A Process Decision Table for Integrated Systems and Software Engineering

Barry Boehm and Jo Ann Lane, USC-CSSE
October 2008
Incremental Commitment Model (ICM): 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 DoD-related issues
  - Clarify “spiral development” usage in DoD Instruction 5000.2
    - Initial phased version (2005)
  - Explain Future Combat System of systems spiral usage to GAO
    - Underlying process principles (2006)
  - Provide framework for human-systems integration
- Integrates strengths of current process models
  - But not their weaknesses
ICM integrates strengths of current process models
But not their weaknesses

- V-Model: Emphasis on early verification and validation
  - But not ease of sequential, single-increment interpretation
- Spiral Model: Risk-driven activity prioritization
  - But not lack of well-defined in-process milestones
- RUP and MBASE: Concurrent engineering stabilized by anchor point milestones
  - But not software orientation
- Lean Development: Emphasis on value-adding activities
  - But not repeatable manufacturing orientation
- Agile Methods: Adaptability to unexpected change
  - But not software orientation, lack of scalability
Process Model Principles

Principles trump diagrams

1. Commitment and accountability
2. Success-critical stakeholder satisficing
3. Incremental growth of system definition and stakeholder commitment
4, 5. Concurrent, iterative system definition and development cycles
   Cycles can be viewed as sequential concurrently-performed phases or spiral growth of system definition
6. Risk-based activity levels and anchor point commitment milestones

Used by 60-80% of CrossTalk Top-5 projects, 2002-2005
# Common Risk-Driven Special Cases of the ICM

<table>
<thead>
<tr>
<th>Special Case</th>
<th>Example</th>
<th>Size, Complexity</th>
<th>Change Rate % /Month</th>
<th>Criticality</th>
<th>NDI Support</th>
<th>Org, Personnel Capability</th>
<th>Key Stage I Activities: Incremental Definition</th>
<th>Key Stage II Activities: Incremental Development, Operations</th>
<th>Time per Build; per Increment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Use NDI</td>
<td>Small Accounting</td>
<td>Complete</td>
<td></td>
<td></td>
<td>Acquire NDI</td>
<td>Use NDI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Agile</td>
<td>E-services</td>
<td>Low</td>
<td>1 – 30</td>
<td>Low-Med</td>
<td>Good; in place</td>
<td>Agile-ready Med-high</td>
<td>Skip Valuation, Architecting phases</td>
<td>Scrum plus agile methods of choice</td>
<td>&lt;= 1 day; 2-6 weeks</td>
</tr>
<tr>
<td>4. Formal Methods</td>
<td>Security kernel; Safety-critical LSI chip</td>
<td>Low</td>
<td>0.3</td>
<td>Extra High</td>
<td>None</td>
<td>Strong formal methods experience</td>
<td>Precise formal specification</td>
<td>Formally-based programming language; formal verification</td>
<td>1-5 days; 1-4 weeks</td>
</tr>
<tr>
<td>5. HW component with embedded SW</td>
<td>Multi-sensor control device</td>
<td>Low</td>
<td>0.3 – 1</td>
<td>Med-Very High</td>
<td>Good; In place</td>
<td>Experienced; med-high</td>
<td>Concurrent HW/SW engineering. CDR-level ICM DCR</td>
<td>IOC Development, LRIP, FRP, Concurrent Version N+1 engineering</td>
<td>SW: 1-5 days; Market-driven</td>
</tr>
<tr>
<td>6. Indivisible IOC</td>
<td>Complete vehicle platform</td>
<td>Med – High</td>
<td>0.3 – 1</td>
<td>High-Very High</td>
<td>Some in place</td>
<td>Experienced; med-high</td>
<td>Determine minimum-IoC likely, conservative cost. Add deferrable SW features as risk reserve</td>
<td>Drop deferrable features to meet conservative cost. Strong award fee for features not dropped</td>
<td>SW: 2-6 weeks; Platform: 6-18 months</td>
</tr>
<tr>
<td>7. NDI- Intensive</td>
<td>Supply Chain Management</td>
<td>Med – High</td>
<td>0.3 – 3</td>
<td>Med-Very High</td>
<td>NDI-driven architecture</td>
<td>NDI-experienced; Med-high</td>
<td>Thorough NDI-suite life cycle cost-benefit analysis, selection, concurrent requirements/ architecture definition</td>
<td>Pro-active NDI evolution influencing, NDI upgrade synchronization</td>
<td>SW: 1-4 weeks; System: 6-18 months</td>
</tr>
<tr>
<td>8. Hybrid agile / plan-driven system</td>
<td>C4ISR</td>
<td>Med – Very High</td>
<td>Mixed parts: 1 – 10</td>
<td>Mixed parts; Med-Very High</td>
<td>Mixed parts</td>
<td>Mixed parts</td>
<td>Full ICM; encapsulated agile in high change, low-medium criticality parts (Often HMI, external interfaces)</td>
<td>Full ICM, three-team incremental development, concurrent V&amp;V, next-increment rebaselining</td>
<td>1-2 months; 9-18 months</td>
</tr>
<tr>
<td>9. Multi-owner system of systems</td>
<td>Net-centric military operations</td>
<td>Very High</td>
<td>Mixed parts: 1 – 10</td>
<td>Very High</td>
<td>Many NDIs; some in place</td>
<td>Related experience, med-high</td>
<td>Full ICM; extensive multi-owner team building, negotiation</td>
<td>Full ICM; large ongoing system/software engineering effort</td>
<td>2-4 months; 18-24 months</td>
</tr>
<tr>
<td>10. Family of systems</td>
<td>Medical Device Product Line</td>
<td>Med – Very High</td>
<td>1 – 3</td>
<td>Med – Very High</td>
<td>Some in place</td>
<td>Related experience, med – high</td>
<td>Full ICM; Full stakeholder participation in product line scope, Strong business case</td>
<td>Full ICM. Extra resources for first system, version control, multi-stakeholder support</td>
<td>1-2 months; 9-18 months</td>
</tr>
</tbody>
</table>

**C4ISR**: Command, Control, Computing, Communications, Intelligence, Surveillance, Reconnaissance. **CDR**: Critical Design Review. **DCR**: Development Commitment Review. **FRP**: Full-Rate Production. **HMI**: Human-Machine Interface. **HW**: Hardware. **IOC**: Initial Operational Capability. **LRIP**: Low-Rate Initial Production. **NDI**: Non-Development Item. **SW**: Software
Case 1: Use NDI

• Exploration phase identifies NDI opportunities
• NDI risk/opportunity analysis indicates risks acceptable
  – Product growth envelope fits within NDI capability
  – Compatible NDI and product evolution paths
  – Acceptable NDI volatility, some open-source components highly volatile
  – Acceptable usability, dependability, interoperability
  – NDI available or affordable
• Example: Small accounting system
• Size/complexity: Low
• Anticipated change rate (% per month): Low
• Criticality: Low
• NDI support: Complete
• Organization and personnel capability: NDI-experienced
• Key Stage I activities: Acquire NDI
• Key State II activities: Use NDI
• Time/build: Driven by time to initialize/tailor NDI
• Time/increment: Driven by NDI upgrades
Case 2: Pure Agile Methods

• Exploration phase determines
  – Low product and project size and complexity
  – Fixing increment defects in next increment acceptable
  – Existing hardware and NDI support of growth envelope
  – Sufficient agile-capable personnel
  – Need to accommodate rapid change, emergent requirements, early user capability

• Example: E-services

• Size/complexity: Low

• Anticipated change rate (% per month): 1-30%

• Criticality: Low to medium

• NDI support: Good; in place

• Organization and personnel capability: Agile-ready, medium to high capability

• Key Stage I activities: Skip Valuation and Architecting phases

• Key State II activities: Scrum plus agile methods of choice

• Time/build: Daily

• Time/increment: 2-6 weeks
Case 3: Architected Agile

- Exploration phase determines
  - Need to accommodate fairly rapid change, emergent requirements, early user capability
  - Low risk of scalability up to 100 people
  - NDI support of growth envelope
  - Nucleus of highly agile-capable personnel
  - Moderate to high loss due to increment defects

- Example: Business data processing
- Size/complexity: Medium
- Anticipated change rate (% per month): 1-10%
- Criticality: Medium to high
- NDI support: Good, most in place
- Organization and personnel capability: Agile-ready, med-high capability
- Key Stage I activities: Combined Valuation and Architecting phase, complete NDI preparation
- Key Stage II activities: Architecture-based scrum of scrums
- Time/build: 2-4 weeks Time/increment: 2-6 months
Case 4: Formal Methods

- Biggest risks: Software/hardware does not accurately implement required algorithm precision, security, safety mechanisms, or critical timing
- Example: Security kernel or safety-critical LSI chip
- Size/complexity: Low
- Anticipated change rate (% per month): 0.3%
- Criticality: Extra high
- NDI support: None
- Organization and personnel capability: Strong formal methods experience
- Key Stage I activities: Precise formal specification
- Key State II activities: Formally-based programming language; formal verification
- Time/build: 1-5 days
- Time/increment: 1-4 weeks
Case 5: Hardware Component with Embedded Software

- Biggest risks: Device recall, lawsuits, production line rework, hardware-software integration
  - DCR carried to Critical Design Review level
  - Concurrent hardware-software design
    - Criticality makes Agile too risky
  - Continuous hardware-software integration
    - Initially with simulated hardware
- Low risk of overrun
  - Low complexity, stable requirements and NDI
  - Little need for risk reserve
  - Likely single-supplier software
Case 5: Hardware Component with Embedded Software (continued)

- Example: Multi-sensor control device
- Size/complexity: Low
- Anticipated change rate (% per month): 0.3-1%
- Criticality: Medium to very high
- NDI support: Good, in place
- Organization and personnel capability: Experienced; medium to high capability
- Key Stage I activities: Concurrent hardware and software engineering; CDR-level ICM DCR
- Key State II activities: IOC Development, LRIP, FRP, concurrent version N+1 engineering
- Time/build: 1-5 days (software)
- Time/increment: Market-driven
Case 6: Indivisible IOC

- Biggest risk: Complexity, NDI uncertainties cause cost-schedule overrun
  - Similar strategies to case 4 for criticality (CDR, concurrent HW-SW design, continuous integration)
  - Add deferrable software features as risk reserve
    - Adopt conservative (90% sure) cost and schedule
    - Drop software features to meet cost and schedule
    - Strong award fee for features not dropped
  - Likely multiple-supplier software makes longer (multi-weekly) builds more necessary
Case 6: Indivisible IOC (continued)

- Example: Complete vehicle platform
- Size/complexity: Medium to high
- Anticipated change rate (% per month): 0.3-1%
- Criticality: High to very high
- NDI support: Some in place
- Organization and personnel capability: Experienced, medium to high capability
- Key Stage I activities: Determine minimum-IOC likely, conservative cost; Add deferrable software features as risk reserve
- Key State II activities: Drop deferrable features to meet conservative cost; Strong award fee for features not dropped
- Time/build: 2-6 weeks (software)
- Time/increment: 6-18 months (platform)
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
- Organization 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 State II activities: Pro-active NDI evolution influencing, NDI upgrade synchronization
- Time/build: 1-4 weeks (software)
- Time/increment: 6-18 months (systems)
Case 8: Hybrid Agile/Plan-Driven System

- Biggest risks: large scale, high complexity, rapid change, mixed high/low criticality, partial NDI support, mixed personnel capability
- Example: C4ISR system
- Size/complexity: Medium to very high
- Anticipated change rate (% per month): Mixed parts; 1-10%
- Criticality: Mixed parts; medium to very high
- NDI support: Mixed parts
- Organization and personnel capability: Mixed parts
- Key Stage I activities: Full ICM; encapsulated agile in high changed; low-medium criticality parts (often HMI, external interfaces)
- Key State II activities: Full ICM, three-team incremental development, concurrent V&V, next-increment rebaselining
- Time/build: 1-2 months
- Time/increment: 9-18 months
Case 9: Multi-Owner System of Systems

- Biggest risks: all those of Case 8 plus
  - Need to synchronize, integrate separately-managed, independently-evolving systems
  - Extremely large-scale; deep supplier hierarchies
  - Rapid adaptation to change extremely difficult
- Example: Net-centric military operations
- Size/complexity: Very high
- Anticipated change rate (% per month): Mixed parts; 1-10%
- Criticality: Very high
- NDI support: Many NDIs; some in place
- Organization and personnel capability: Related experience, medium to high
- Key Stage I activities: Full ICM; extensive multi-owner teambuilding, negotiation
- Key State II activities: Full ICM; large ongoing system/software engineering effort
- Time/build: 2-4 months Time/increment: 18-24 months
Case 10: Family of Systems

- Biggest risks: all those of Case 8 plus
  - Need to synchronize, integrate separately-managed, independently-evolving systems
  - Extremely large-scale; deep supplier hierarchies
  - Rapid adaptation to change extremely difficult
- Example: Medical device product line
- Size/complexity: Medium to very high
- Anticipated change rate (% per month): 1-3%
- Criticality: Medium to very high
- NDI support: Some in place
- Organization and personnel capability: Related experience, medium to high capability
- Key Stage I activities: Full ICM; full stakeholder participation in product line scoping; strong business case
- Key State II activities: Full ICM; extra resources for first system, version control, multi-stakeholder support
- Time/build: 1-2 months   Time/increment: 9-18 months
Frequently Asked Question

Q: Having all that ICM 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 ICM helps you adapt to it.
   – And it helps you collaborate with other organizations that may use different special cases.