(s): Use NDI</td>
<td></td>
</tr>
<tr>
<td>Time Build: n/a</td>
<td></td>
</tr>
<tr>
<td>Time Increment: Vendor-driven</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 3: Architected Agile</th>
<th>Case 4: Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Example:</strong> Business data processing</td>
<td></td>
</tr>
<tr>
<td>Size, Complexity: Medium</td>
<td></td>
</tr>
<tr>
<td>Typical Change Raw/Min: 1-10%</td>
<td></td>
</tr>
<tr>
<td>Criticality: Medium to high</td>
<td></td>
</tr>
<tr>
<td>NDI Support: Good, most in place</td>
<td></td>
</tr>
<tr>
<td>Organizational/Personnel Capability: Agile-ready, medium to high experience</td>
<td></td>
</tr>
<tr>
<td>Key Stage I Activities: Incremental Development: Incremental Development/Operation(s): Architect妄想ism of System(s)</td>
<td></td>
</tr>
<tr>
<td>Time Build: 2-4 weeks</td>
<td></td>
</tr>
<tr>
<td>Time Increment: 2-6 months</td>
<td></td>
</tr>
</tbody>
</table>
## Common Risk-Driven Special Cases of the ICSM (Cases 5-8)

<table>
<thead>
<tr>
<th>Case 5: Hardware with Embedded Software Component</th>
<th>Case 6: Indivisible IOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Maintenance control device</td>
<td>Example: Complex vehicle platform</td>
</tr>
<tr>
<td>Site Complexity: Low</td>
<td>Site Complexity: Medium to high</td>
</tr>
<tr>
<td>Typical Change Raw Material: 0.3 - 1.0</td>
<td>Typical Change Raw Material: 0.3 - 1.0</td>
</tr>
<tr>
<td>Criticality: Medium to very high</td>
<td>Criticality: High to very high</td>
</tr>
<tr>
<td>NDIR Support: On-shelf, in plant</td>
<td>NDIR Support: Some in plant</td>
</tr>
<tr>
<td>Organizational Personnel Capability: Experimentation, medium high</td>
<td>Organizational Personnel Capability: Experimentation, medium to high</td>
</tr>
<tr>
<td>Key Stages I Activity (Incremental Definitions): Consistent hardwaresoftware engineering, CDR-level ICSM DCR</td>
<td>Key Stages I Activity (Incremental Definitions): Determine minimum IOC likely, conservative cost. Add deliverable software features as not required</td>
</tr>
<tr>
<td>Key Stages II Activity (Incremental Development/Operation): IOC development, LRP, FPR, Consistent version N+1 engineering</td>
<td>Key Stages II Activity (Incremental Development/Operation): Deep deliverable features most conservative cost. Being saved for the features not dropped</td>
</tr>
<tr>
<td>Time Build: Software: 1-4 weeks</td>
<td>Time Build: Software: 1-4 weeks</td>
</tr>
<tr>
<td>Time Increase: Medium-drum</td>
<td>Time Increase: Platform: 6-12 months</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 7: NDI-Intensive</th>
<th>Case 8: Hybrid Agile/Plug-Driven System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: High density management</td>
<td>Example: PABEX system</td>
</tr>
<tr>
<td>Site Complexity: Medium to high</td>
<td>Site Complexity: Medium to very high</td>
</tr>
<tr>
<td>Typical Change Raw Material: 0.3 - 3.0</td>
<td>Typical Change Raw Material: Mixed parts, 1-10 percent</td>
</tr>
<tr>
<td>Criticality: Medium to very high</td>
<td>Criticality: Mixed parts, Medium to very high</td>
</tr>
<tr>
<td>NDIR Support: NDI-driven architecture</td>
<td>NDIR Support: Mixed parts</td>
</tr>
<tr>
<td>Organizational Personnel Capability: NDI-optimized, medium to high</td>
<td>Organizational Personnel Capability: Mixed parts</td>
</tr>
<tr>
<td>Key Stages I Activity (Incremental Definitions): Thorough NDI study</td>
<td>Key Stages I Activity (Incremental Definitions): Full ICSM, encapsulated agile in high change, low-medium stability parts (often NDI, internal interfaces)</td>
</tr>
<tr>
<td>Key Stages II Activity (Incremental Development/Operation): Pre-Native NDI evolution influencing NDI upgrade synchronization</td>
<td>Key Stages II Activity (Incremental Development/Operation): Full ICSM, three-stage incremental development, consistent V&amp;V, incremental retooling</td>
</tr>
<tr>
<td>Time Build: Software: 1-4 weeks</td>
<td>Time Build: 1-2 months</td>
</tr>
<tr>
<td>Time Increase: Systems: 6-12 months</td>
<td>Time Increase: Platform: 6-12 months</td>
</tr>
</tbody>
</table>

## Common Risk-Driven Special Cases of the ICSM (Cases 9-11)

<table>
<thead>
<tr>
<th>Case 9: Multi-Ower Directed System of Systems</th>
<th>Case 10: Family of Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Nineteen military operations</td>
<td>Example: Medieval sword pedestal line</td>
</tr>
<tr>
<td>Site Complexity: Very high</td>
<td>Site Complexity: Medium to very high</td>
</tr>
<tr>
<td>Typical Change Raw Material: 1.5 - 3</td>
<td>Typical Change Raw Material: 1.5</td>
</tr>
<tr>
<td>Criticality: High to very high</td>
<td>Criticality: Medium to very high</td>
</tr>
<tr>
<td>NDIR Support: Many NDI, some in plant</td>
<td>NDIR Support: Some in plant</td>
</tr>
<tr>
<td>Organizational Personnel Capability: Related expertise, medium to high</td>
<td>Organizational Personnel Capability: Related expertise, medium to high</td>
</tr>
<tr>
<td>Key Stages I Activity (Incremental Definitions): Full ICSM, commonality and team building, operation</td>
<td>Key Stages I Activity (Incremental Definitions): High Validation and Anti-Priming phases</td>
</tr>
<tr>
<td>Key Stages II Activity (Incremental Development/Operation): Full ICSM, three-stage incremental development, consistent V&amp;V, incremental retooling</td>
<td>Key Stages II Activity (Incremental Development/Operation): Stream line agile methods of design</td>
</tr>
<tr>
<td>Time Build: 1-4 months</td>
<td>Time Build: 1-2 months</td>
</tr>
<tr>
<td>Time Increase: 15-24 months</td>
<td>Time Increase: 1-12 months</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case 11: Brompton</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Example: Incremental legacy processes</td>
<td>Example: Incremental legacy processes</td>
</tr>
<tr>
<td>Site Complexity: High to very high</td>
<td>Site Complexity: High to very high</td>
</tr>
<tr>
<td>Criticality: Medium to very high</td>
<td>Criticality: Medium to very high</td>
</tr>
<tr>
<td>NDIR Support: NDI as legacy enhancement</td>
<td>NDIR Support: NDI as legacy enhancement</td>
</tr>
<tr>
<td>Organizational Personnel Capability: Legacy re-engineering</td>
<td>Organizational Personnel Capability: Legacy re-engineering</td>
</tr>
<tr>
<td>Key Stages I Activity (Incremental Definitions): Re-engineering legacy into services</td>
<td>Key Stages I Activity (Incremental Definitions): Re-engineering legacy into services</td>
</tr>
<tr>
<td>Key Stages II Activity (Incremental Development/Operation): Incremental legacy processes</td>
<td>Key Stages II Activity (Incremental Development/Operation): Incremental legacy processes</td>
</tr>
<tr>
<td>Time Build: 1-2 months</td>
<td>Time Build: 1-2 months</td>
</tr>
<tr>
<td>Time Increase: 1-6 months</td>
<td>Time Increase: 1-6 months</td>
</tr>
</tbody>
</table>
Common Risk-Driven Special Cases of the ICSM (Cases 12a/b)

Case 12a: Net-Centric Services – Community Support
- Example: Community services or special interest group
- Size, Complexity: Low to medium
- Criticality: Low to medium
- NIS Support: Tailorable service elements
- Organizational Personnel Capability: NIS-competent
- Key Stages II Activities: Incremental Development, Operations
- Time to Build: ~1 day
- Time to Increase: 1-12 months

Case 12b: Net-Centric Services – Quick Response Decision Support
- Example: Response to computer initiative
- Size, Complexity: Medium to high
- Criticality: Medium or high
- NIS Support: Tailorable service elements
- Organizational Personnel Capability: NIS-competent
- Key Stages II Activities: Incremental Development, Operations
- Time to Build: ~1 day
- Time to Increase: Quick response-driven

Tutorial Outline

- Current DoD acquisition challenges
- Overview of ICSM
- Risk-based balance of agility and assurance
- ICSM process decision table
- Guidance and examples for using the ICSM
- ICSM tailoring exercise
Case 3: For medium-size (20-80 people), medium complexity (reasonably mature and scalable technology; largely compatible shareholders), agile methods can be scaled using an Architected Agile approach with early investment in a largely change-prescient architecture and user/developer/customer team building. For relatively stable projects (0.3-1% change/month), plan-driven methods can be used with low risk. But for higher rates of changes (1-10%/month), a more agile approach is less risky. A risk analysis of a 50-person, medium sized architecture-based agile supply chain management project is provided on pages 106-121 of (Boehm and Turner, 2004). A number of organizations in such areas as corporate infrastructure, medical, aerospace, and ERP applications have reported significant gains in adaptability and quality of the Architected Agile approach over plan-driven methods for such projects. However, others that had less capable and agile-ready people, less management and customer commitment, and less up-front architecture investment have not. (Boehm, 2007)
USA Medical Case Study

- 1400 software people; 7M SLOC; 7 sites
  - 4 in Europe, 2 in India
- 500 medical applications; 500 financial; others
- Survivability-critical software problems
  - Reliability, productivity, performance, interoperability
  - Sarbanes-Oxley requirements
  - Management receptive to radical change
- Some limited experimental use of agile methods
  - Led by top software technologist/manager
- Committed to total change around Scrum and XP

USA Medical Adoption Profile

- July 2004 - July 2005
  - Recruit top people from all sites into core team(s)
  - Get external expert help
  - Develop architecture
  - Early Scrum successes with infrastructure
  - Revise policies and practices
  - Train, reculture everyone
  - Manage expectations
- July 2005 - July 2006
  - Begin full-scale development
  - Core teams as mentors
Architected Agile – USA Medical

• Include customers and marketers
  – New roles; do's/don'ts/opportunities; CRACK personnel; full collaboration and teamwork; expectations management

• Scrum; most XP practices; added company practices
  – 6-12 person teams with team rooms, dedicated servers
  – Hourly smoke test; nightly build and regression test
  – Just-in-time analysis; story-point estimates; fail fast; detailed short-term plans; company architecture compliance
  – Embrace change in applications and practices
  – Global teams: wikis, daily virtual meetings, act as if next-door

• Release management
  – 2-12 week architecting Sprint Zero; 3-10 1-month Sprints; Release Sprint; 1-6 month beta test
  – Next Sprint Zero concurrent with Release Sprint

• Initiative manager and team
  – Define practices; evolve infrastructure; provide training; guide implementation; evaluate compliance/usage; continuous improvement
Case 7: NDI-Intensive

- Biggest risks: incompatible NDI; rapid change, business/mission criticality; low NDI assessment and integration experience; supply chain stakeholder incompatibilities
- Example: Supply chain management
- Size/complexity: Medium to high
- Anticipated change rate (% per month): 0.3-3%
- Criticality: Medium to very high
- NDI support: NDI-driven architecture
- Organizational and personnel capability: NDI-experienced; medium to high capability
- Key Stage I activities: Thorough NDI-suite life cycle cost-benefit analysis, selection, concurrent requirements and architecture definition
- Key Stage II activities: Pro-active NDI evolution influencing, NDI upgrade synchronization
- Time/build: 1-4 weeks (software)
- Time/increment: 6-18 months (systems)

Case 7: Our experiences in developing USC web-service applications between 1996 and 2004 was that they went from 28% of the application’s functionality being delivered by NDI components to 80% (Yang et al, 2005). A similar trend was identified by the 2001 Standish Report, which reported that 53% of the functionality of commercial software applications was being delivered by NDI components in 2000 (Standish, 2001). The economics of NDI-intensive systems dictates a bottom-up versus a top-down approach to system development, in which the capability envelope of the NDI determines the affordable requirements, rather than a top-down requirements-to-capability approach. A large supply-chain management system may need to choose among several NDI candidates each for such functions as inventory control, trend analysis, supplier/customer relations management, transportation planning, manufacturing control, and financial transactions; and evaluate not only the candidates’ cost/performance aspects, but also their interoperability with each other and with the corporation’s legacy infrastructure. Besides NDI assessment, other significant sources of effort can be NDI tailoring, NDI integration, and effect of NDI version volatility and obsolescence; see (Yang et al, 2005).

A particular challenge in Stage II is the effect of NDI volatility and obsolescence. Surveys have indicated that commercial NDI products have a new release about every 10 months, and that old releases are supported by the vendor for about 3 releases. Some large systems have had about 120 NDI components, indicating that about 12 components will have new releases each month, and that not upgrading will leave each component unsupported in about 30 months. In such cases, a great deal of attention needs to be paid to upgrade synchronization, and to pro-active NDI evolution influencing. Some large organizations synchronize their NDI upgrades to their major re-training cycles of about 12-18 months. For additional NDI best practices, see (Wallnau et al, 2002).
This figure presents the overall CBA decision framework that composes the assessment, tailoring, glue code, and custom code development process elements within an overall development lifecycle. This framework presents the dominant decisions and activities within CBA development as abstracted from our observations and analysis of USC e-services and CSE-affiliate projects.

The CBA process is undertaken by “walking” a path from “start” to “Non-CBA Activities” that connects (via arrows) activities as indicated by boxes and decisions that are indicated by ovals. Activities result in information that is passed on as input to either another activity or used to make a decision. Information follows the path that best describes the activity or decision output. Only one labeled path may be taken at any given time for any particular walk; however it is possible to perform multiple activities simultaneously (e.g. developing custom application code and glue code, multiple developers assessing or tailoring).

The small circles with letters A, T, G, C indicate the assessment, tailoring, glue code, and custom code development process elements respectively. With the exception of the latter, each of these areas will be expanded and elaborated in the sections that follow. These areas can generate the development activity sequences indicated in Table 1 by noting the order that these process elements are visited. Each area may enter and exit in numerous ways both from within the area itself or by following the decision framework of Figure 3.1. In addition, this scheme was developed from and is consistent with the CBA activity distributions.
COTS assessment aims at helping to make buy-or-build choices and helping select the most satisfactory combination of COTS products from various candidates.

It not only identifies the key tasks, but also emphasize the concurrency and high coupling of how COTS assessment fits with tailoring, glue code, and the overall process.
SoSE/Multiple COTS Synchronization Challenges
One of the USC e-services COTS-based applications involved the development of a viewing capability for oversized images. The original client needed a system to support viewing of digitized collections of old historical newspapers, but other users became interested in the capability for dealing with maps, artworks and other large digitized images. The full system capability included not just image navigation and zoom-in/zoom-out; but image catalog and metadata storage, update, search, and browse; image archive management; and access administration capabilities.

Several COTS products were available for the image processing functions, each with its strengths and weaknesses. None could cover the full system capability, although other COTS capabilities were available for some of these.
The original client was a USC librarian whose collections included access to some recently-digitized newspapers covering the early history of Los Angeles. Her main problem was that the newspapers were too large to fit on mainstream computer screens. She was aware that some COTS products were available to do this. She wanted the student developer team to identify the best COTS product to use, and to integrate it into a service for accessing the newspapers' content, covering the full system capability.

Her manager, who served as the customer, had two top-priority system constraints as her primary win conditions. One was to keep the cost of the COTS product below $25K. The other was to get reasonably mature COTS products with at least 5 existing supported customers.

The team quickly used these top-priority constraints to filter out two COTS products: system XYZ was too expensive, and system ABC had only one beta-test customer. The other two OIV COTS products, ER Mapper and Mr. SID, had different user interfaces; the major risk was to select one that users would subsequently find unacceptable. This risk was addressed by exercising the two products;
Entrance:

The first three steps establish the preconditions for entering the CBA Assessment decision framework such as top-level evaluation criteria, weights, and scenarios; candidate COTS products.

Spiral step 4 establishes the entry into Assessment process element.

Assessment Process Element

Detailed assessment involved Tailoring to accommodate the newspaper image files, and identified ER Mapper as the best OIV solution.

Exit:

In the “Partial COTS solution best” direction.
Following the Assessment Framework

the initial filtering step eliminated some candidates (XYZ and ABC), but not ER Mapper or Mr. SID. The risk assessment in Table 3 required the two COTS products to be exercised, which involved Tailoring to accommodate the newspaper image files, but not glue code at this point. The evaluation identified ER Mapper as the best OIV solution, but only as a partial solution for other needed functions such as cataloguing, search, and archiving.

exits back to the overall CBA decision Framework (Figure 3.1) in the “Partial COTS solution best” direction. But it cannot proceed further until the Win Win Spiral process determines whether either applications code or added COTS products or both need to be developed for the rest of the application.
Go-back:

The new stakeholders and OC&P’s in cycle 2 required the project to backtrack to the beginning of the Assessment process element.

Assessment Process Element

ER Mapper was filtered out when it declined to guarantee early Unix and Mac versions.

Tailoring Process Element

Tailoring was required to verify that Mr. SID performed satisfactorily on Unix and Mac platforms.

Assessment Process Element

Concurrently, Assessment filtering and evaluation tasks were being performed for the cataloguing and GUI functions.

This concurrency is a necessary attribute of most current and future CBA processes. Simple deterministic process representations are simply inadequate to address the dynamism, time-criticality, and varying risk/opportunity patterns of such CBA’s. However, the Win Win spiral process provides a workable framework for dealing with risk-driven concurrency, and the composable CBA decision framework and process elements provide workable approaches for handling the associated CBA activities.
Assessment Process Element:

The Assessment of detailed interoperability characteristics of Mr. SID, MySQL, and the GUI software on the Windows, Unix, and Mac platforms.

The interoperability assessment involved:

Tailoring Process Element

Tailoring was required to verify that Mr. SID performed satisfactorily on Unix and Mac platforms.

Glue Code Process Element

To enable interoperability
Case 10: Families of systems are typically a set of systems that belong to a product line and can be easily used to customize a solution for a given need. This might be a suite of medical devices or a suite of applications to customer support. This is often the set of systems developed by a vendor that become the NDI components for CASE 7 above. Again, the rigor required for the SoS case is present here. However, in this situation, the family of systems is typically owned and evolved by a single organization/vendor and presents a case where the owning organization has much more control over the evolution of the components of the family of systems, thus possibly reducing some risks and allowing the ICSM process to be a little more streamlined.
ICSM and Brownfield Development

- Many process models are Greenfield-oriented
  - Requirements ➔ Design ➔ Develop ➔ Test ➔ Operate
- Failed Greenfield project example
  - Corporate central financial system
  - To replace spaghetti-code collection of COBOL programs
- Improved ICSM Brownfield approach
  - Concurrent new-system definition and legacy system re-engineering

Failed Greenfield Corporate Financial System

- Used waterfall approach
  - Gathered requirements
  - Chose best-fit ERP system
  - Provided remaining enhancements
- Needed to ensure continuity of service
  - Planned incremental phase-in of new services
- Failed due to inability to selectively phase out legacy services
  - Dropped after 2 failed tries at cost of $40M
Legacy Systems Patched, Highly Coupled Financial and Non-Financial Services

ICSMA Approach to Brownfield Engineering

- Understanding needs
  - Analysis of legacy system difficulties

- Envisioning opportunities
  - Concurrently decouple legacy financial and non-financial services, explore new system phase-in and architecture options

- System scoping and architecting
  - Extract legacy financial, non-financial services
  - Prioritize, plan for incremental financial services phase-in/out

- Feasibility evidence development
  - Successful examples of representative service extractions
  - Evidence of cost, schedule, performance feasibility
Result of Legacy Re-engineering

Legacy Business Services
- Contract Services
  - Contract Financial Services
    - Billing
    - Subcontract payments
    - ...
  - Contract Non-Financial Services
    - Deliverables management
    - Terms compliance
    - ...
- General Financial Services
  - Accounting
  - Budgeting
  - Earned value
  - Payroll
  -...
- General Non-Financial Services
  - Progress tracking
  - Change tracking
  - ...
- Project Financial Services
  - WBS
  - Expenditure categories
  - ...
- Project Non-Financial Services
  - Scheduling
  - Staffing
  - Reqs CM
  - ...

Frequently Asked Question

- **Q:** Having all that ICSM generality and then using the decision table to come back to a simple model seems like an overkill.
  - If my risk patterns are stable, can’t I just use the special case indicated by the decision table?

- **A:** Yes, you can and should – as long as your risk patterns stay stable. But as you encounter change, the ICSM helps you adapt to it.
  - And it helps you collaborate with other organizations that may use different special cases.
Tutorial Outline

- Current DoD acquisition challenges
- Overview of ICSM
- Risk-based balance of agility and assurance
- ICSM process decision table
- Guidance and examples for using the ICSM
- ICSM tailoring exercise

ICSM Tailoring Exercise:
SupplyChain.com agent-based system

1. Analyze project and determine key decision drivers
2. Based on key decision drivers, identify candidate ICSM special case for development
3. Evaluate environment, agile, and plan-driven risks, and associated risk mitigation strategies
SupplyChain.com Profile

- Turnkey agent-based supply chain management systems
- Distributed, multi-organization teams of about 50 mostly-experienced people
- Parts of applications are relatively stable, while others are highly volatile
- Architectures are driven by a few key COTS packages that are also continually evolving

The ICSM Process Decision Table:
Key Decision Inputs

- Product and project size and complexity
- Requirements volatility
- Mission criticality
- Nature of Non-Developmental/COTS item support
  - Commercial, open-source, reused components
- Organizational and Personnel Capability
Common Risk-Driven Special Cases of the ICSM (Cases 1-4)

Case 1: Use NDI
- Example: Small accounting system
- Size: Complexity: Small
- Business drivers: N/A
- Criticality: Low
- NDI Support: Good
- Organizational Personal Capability: Medium
- Key SIs: Activity: Document Impact and Significance
- Time to Build: None
- Time to Implement: None

Case 2: Agile
- Example: Exerciser
- Size: Complexity: Low
- Criticality: Medium
- NDI Support: Good
- Organizational Personal Capability: Medium
- Key SIs: Activity: Organizational Readiness
- Time to Build: None
- Time to Implement: None

Case 3: Architected Agile
- Example: Business data processing
- Size: Complexity: Medium
- Criticality: Medium
- NDI Support: Good
- Organizational Personal Capability: Medium
- Key SIs: Activity: Architecture Design
- Time to Build: 2 weeks
- Time to Implement: 2 months

Case 4: Formal Methods
- Example: Security breach
- Size: Complexity: Low
- Criticality: Medium
- NDI Support: Good
- Organizational Personal Capability: Medium
- Key SIs: Activity: Formal Methodology
- Time to Build: 1 day
- Time to Implement: 1 day

Risks Used in Risk Comparison

- Environmental risks
  - Technology uncertainties
  - Many stakeholders
  - Complex system-of-systems
  - Legacy compatibility

- Agility risks
  - Scalability
  - Use of simple design
  - Personnel turnover
  - Too-frequent releases
  - Not enough agile-capable people

- Plan-driven risks
  - Rapid change
  - Need for rapid results
  - Emergent requirements
  - Not enough discipline-capable people
SupplyChain.com Home Ground Chart

Figure 5-2  Home Ground Chart for SupplyChain.com

SupplyChain.com Risks

<table>
<thead>
<tr>
<th>Risk Items</th>
<th>Risk Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environmental risks</td>
<td></td>
</tr>
<tr>
<td>- Risk: Uncertainty</td>
<td></td>
</tr>
<tr>
<td>- Goal: Many stakeholders</td>
<td></td>
</tr>
<tr>
<td>- Complex: Complex system of systems</td>
<td></td>
</tr>
<tr>
<td>Risks of using agile methods</td>
<td></td>
</tr>
<tr>
<td>- Scrum: Scalability and criticality</td>
<td></td>
</tr>
<tr>
<td>- TNG: Use of single design</td>
<td></td>
</tr>
<tr>
<td>- Lean: Personnel turnover</td>
<td></td>
</tr>
<tr>
<td>- Agile: Not enough people skilled in agile methods</td>
<td></td>
</tr>
<tr>
<td>Risks of using plan-driven methods</td>
<td></td>
</tr>
<tr>
<td>- P-Change: Rapid change</td>
<td></td>
</tr>
<tr>
<td>- P-Speed: Need for capital results</td>
<td></td>
</tr>
<tr>
<td>- P-Start: Emergent requirements</td>
<td></td>
</tr>
<tr>
<td>- P-End: Not enough people skilled in plan-driven methods</td>
<td></td>
</tr>
</tbody>
</table>

Risk rating scale:
- = Serious but manageable risk
- = Moderate risk
- = Minimal risk
- = Very serious but manageable risk
- = Showstoper risk

Figure 5-3  Risk Exposure (RE) Profiler: Planning Detail

Time and Effort Involved in Plans

- = Risk exposure due to inadequate plan
- = Risk exposure due to market share erosion
- = Sum of risk exposures

October 2010  Copyright © USC-CESE  76
ICSM Summary

- Current acquisition processes not well suited 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
  - Can be adopted incrementally
- Major implications for funding, contracting, career paths

Implications for funding, contracting, career paths

- Incremental vs. total funding
  - Often with evidence-based competitive downselect
- 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
ICSM Transition Paths

- Existing programs may benefit from some ICSM principles and practices, but not others
- Problem programs may find some ICSM practices helpful in recovering viability
- Primary opportunities for incremental adoption of ICSM principles and practices
  - Supplementing traditional requirements and design reviews with development and review of feasibility evidence
  - Stabilized incremental development and concurrent architecture rebaselining
  - Using schedule as independent variable and prioritizing features to be delivered
  - Continuous verification and validation
  - Using the process decision table
- For additional ICSM information, see http://csse.usc.edu (Tech Report 2009-500)

References - I

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

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

List of Acronyms

BL  Baselined
C4ISR  Command, Control, Computing, Communications, Intelligence, Surveillance, Reconnaissance
CD  Concept Development
CDR  Critical Design Review
COTS  Commercial Off-the-Shelf
DCR  Development Commitment Review
DI  Development Increment
DoD  Department of Defense
EGR  Exploration Commitment Review
EVMS  Earned Value Management System
FCR  Foundations Commitment Review
FED  Feasibility Evidence Description
FMEA  Failure Modes and Effects Analysis
FRP  Full-Rate Production
GAO  Government Accountability Office
GUI  Graphical User Interface
### List of Acronyms (continued)

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMI</td>
<td>Human-Machine Interface</td>
</tr>
<tr>
<td>HSI</td>
<td>Human-System Interface</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>ICSM</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>IS&amp;SE</td>
<td>Integrating Systems and Software Engineering</td>
</tr>
<tr>
<td>LCO</td>
<td>Life Cycle Objectives</td>
</tr>
<tr>
<td>LRIP</td>
<td>Low-Rate Initial Production</td>
</tr>
<tr>
<td>MBASE</td>
<td>Model-Based Architecting and Software Engineering</td>
</tr>
<tr>
<td>NDI</td>
<td>Non-Developmental Item</td>
</tr>
<tr>
<td>NRC</td>
<td>National Research Council</td>
</tr>
<tr>
<td>OC</td>
<td>Operational Capability</td>
</tr>
<tr>
<td>OCR</td>
<td>Operations Commitment Review</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>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>PM</td>
<td>Program Manager</td>
</tr>
<tr>
<td>PR</td>
<td>Public Relations</td>
</tr>
<tr>
<td>PRR</td>
<td>Product Release Review</td>
</tr>
<tr>
<td>RUP</td>
<td>Rational Unified Process</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>SSE</td>
<td>Systems and Software Engineering</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SwE</td>
<td>Software Engineering</td>
</tr>
<tr>
<td>SysE</td>
<td>Systems Engineering</td>
</tr>
<tr>
<td>Sys Engr</td>
<td>Systems Engineer</td>
</tr>
<tr>
<td>S&amp;SE</td>
<td>Systems and Software Engineering</td>
</tr>
<tr>
<td>USD (AT&amp;L)</td>
<td>Under Secretary of Defense for Acquisition, Technology, and Logistics</td>
</tr>
<tr>
<td>VCR</td>
<td>Validation Commitment Review</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>Verification and Validation</td>
</tr>
<tr>
<td>WBS</td>
<td>Work Breakdown Structure</td>
</tr>
<tr>
<td>WMI</td>
<td>Warfighter-Machine Interface</td>
</tr>
</tbody>
</table>