Agility in Defense SE & Acquisition: Some Critical Success Factors

Barry Boehm, USC
NDIA SE Conference
October 30, 2014
Summary

• Agile Defense SE & Acquisition and BBP 3.0
  – Better Buying Power

• Critical success factors vary by life cycle type
  – In-house agile life cycle
  – Outsourced agile development
  – Outsourced agile life cycle

• Other critical success factors
  – Avoiding a herd mentality
  – Process, product, people, project, and risk factors
  – Agility and safety
  – Avoiding one-size-fits-all process models
Agile SE & Acquisition and BBP 3.0

- Eliminate Unproductive Processes and Bureaucracy
  - Reduce cycle times while ensuring sound investments
  - Streamline documentation requirements and staff reviews

- Improve Tradecraft in Acquisition of Services
  - Strengthen contract management outside the normal acquisition chain

- Improve the Professionalization of the Total Acquisition Workforce
  - Strengthen organic engineering capabilities
Critical Success Factors: In-house agile life cycle

• Prepare for early success
  – Highly agile-qualified sponsors and performers
  – Top management buy-in
  – Involve all key organizations

• Choose project with enterprise-wide benefits
  – Based on evidence of critical success factors
  – Process, product, people, project, risk
    • These will indicate when not to go agile as well
  – Emphasize success and commitment to expand usage

• Incrementally expand with aid of initial project benefits
4 Potential Critical Success Factors
From SERC Expedited-SE study

Final Database
Over 30 Interviews with Gov’t/ Industry Rapid Development Organizations
Over 23,500 words from interview notes

People, Product, Process … all in a Project Context
Medical Case Study -- USA

• 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
Medical-USA 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
Key Practices – 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
CSFs: Outsourced Agile Development

• Prepare organization for life-cycle takeover
  – Highly capable in-house development performers, customers
    • AFR 63-123: Committed, representative, authorized, collaborative, knowledgeable

• Evidence-based source selection
  – Reference checking; software engineering exercise
  – Scalability to support full operational capability
  – Committed key personnel

• Significant award fee to developer
  – Flowdown to performers
    • Explicit criteria (Reifer-Boehm 2006); Vested Outsourcing (Vitasek 2011)
CSFs: Outsourced Agile Life Cycle

• Similar to agile development outsourcing
  – Highly capable customers
    • AFR 63-123: Committed, representative, authorized, collaborative, knowledgeable

• Evidence-based source selection
  – Reference checking; software engineering exercise
  – Scalability to support full operational capability
  – Committed key personnel

• Evidence-based release decision reviews
  – Acquirers part of agile rebaselining team

• Significant award fee to developer
  – Flowdown to performers
    • Explicit criteria (Reifer-Boehm 2006); Vested Outsourcing (Vitasek 2011)
Evolutionary Spiral Acquisition

- Rapid Change
  - Foreseeable Change (Plan)
  - Short Development Increments
- Increment N Baseline
  - Stable Development Increments
- High Assurance
  - Continuous V&V
  - Current V&V Resources
- Agile Rebaselining for Future Increments
  - Deferrals
  - Short, Stabilized Development of Increment N
    - Artifacts
    - Concerns
- Verification and Validation (V&V) of Increment N
  - Future V&V Resources
  - Future Increment Baselines
  - Increment N Transition/Operations and Maintenance
• Critical success factors vary by life cycle type
  – In-house agile life cycle
  – Outsourced agile development
  – Outsourced agile life cycle

Other critical success factors
  – Avoiding a herd mentality
  – Process, product, people, project, and risk factors
  – Agility and safety
  – Avoiding one-size-fits-all process models
Is Everybody Doing Agile?

1991), has come to encompass a range of development techniques now more commonly called agile or lean.

Research on agile/lean scheduling is particularly relevant now that agile/lean methods dominate the
Relative Size of Agile and Architecture Home Grounds

<table>
<thead>
<tr>
<th>Criticality, Size</th>
<th>Low (78%)</th>
<th>High (22%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Either</td>
<td>Arch</td>
</tr>
<tr>
<td>Low (80%)</td>
<td>Agile</td>
<td>Both</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Criticality, Size</th>
<th>Low (28%)</th>
<th>High (72%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High</td>
<td>Either</td>
<td>Architecture</td>
</tr>
<tr>
<td>Low (80%)</td>
<td>Agile</td>
<td>Both</td>
</tr>
</tbody>
</table>

Based on size distributions in financial sector (Highsmith 2002)
- 65% small (<10 people)
- 25% medium (11-50 people)
- 10% large (>50 people)
<table>
<thead>
<tr>
<th>Accelerators/Ratings</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Factors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simplicity</td>
<td>Extremely complex</td>
<td>Highly complex</td>
<td>Mod. complex</td>
<td>Moderately simple</td>
<td>Highly simple</td>
<td>Extremely simple</td>
</tr>
<tr>
<td>Element Reuse</td>
<td>None (0%)</td>
<td>Minimal (15%)</td>
<td>Some (30%)</td>
<td>Moderate (50%)</td>
<td>Considerate (70%)</td>
<td>Extensive (90%)</td>
</tr>
<tr>
<td>Low-Priority</td>
<td>Never</td>
<td>Rarely</td>
<td>Sometimes</td>
<td>Often</td>
<td>Usually</td>
<td>Anytime</td>
</tr>
<tr>
<td>Deferrals</td>
<td>None (0%)</td>
<td>Minimal (15%)</td>
<td>Some (30%)</td>
<td>Moderate (50%)</td>
<td>Considerate (70%)</td>
<td>Extensive (90%)</td>
</tr>
<tr>
<td>Key Technology</td>
<td>&gt;0 TRL 1-2 or 1 TRL 3</td>
<td>1 TRL 3 or &gt; 1 TRL 4</td>
<td>1 TRL 4 or &gt; 2 TRL 5</td>
<td>1-2 TRL 5 or &gt;2 TRL 6</td>
<td>1-2 TRL 6</td>
<td>All &gt; TRL 7</td>
</tr>
<tr>
<td>Maturity</td>
<td>Common core</td>
<td>Initial core</td>
<td>Intermediate</td>
<td>Advanced</td>
<td>Elite</td>
<td>Ultra-advanced</td>
</tr>
<tr>
<td><strong>Process Factors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concurrent</td>
<td>Highly sequential</td>
<td>Mostly sequential</td>
<td>2 artifacts mostly concurrent</td>
<td>All artifacts mostly concurrent</td>
<td>Fully concurrent</td>
<td>Fully concurrent</td>
</tr>
<tr>
<td>Operational Concept</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Requirements,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture, V&amp;V</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Streamlining</td>
<td>Heavily bureaucratic</td>
<td>Largely bureaucratic</td>
<td>Conservative bureaucratic</td>
<td>Moderate streamline</td>
<td>Mostly streamlined</td>
<td>Fully streamlined</td>
</tr>
<tr>
<td>General SE tool</td>
<td>Simple tools, weak integration</td>
<td>Minimal CIM</td>
<td>Some CIM</td>
<td>Moderate CIM</td>
<td>Considerable CIM</td>
<td>Extensive CIM</td>
</tr>
<tr>
<td>support CIM (Coverage, Integration, Maturity)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Project Factors</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project size (peak # of personnel)</td>
<td>Over 300</td>
<td>Over 100</td>
<td>Over 30</td>
<td>Over 10</td>
<td>Over 3</td>
<td>&lt; 3</td>
</tr>
<tr>
<td>Collaboration support</td>
<td>Globally distributed weak communication, data sharing</td>
<td>Nationally distributed, some sharing</td>
<td>Regionally distributed, moderate sharing</td>
<td>Metro-area distributed, good sharing</td>
<td>Simple campus, strong sharing</td>
<td>Largely collocated, very strong sharing</td>
</tr>
<tr>
<td>Single-domain</td>
<td>Simple MMPTs, weak integration</td>
<td>Minimal CIM</td>
<td>Some CIM</td>
<td>Moderate CIM</td>
<td>Considerable CIM</td>
<td>Extensive CIM</td>
</tr>
<tr>
<td>MMPTs (Models,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods, Processes,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-domain</td>
<td>Simple, weak integration</td>
<td>Minimal CIM</td>
<td>Some CIM or not needed</td>
<td>Moderate CIM</td>
<td>Considerable CIM</td>
<td>Extensive CIM</td>
</tr>
<tr>
<td>MMPTs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>People Factors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General SE KSAs</td>
<td>Weak KSAs</td>
<td>Some KSAs</td>
<td>Moderate KSAs</td>
<td>Good KSAs</td>
<td>Strong KSAs</td>
<td>Very strong KSAs</td>
</tr>
<tr>
<td>(Knowledge, Skills,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agility)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single-Domain KSAs</td>
<td>Weak</td>
<td>Some</td>
<td>Moderate</td>
<td>Good</td>
<td>Strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Multi-Domain KSAs</td>
<td>Weak</td>
<td>Some</td>
<td>Moderate or not needed</td>
<td>Good</td>
<td>Strong</td>
<td>Very strong</td>
</tr>
<tr>
<td>Risk Acceptance Factor</td>
<td>Highly risk-averse</td>
<td>Partly risk-averse</td>
<td>Balanced risk aversion, acceptance</td>
<td>Moderately risk-accepting</td>
<td>Considerably risk-accepting</td>
<td>Strongly risk-accepting</td>
</tr>
</tbody>
</table>
## CORADMO-SE Calibration Data
### Mostly Commercial; Some DoD

<table>
<thead>
<tr>
<th>Application Type</th>
<th>Technologies</th>
<th>Person Months</th>
<th>Duration (Months)</th>
<th>Duration / √PM</th>
<th>Product</th>
<th>Process</th>
<th>Project</th>
<th>People</th>
<th>Risk</th>
<th>Multiplier</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insurance agency system</td>
<td>HTML/VB</td>
<td>34.94</td>
<td>3.82</td>
<td>0.65</td>
<td>VII</td>
<td>VII</td>
<td>XII</td>
<td>VII</td>
<td>N</td>
<td>0.68</td>
<td>5%</td>
</tr>
<tr>
<td>Scientific/engineering</td>
<td>C++</td>
<td>18.66</td>
<td>3.72</td>
<td>0.86</td>
<td>L</td>
<td>VH</td>
<td>VH</td>
<td>VII</td>
<td>N</td>
<td>0.80</td>
<td>-7%</td>
</tr>
<tr>
<td>Compliance - expert</td>
<td>HTML/VB</td>
<td>17.89</td>
<td>3.36</td>
<td>0.79</td>
<td>VH</td>
<td>VH</td>
<td>XH</td>
<td>VH</td>
<td>N</td>
<td>0.68</td>
<td>-15%</td>
</tr>
<tr>
<td>Barter exchange</td>
<td>SQL/VB/HTML</td>
<td>112.58</td>
<td>9.54</td>
<td>0.90</td>
<td>VH</td>
<td>H</td>
<td>H</td>
<td>VH</td>
<td>N</td>
<td>0.75</td>
<td>-16%</td>
</tr>
<tr>
<td>Options exchange site</td>
<td>HTML/SQL</td>
<td>13.94</td>
<td>2.67</td>
<td>0.72</td>
<td>VII</td>
<td>VH</td>
<td>XH</td>
<td>VH</td>
<td>N</td>
<td>0.68</td>
<td>-5%</td>
</tr>
<tr>
<td>Commercial HMI</td>
<td>C++</td>
<td>205.27</td>
<td>13.81</td>
<td>0.96</td>
<td>L</td>
<td>N</td>
<td>N</td>
<td>VH</td>
<td>N</td>
<td>0.93</td>
<td>-3%</td>
</tr>
<tr>
<td>Options exchange site</td>
<td>HTML</td>
<td>42.41</td>
<td>4.48</td>
<td>0.69</td>
<td>VH</td>
<td>VH</td>
<td>XH</td>
<td>VH</td>
<td>N</td>
<td>0.68</td>
<td>-1%</td>
</tr>
<tr>
<td>Time and billing</td>
<td>C++/VB</td>
<td>26.87</td>
<td>4.80</td>
<td>0.93</td>
<td>L</td>
<td>VH</td>
<td>VH</td>
<td>VH</td>
<td>N</td>
<td>0.80</td>
<td>-14%</td>
</tr>
<tr>
<td>Hybrid Web/client-server</td>
<td>VB/HTML</td>
<td>70.93</td>
<td>8.62</td>
<td>1.02</td>
<td>L</td>
<td>N</td>
<td>VH</td>
<td>VH</td>
<td>N</td>
<td>0.87</td>
<td>-15%</td>
</tr>
<tr>
<td>ASP</td>
<td>HTML/VB/SQL</td>
<td>9.79</td>
<td>1.39</td>
<td>0.44</td>
<td>VII</td>
<td>VII</td>
<td>XII</td>
<td>VII</td>
<td>N</td>
<td>0.68</td>
<td>53%</td>
</tr>
<tr>
<td>On-line billing/tracking</td>
<td>VB/HTML</td>
<td>17.20</td>
<td>2.70</td>
<td>0.65</td>
<td>VII</td>
<td>VII</td>
<td>XII</td>
<td>VII</td>
<td>N</td>
<td>0.68</td>
<td>4%</td>
</tr>
<tr>
<td>Palm email client</td>
<td>C/HTML</td>
<td>4.53</td>
<td>1.45</td>
<td>0.68</td>
<td>N</td>
<td>VH</td>
<td>VH</td>
<td>VH</td>
<td>N</td>
<td>0.76</td>
<td>12%</td>
</tr>
</tbody>
</table>
Agility and Safety Failure Examples

• Responding to change over following a plan
  – Source of software-induced rocket failures

• Easiest-first features; safety as deferrable feature
  – Can’t make unsafe code safe via refactoring

• Nominal-case test-first
  – Fixing defects in next release

• Safety novices
  – No knowledge of hazard analysis, fault tree analysis, failure modes and effects analysis; redundancy and recovery techniques; COTS safety risk analysis
Agility and Safety: Under What Conditions?

- **Praxis-level development teams**
  - Safety-savvy; thorough; collaborative; change-adaptive

- **Architected agile process**
  - With validated feasibility evidence, change control

- **Providing adequate budget and schedule**
  - For safe development practices, V&V

- **Avoiding unsafe agile practices**
  - Collective code ownership; change trumps following plans; simple design; next-increment defect fixing; safety as deferrable feature

- **Using safety-enhancing agile practices**
  - Pair development; coding standards; safety-oriented test-first; continuous integration
Summary

• Critical success factors vary by life cycle type
  – In-house agile life cycle
  – Outsourced agile development
  – Outsourced agile life cycle

• Other critical success factors
  – Avoiding a herd mentality
  – Process, product, people, project, and risk factors
  – Agility and safety
  Avoiding one-size-fits-all process models
The Procrustean Bed: Dangers of one-size fits-all process models

- Procrustes: Greek Mythology
  - Rogue smith and bandit
  - Hostel with one-size-fits-all bed
  - Guests too small: stretch them to fit
  - Guests too large: lop off the offending parts
Build Your Own Procrustean Bed

• Pure Waterfall, Vee: Fixed Price and Spec Contract
  – Lop off needed changes as requirements creep
• Pure Agile: Easiest First; Dedicated On-Site Customer
  – Later scalability and assurance problems; single-failure point
• Voice of the Customer: Accept All “Requirements”
  – Gold-plating; neglect voices of acquirer, developer, owner
• Piling on Incompatible Constraints: No Way Out
  – Project Example: Waterfall, COTS, Ada, GOTS Reuse
• Inflexible Standards: No Choice But Tailoring Down
  – MIL-STD-498: choice of 23, 6, or 1 DID denied
• Overconstrained Maturity Models: Excluding Expertise
  – Software CMM: Exclude software group from system rqts.
Procrustean Example: DoD Acquisition Process
• 4 Swim Lanes + 2 Hybrids

• **Model 1:** Hardware Intensive Program

• **Model 2:** Defense Unique SW-Intensive Program

• **Model 3:** Incrementally Fielded SW-Intensive Program

• **Model 4:** Accelerated Acquisition Program

• **Hybrid Program A** (Hardware Dominant)

• **Hybrid Program B** (Software Dominant)
The Incremental Commitment Spiral Model (ICSM)
## Case 1: Use NDI

**Example:** Small accounting system  
**Size, Complexity:** Size variable, complexity low  
**Typical Change Rate/Month:** Negligible  
**Criticality:** n/a  
**NDI Support:** Complete  
**Organizational Personnel Capability:** NDI-experienced (medium)  
**Key Stage I Activities (Incremental Definition):** Acquire NDI  
**Key Stage II Activities (Incremental Development/Operations):** Use NDI  
**Time/Build:** n/a  
**Time/Increment:** Vendor-driven

## Case 2: Agile

**Example:** E-services  
**Size, Complexity:** Low  
**Typical Change Rate/Month:** 1-30%  
**Criticality:** Low to medium  
**NDI Support:** Good, in place  
**Organizational Personnel Capability:** Agile-ready, medium-high experience  
**Key Stage I Activities (Incremental Definition):** Skip Valuation and Architecting phases  
**Key Stage II Activities (Incremental Development/Operations):** Scrum plus agile methods of choice  
**Time/Build:** <= 1 day  
**Time/Increment:** 2-6 weeks

## Case 3: Architected Agile

**Example:** Business data processing  
**Size, Complexity:** Medium  
**Typical Change Rate/Month:** 1-10%  
**Criticality:** Medium to high  
**NDI Support:** Good, most in place  
**Organizational Personnel Capability:** Agile-ready, medium to high experience  
**Key Stage I Activities (Incremental Definition):** Combine Valuation, Architecting phases. Complete NDI preparation.  
**Key Stage II Activities (Incremental Development/Operations):** Architecture-based Scrum of Scrums  
**Time/Build:** 2-4 weeks  
**Time/Increment:** 2-6 months

## Case 4: Formal Methods

**Example:** Security kernel; Safety-critical LSI chip  
**Size, Complexity:** Low  
**Typical Change Rate/Month:** 0.3%  
**Criticality:** Extra high  
**NDI Support:** None  
**Organizational Personnel Capability:** Strong formal methods experience  
**Key Stage I Activities (Incremental Definition):** Precise formal specification  
**Key Stage II Activities (Incremental Development/Operations):** Formally-based programming language; formal verification  
**Time/Build:** 1-5 days  
**Time/Increment:** 1-4 weeks
Conclusions

- Success-critical to achieve both agility and quality
- Hybrid architected-agile methods emerging
  - Incremental commitment framework
  - Early development, validation of scalable architecture
  - Concurrent engineering with synchronization milestones
  - Scrum plus organizational essentials
- Success stories emerging
  - Management commitment to objectives and strategy
    - With incremental feasibility checkpoints
  - Strong core team of technical and management leaders
  - Thorough preparation of organizations, people, infrastructure
    - Involvement, architecture, policies, practices, plans, training
    - Incentives to address both agility and quality
  - Continuous change monitoring and adaptation