Future Challenges for Software Data Collection and Analysis

Barry Boehm, USC-CSSE
PROMISE 2009 Keynote
May 18, 2009
Future Software Measurement Challenges

• Emergent requirements
  – Example: Virtual global collaboration support systems
  – Need to manage early concurrent engineering

• Rapid change
  – In competitive threats, technology, organizations, environment

• Net-centric systems of systems
  – Incomplete visibility and control of elements

• Model-driven, service-oriented, Brownfield systems
  – New phenomenology, counting rules

• Always-on, never-fail systems
  – Need to balance agility and discipline
Emergent Requirements
– Example: Virtual global collaboration support systems

• View sharing, navigation, modification; agenda control; access control
• Mix of synchronous and asynchronous participation
• No way to specify collaboration support requirements in advance
• Need greater investments in concurrent engineering
  – of needs, opportunities, requirements, solutions, plans, resources
The Broadening Early Cone of Uncertainty (CU)

- Need greater investments in narrowing CU
  - Mission, investment, legacy analysis
  - Competitive prototyping
  - Concurrent engineering
  - Associated estimation methods and management metrics

- Larger systems will often have subsystems with narrower CU’s
The Incremental Commitment Life Cycle Process: Overview

<table>
<thead>
<tr>
<th>General/DoD Milestones</th>
<th>ICM Lifecycle Phases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploration Commitment Review</td>
<td>Exploration</td>
</tr>
<tr>
<td>Valuation Commitment Review</td>
<td>Valuation</td>
</tr>
<tr>
<td>Foundations Commitment Review</td>
<td>Foundations</td>
</tr>
<tr>
<td>Development Commitment Review</td>
<td>Development</td>
</tr>
<tr>
<td>Operations Commitment Review</td>
<td>Operations</td>
</tr>
</tbody>
</table>

### Stage I: Definition
- **Anchor Point**
- **Milestones**
  - ECR
  - VCR/CD
  - FCR/A

### Stage II: Development and Operations
- **Risk patterns**
- **Synchronize, stabilize concurrency via FEDs**

#### Activities

<table>
<thead>
<tr>
<th>Concurrent risk-and-opportunity-driven growth of system understanding and definition</th>
<th>Initial scoping</th>
<th>Concept definition</th>
<th>System life-cycle architecture and ops concept</th>
<th>Increment 1 development</th>
<th>Increment 1 operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation of evidence of feasibility to proceed</td>
<td>Feasibility Evidence</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Stakeholder review and commitment</td>
<td>High, but addressable</td>
<td>Acceptable</td>
<td>Risk?</td>
<td>Risk?</td>
<td>Risk?</td>
</tr>
</tbody>
</table>

Adjust scope, priorities, or discontinue
# ICM HSI Levels of Activity for Complex Systems

<table>
<thead>
<tr>
<th>Activity category</th>
<th>Levels of activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Envisioning</td>
<td></td>
</tr>
<tr>
<td>opportunities</td>
<td></td>
</tr>
<tr>
<td>System scoping</td>
<td></td>
</tr>
<tr>
<td>Understanding</td>
<td></td>
</tr>
<tr>
<td>needs</td>
<td></td>
</tr>
<tr>
<td>Goals/objectives</td>
<td></td>
</tr>
<tr>
<td>Requirements</td>
<td></td>
</tr>
<tr>
<td>Architecting</td>
<td></td>
</tr>
<tr>
<td>a. system</td>
<td></td>
</tr>
<tr>
<td>b. human</td>
<td></td>
</tr>
<tr>
<td>c. hardware</td>
<td></td>
</tr>
<tr>
<td>d. software</td>
<td></td>
</tr>
<tr>
<td>Life-cycle planning</td>
<td></td>
</tr>
<tr>
<td>Feasibility</td>
<td></td>
</tr>
<tr>
<td>Evidence</td>
<td></td>
</tr>
<tr>
<td>Negotiating</td>
<td></td>
</tr>
<tr>
<td>commitments</td>
<td></td>
</tr>
<tr>
<td>Development</td>
<td>OC₁</td>
</tr>
<tr>
<td>and evolution</td>
<td>OC₂</td>
</tr>
<tr>
<td>Monitoring</td>
<td></td>
</tr>
<tr>
<td>and control</td>
<td></td>
</tr>
<tr>
<td>Operations</td>
<td>OC₁</td>
</tr>
<tr>
<td>and retirement</td>
<td>Legacy</td>
</tr>
<tr>
<td>Organizational</td>
<td></td>
</tr>
<tr>
<td>capability</td>
<td>OC₁</td>
</tr>
<tr>
<td>improvement</td>
<td>OC₂</td>
</tr>
</tbody>
</table>
Nature of FEDs and Anchor Point Milestones

- **Evidence** provided by developer and validated by independent experts that:
  - If the system is built to the specified architecture, it will
    - Satisfy the specified operational concept and requirements
      - Capability, interfaces, level of service, and evolution
    - 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

- **Shortfalls in evidence are uncertainties and risks**
  - Should be resolved or covered by risk management plans

- **Assessed in increasing detail at major anchor point milestones**
  - Serves as basis for stakeholders’ commitment to proceed
  - Serves to synchronize and stabilize concurrently engineered elements
  - Can be used to strengthen current schedule- or event-based reviews
Key Point: Need to Show Evidence

• Not just traceability matrices and PowerPoint charts
• Evidence can include results of
  – Prototypes: networks, robots, user interfaces, COTS interoperability
  – Benchmarks: performance, scalability, accuracy
  – Exercises: mission performance, interoperability, security
  – Models: cost, schedule, performance, reliability; tradeoffs
  – Simulations: mission scalability, performance, reliability
  – Early working versions: infrastructure, data fusion, legacy compatibility
  – Representative past projects
  – Combinations of the above
• Validated by independent experts
  – Realism of assumptions
  – Representativeness of scenarios
  – Thoroughness of analysis
  – Coverage of key off-nominal conditions
• Much more effort data, product data to collect and analyze
COSYSMO Operational Concept

# Requirements
- # Scenarios
- # Algorithms
- Volatility Factor

# Interfaces

# Scenarios

# Algorithms

Volatility Factor

- Application factors
  - 8 factors
- Team factors
  - 6 factors
- Schedule driver

Size

Drivers

Effort

Multipliers

Calibration

Effort

WBS guided by ISO/IEC 15288
<table>
<thead>
<tr>
<th>COSYSMO Application Factor Description</th>
<th>Identifier</th>
<th>Current Prod. Range</th>
<th>Suggested Prod. Range</th>
<th>LOW (V)</th>
<th>LOW (L)</th>
<th>NOM (N)</th>
<th>HIGH (H)</th>
<th>VHIGH (VH)</th>
<th>XHIGH (XH)</th>
<th>Rating Selected</th>
<th>Resulting Multiplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Understanding</td>
<td>RQMT</td>
<td>1.73</td>
<td>1.73</td>
<td>1.40</td>
<td>1.20</td>
<td>1.00</td>
<td>0.90</td>
<td>0.81</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>Architecture Complexity</td>
<td>ARCH</td>
<td>1.66</td>
<td>1.66</td>
<td>1.28</td>
<td>1.14</td>
<td>1.00</td>
<td>0.88</td>
<td>0.77</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>Level of Service (KPP) Requirements</td>
<td>LSVC</td>
<td>2.50</td>
<td>2.50</td>
<td>0.66</td>
<td>0.83</td>
<td>1.00</td>
<td>1.33</td>
<td>1.65</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>Migration Complexity</td>
<td>MIGR</td>
<td>1.50</td>
<td>1.50</td>
<td>****</td>
<td>****</td>
<td>1.00</td>
<td>1.25</td>
<td>1.50</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>No. and Diversity of Installations/Platforms</td>
<td>INST</td>
<td>1.50</td>
<td>1.50</td>
<td>****</td>
<td>****</td>
<td>1.00</td>
<td>1.25</td>
<td>1.50</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>No. of Recursive Levels in the Design</td>
<td>RECU</td>
<td>1.50</td>
<td>1.50</td>
<td>0.82</td>
<td>0.91</td>
<td>1.00</td>
<td>1.12</td>
<td>1.23</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>Documentation to Match Lifecycle Needs</td>
<td>DOCU</td>
<td>0.67</td>
<td>0.67</td>
<td>0.82</td>
<td>0.91</td>
<td>1.00</td>
<td>1.12</td>
<td>1.23</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
<tr>
<td>Technology Maturity</td>
<td>TMAT</td>
<td>2.50</td>
<td>2.50</td>
<td>1.75</td>
<td>1.37</td>
<td>1.00</td>
<td>0.85</td>
<td>0.70</td>
<td>****</td>
<td>N</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Productivity Range (PR) is the Highest Number / Lowest Number and is an indication of the "Relative Degree of Influence" of this parameter on SE effort as currently. The "Suggested" column has no immediate impact in the COSYSMO SE Costing Mode. However, for the COSYSMO SE Data Collection Mode, it serves as a means of collecting your inputs as to what you think the "Relative Degree of Influence" of this parameter should be based upon your overall experience (not specific to the past program being characterized). If you agree with the "Current" number, do nothing. If you disagree, simply overwrite the current number with a new number n (n>1.0) in the appropriate cell.

Select the Rating from the pull-down that best represents the Rating program being estimated in the Mode or in the SE Data Collection Rating that best characterizes the program for which you are providing a costing estimate.
Next-Generation Systems Challenges

• Emergent requirements
  – Example: Virtual global collaboration support systems
  – Need to manage early concurrent engineering

• Rapid change
  – In competitive threats, technology, organizations, environment

• Net-centric systems of systems
  – Incomplete visibility and control of elements

• Model-driven, service-oriented, Brownfield systems
  – New phenomenology, counting rules

• Always-on, never-fail systems
  – Need to balance agility and discipline
Rapid Change Creates a Late Cone of Uncertainty
– Need evolutionary/incremental vs. one-shot development
- - No simple boundary between development and maintenance

Uncertainties in competition, technology, organizations, mission priorities

Phases and Milestones

Feasibility
Plans and Rqts.
Product Design Spec.
Product Design
Detail Design Spec.
Detail Design
Devel. and Test
Accepted Software

Relative Cost Range
0.25x 0.5x 0.67x 0.8x 1.25x 1.5x 2x 4x

May 18, 2009
Incremental Development Productivity Decline (IDPD)

- **Example: Site Defense BMD Software**
  - 5 builds, 7 years, $100M
  - Build 1 productivity over 300 SLOC/person month
  - Build 5 productivity under 150 SLOC/PM
    - Including Build 1-4 breakage, integration, rework
    - 318% change in requirements across all builds
    - IDPD factor = 20% productivity decrease per build
  - Similar trends in later unprecedented systems
  - Not unique to DoD: key source of Windows Vista delays

- **Maintenance of full non-COTS SLOC, not ESLOC**
  - Build 1: 200 KSLOC new; 200K reused@20% = 240K ESLOC
  - Build 2: 400 KSLOC of Build 1 software to maintain, integrate
“Equivalent SLOC” Paradoxes

• Not a measure of software size
• Not a measure of software effort
• Not a measure of delivered software capability
• A quantity derived from software component sizes and reuse factors that helps estimate effort
• Once a product or increment is developed, its ESLOC loses its identity
  – Its size expands into full SLOC
  – Some people apply reuse factors to this to determine an ESLOC quantity for the next increment
    • But this has no relation to the product’s size
IDPD Cost Drivers: Conservative 4-Increment Example

- Some savings: more experienced personnel (5-20%)
  - Depending on personnel turnover rates
- Some increases: code base growth, diseconomies of scale, requirements volatility, user requests
  - Breakage, maintenance of full code base (20-40%)
  - Diseconomies of scale in development, integration (10-25%)
  - Requirements volatility; user requests (10-25%)
- Best case: 20% more effort (IDPD=6%)
- Worst case: 85% (IDPD=23%)
Effects of IDPD on Number of Increments

- Model relating productivity decline to number of builds needed to reach 8M SLOC Full Operational Capability
- Assumes Build 1 production of 2M SLOC @ 100 SLOC/PM
  - 20000 PM/ 24 mo. = 833 developers
  - Constant staff size for all builds
- Analysis varies the productivity decline per build
  - Extremely important to determine the incremental development productivity decline (IDPD) factor per build
# Choosing and Costing Incremental Development Forms

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Pros</th>
<th>Cons</th>
<th>Cost Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evolutionary</td>
<td>Small: Agile</td>
<td>Adaptability to change</td>
<td>Easiest-first; late, costly breakage</td>
<td>Small: Planning-poker-type</td>
</tr>
<tr>
<td>Sequential</td>
<td>Large: Evolutionary Development</td>
<td></td>
<td></td>
<td>Large: Parametric with IDPD</td>
</tr>
<tr>
<td>Prespecified</td>
<td>Platform base plus PPPIs</td>
<td>Prespecifiable full-capability requirements</td>
<td>Emergent requirements or rapid change</td>
<td>COINCOMO with no increment overlap</td>
</tr>
<tr>
<td>Sequential</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overlapped Evolutionary</td>
<td>Product lines with ultrafast change</td>
<td>Modular product line</td>
<td>Cross-increment breakage</td>
<td>Parametric with IDPD and Requirements Volatility</td>
</tr>
<tr>
<td>Rebaselining</td>
<td>Mainstream product lines; Systems of systems</td>
<td>High assurance with rapid change</td>
<td>Highly coupled systems with very rapid change</td>
<td>COINCOMO, IDPD for development; COSYSMO for rebaselining</td>
</tr>
<tr>
<td>Evolutionary</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IDPD:** Incremental Development Productivity Decline, due to earlier increments breakage, increasing code base to integrate

**PPPIs:** Pre-Planned Product Improvements

**COINCOMO:** COCOMO Incremental Development Model (COCOMO II book, Appendix B)

**COSYSMO:** Systems Engineering Cost Model (in-process COSYSMO book)

All Cost Estimation approaches also include expert-judgment cross-check.
Next-Generation Systems Challenges

• Emergent requirements
  – Example: Virtual global collaboration support systems
  – Need to manage early concurrent engineering

• Rapid change
  – In competitive threats, technology, organizations, environment

• Net-centric systems of systems
  – Incomplete visibility and control of elements

• Model-driven, service-oriented, Brownfield systems
  – New phenomenology, counting rules

• Always-on, never-fail systems
  – Need to balance agility and discipline
## Further Attributes of Future Challenges

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Pros</th>
<th>Cons</th>
<th>Cost Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems of Systems</td>
<td>• Directed: Future Combat Systems</td>
<td>• Interoperability</td>
<td>• Often-conflicting partner priorities</td>
<td>• Staged hybrid models</td>
</tr>
<tr>
<td></td>
<td>• Acknowledged: Missile Defense Agency</td>
<td>• Rapid Observe-Orient- Decide-Act (OODA) loop</td>
<td>• Change processing very complex</td>
<td>• Systems engineering: COSYSMO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Multi-organization development costing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Lead Systems integrator costing</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Requirements volatility effects</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Integration&amp;test: new cost drivers</td>
</tr>
<tr>
<td>Model-Driven Development</td>
<td>• Business 4th-generation languages (4GLs)</td>
<td>• Cost savings</td>
<td>• Multi-model composition incapabilities</td>
<td>• Models directives as 4GL source code</td>
</tr>
<tr>
<td></td>
<td>• Vehicle-model driven development</td>
<td>• User-development advantages</td>
<td>• Model extensions for special cases (platform-payload)</td>
<td>• Multi-model composition similar to COTS integration, Brownfield integration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Fewer error sources</td>
<td>• Brownfield complexities</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• User-development V&amp;V</td>
<td></td>
</tr>
<tr>
<td>Brownfield</td>
<td>• Legacy C4ISR System</td>
<td>• Continuity of service</td>
<td>• Legacy re-engineering often complex</td>
<td>• Models for legacy re-engineering, mega-refactoring</td>
</tr>
<tr>
<td></td>
<td>• Net-Centric weapons platform</td>
<td>• Modernization of infrastructure</td>
<td>• Mega-refactoring often complex</td>
<td>• Reuse model for refactored legacy</td>
</tr>
<tr>
<td></td>
<td>• Multicore-CPU upgrades</td>
<td>• Ease of maintenance</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

May 18, 2009

Copyright © USC-CSSE
## Further Attributes of Future Challenges (Continued)

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Pros</th>
<th>Cons</th>
<th>Cost Estimation</th>
</tr>
</thead>
</table>
| Ultrareliable Systems     | • Safety-critical systems  
• Security-critical systems  
• High-performance real-time systems | • System resilience, survivability  
• Service-oriented usage opportunities | • Conflicts among attribute objectives  
• Compatibility with rapid change      | • Cost model extensions for added assurance levels  
• Change impact analysis models |
| Competitive Prototyping   | • Stealth vehicle fly-offs  
• Agent-based RPV control  
• Combinations of challenges | • Risk buy-down  
• Innovation modification  
• In-depth exploration of alternatives | • Competitor evaluation often complex  
• Higher up-front cost  
• But generally good ROI  
• Tech-leveling avoidance often complex | • Competition preparation, management costing  
• Evaluation criteria, scenarios, testbeds  
• Competitor budget estimation  
• Virtual, proof-of-principle, robust prototypes |
Net-Centric Systems of Systems Challenges

• Need for rapid adaptation to change
  – See first, understand first, act first, finish decisively

• Built-in authority-responsibility mismatches
  – Increasing as authority decreases through Directed, Acknowledged, Collaborative, and Virtual SoS classes
    • Incompatible element management chains, legacy constraints, architectures, service priorities, data, operational controls, standards, change priorities...

• High priority on leadership skills, collaboration incentives, negotiation support such as cost models
  – SoS variety and complexity makes compositional cost models more helpful than one-size-fits-all models
Example: SoSE Synchronization Points

SoS-Level

Source Selection

Candidate Supplier/Strategic Partner n

Candidate Supplier/Strategic Partner 1

LCO-type Proposal & Feasibility Info

System x

System C

System B

System A

May 18, 2009

Copyright © USC-CSSE
Average Change Processing Time:
Two Complex Systems of Systems

Average workdays to process changes

Incompatible with turning within adversary’s OODA loop
### Compositional approaches: Directed systems of systems

<table>
<thead>
<tr>
<th></th>
<th>Inception</th>
<th>Elaboration</th>
<th>SoS Architecting</th>
<th>Increment 1</th>
<th>Increments 2,... n</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LCO</strong></td>
<td>Effort COSYSMO-like.</td>
<td>RFP, SOW, Evaluations, Contracting</td>
<td>Assess compatibility, shortfalls</td>
<td>Assess sources of change; Negotiate rebaselined LCA package at all levels</td>
<td>Similar, with added change traffic from users...</td>
</tr>
<tr>
<td></td>
<td>Schedule = Effort/Staff</td>
<td>Effort/Staff</td>
<td>Effort/staff at all levels</td>
<td>COSOSIMO-like</td>
<td>LCA-2</td>
</tr>
<tr>
<td></td>
<td>Try to model ideal staff size</td>
<td>COSOSIMO-like</td>
<td>COSOSIMO-like</td>
<td>COSOSIMO-like</td>
<td></td>
</tr>
<tr>
<td><strong>LCA</strong></td>
<td>Proposals</td>
<td>Develop to spec</td>
<td>CORADMO-like</td>
<td>Integrate COSOSIMO-like</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Degree of Completeness</td>
<td>Risk-manage slow-performer, completeness</td>
<td>risks, rework</td>
<td>LCA-2 shortfalls</td>
<td></td>
</tr>
<tr>
<td><strong>IOC</strong></td>
<td>Proposal Feasibility</td>
<td>Proposal Feasibility</td>
<td>Proposal Feasibility</td>
<td>Proposal Feasibility</td>
<td></td>
</tr>
</tbody>
</table>

**Suppliers – Agile**

- LSI IPTs – Agile
- Suppliers – Agile

**LSI – Agile**

- Customer, Users
- LSI – Agile

**Suppliers – Integrators**

- LSI – Integrators
- Suppliers – PD – V&V

---

May 18, 2009

Copyright © USC-CSSE
How Much Architecting is Enough?
- Larger projects need more

Sweet Spot Drivers:
Rapid Change: leftward
High Assurance: rightward
# Comparison of Cost Model Parameters

<table>
<thead>
<tr>
<th>Parameter Aspects</th>
<th>COSYSMO</th>
<th>COSOSIMMO</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size drivers</strong></td>
<td># of system requirements&lt;br&gt;# of system interfaces&lt;br&gt;# operational scenarios&lt;br&gt;# algorithms</td>
<td># of SoS requirements&lt;br&gt;# of SoS interface protocols&lt;br&gt;# of constituent systems&lt;br&gt;# of constituent system organizations&lt;br&gt;# operational scenarios</td>
</tr>
<tr>
<td><strong>“Product” characteristics</strong></td>
<td>Size/complexity&lt;br&gt;Requirements understanding&lt;br&gt;Architecture understanding&lt;br&gt;Level of service requirements&lt;br&gt;# of recursive levels in design&lt;br&gt;Migration complexity&lt;br&gt;Technology risk&lt;br&gt;#/ diversity of platforms/installations&lt;br&gt;Level of documentation</td>
<td>Size/complexity&lt;br&gt;Requirements understanding&lt;br&gt;Architecture understanding&lt;br&gt;Level of service requirements&lt;br&gt;Component system maturity and stability&lt;br&gt;Component system readiness</td>
</tr>
<tr>
<td><strong>Process characteristics</strong></td>
<td>Process capability&lt;br&gt;Multi-site coordination&lt;br&gt;Tool support</td>
<td>Maturity of processes&lt;br&gt;Tool support&lt;br&gt;Cost/schedule compatibility&lt;br&gt;SoS risk resolution</td>
</tr>
<tr>
<td><strong>People characteristics</strong></td>
<td>Stakeholder team cohesion&lt;br&gt;Personnel/team capability&lt;br&gt;Personnel experience/continuity</td>
<td>Stakeholder team cohesion&lt;br&gt;SoS team capability</td>
</tr>
</tbody>
</table>
Model-Driven, Service-Oriented, Brownfield Systems

New phenomenology, counting rules

• Product generation from model directives
  – Treat as very high level language: count directives

• Model reuse feasibility, multi-model incompatibilities
  – Use Feasibility Evidence progress tracking measures

• Functional vs. service-oriented architecture mismatches
  – Part-of (one-many) vs. served-by (many-many)

• Brownfield legacy constraints, reverse engineering
  – Reverse-engineer legacy code to fit new architecture
  – Elaborate COSYSMO Migration Complexity cost driver
  – Elaborate COCOMO II reuse model for reverse 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
ICM 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 mgmt.
  - Terms compliance

Project Services
- 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

...
Always-on, never-fail systems

- Consider using “weighted SLOC” as a productivity metric
- Some SLOC are “heavier to move into place” than others
  - And largely management uncontrollables
  - Examples: high values of COCOMO II cost drivers
    - RELY: Required Software Reliability
    - DATA: Database Size
    - CPLX: Software Complexity
    - DOCU: Required Documentation
    - RUSE: Required Development for Future Reuse
    - TIME: Execution Time Constraint
    - STOR: Main Storage Constraint
    - SCED: Required Schedule Compression

- Provides way to compare productivities across projects
  - And to develop profiles of project classes
COSECMO Estimation Trends
Effort by Assurance Levels for Different Size Projects

- Plot of projects where only SECU & effort increasing drivers
- Efforts seem a little low based on values from Orange Book projects
Balancing Agility and Assurance

- No one-size-fits-all estimation and metrics approach
  - Need compositional approach for both phases and components
- ICM decision table provides criteria for component processes, estimation methods, management metrics
  - Agile: Planning poker/ Wideband Delphi; story burndown
  - Architected agile: planning and implementation sprints: agile plus FED preparation estimation and progress monitoring
  - Mission platforms: hardware, software cost models plus FED preparation estimation and progress monitoring; Leading Indicators and Macro Risk Tool
  - Systems of Systems: composite estimation models; FED estimation and monitoring; extended Macro Risk Tool
### 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</td>
<td>Use NDI</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Complete</td>
<td>Acquire NDI</td>
<td>Use NDI</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Agile</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>3</td>
<td>Architected Agile</td>
<td>Med</td>
<td>1 – 10</td>
<td>Med-High</td>
<td>Good; most in place</td>
<td>Agile-ready Med-high</td>
<td>Combine Valuation, Architecting phases. Complete NDI preparation</td>
<td>Architecture-based Scrums of Scrums</td>
<td>2-4 weeks; 2-6 months</td>
</tr>
<tr>
<td>4</td>
<td>Formal Methods</td>
<td>Low</td>
<td>0.3 – 1</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</td>
<td>HW component with embedded SW</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</td>
<td>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>
</tr>
<tr>
<td>7</td>
<td>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; 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</td>
<td>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>
</tr>
<tr>
<td>9</td>
<td>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 NDI; 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>
</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: Hard ware.  
IOC: Initial Operational Capability.  
LRIP: Low-Rate Initial Production.  
NDI: Non-Development Item.  
SW: Software
Conclusions

• Future trends imply need to concurrently address new estimation and management metrics challenges
  – Emergent requirements, rapid change, net-centric systems of systems, MDD/SOA/Brownfield, ultrahigh assurance

• Need to work out cost drivers, estimating relationships for new phenomena
  – Incremental Development Productivity Decline (IDPD)
  – ESLOC and milestone definitions
  – Compositional approach for systems of systems
  – NDI, model, and service composability
  – Re-engineering, migration of legacy systems
  – Ultra-reliable systems development
  – Cost/schedule tradeoffs

• Need adaptive data collection & analysis feedback cycle
TRW/COCOMO II Experience Factory: II

System objectives: fcn’y, perf., quality
Corporate parameters: tools, processes, reuse

COCOMO II

Rescope

Cost, Sched, Risks

Yes

Milestone plans, resources

Milestone expectations

Ok?

No

Execute project to next Milestone

Revise Milestones, Plans, Resources

M/S Results

No

Revised Expectations

Ok?

Yes

Done?

No

End

Yes

May 18, 2009
TRW/COCOMO II Experience Factory: IV

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

Evaluate Corporate SW Improvement Strategies

Rescope

COCOMO II

Ok?

Cost, Sched, Risks

Ok?

Yes

No

Accumulate COCOMO II calibration data

Recalibrate COCOMO II

Milestone plans, resources

Milestone expectations

Yes

No

Executes project to next Milestone

Revise Milestones, Plans, Resources

M/S Results

No

Yes

Revised Expectations

Done?

End

Ok?
References


## List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Assessment and Assimilation</td>
</tr>
<tr>
<td>AAF</td>
<td>Adaptation Adjustment Factor</td>
</tr>
<tr>
<td>AAM</td>
<td>Adaptation Adjustment Modifier</td>
</tr>
<tr>
<td>COCOMO</td>
<td>Constructive Cost Model</td>
</tr>
<tr>
<td>COSOSIMO</td>
<td>Constructive System of Systems Integration Cost Model</td>
</tr>
<tr>
<td>COSYSMO</td>
<td>Constructive Systems Engineering Cost Model</td>
</tr>
<tr>
<td>COTS</td>
<td>Commercial Off-The-Shelf</td>
</tr>
<tr>
<td>CU</td>
<td>Cone of Uncertainty</td>
</tr>
<tr>
<td>DCR</td>
<td>Development Commitment Review</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>ECR</td>
<td>Exploration Commitment Review</td>
</tr>
<tr>
<td>ESLOC</td>
<td>Equivalent Source Lines of Code</td>
</tr>
<tr>
<td>EVMS</td>
<td>Earned Value Management System</td>
</tr>
<tr>
<td>FCR</td>
<td>Foundations Commitment Review</td>
</tr>
<tr>
<td>FDN</td>
<td>Foundations, as in FDN Package</td>
</tr>
<tr>
<td>FED</td>
<td>Feasibility Evidence Description</td>
</tr>
<tr>
<td>GD</td>
<td>General Dynamics</td>
</tr>
<tr>
<td>GOTS</td>
<td>Government Off-The-Shelf</td>
</tr>
</tbody>
</table>
List of Acronyms (continued)

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM</td>
<td>Incremental Commitment Model</td>
</tr>
<tr>
<td>IDPD</td>
<td>Incremental Development Productivity Decline</td>
</tr>
<tr>
<td>IOC</td>
<td>Initial Operational Capability</td>
</tr>
<tr>
<td>LCA</td>
<td>Life Cycle Architecture</td>
</tr>
<tr>
<td>LCO</td>
<td>Life Cycle Objectives</td>
</tr>
<tr>
<td>LMCO</td>
<td>Lockheed Martin Corporation</td>
</tr>
<tr>
<td>LSI</td>
<td>Lead System Integrator</td>
</tr>
<tr>
<td>MDA</td>
<td>Model-Driven Architecture</td>
</tr>
<tr>
<td>NDA</td>
<td>Non-Disclosure Agreement</td>
</tr>
<tr>
<td>NDI</td>
<td>Non-Developmental Item</td>
</tr>
<tr>
<td>NGC</td>
<td>Northrop Grumman Corporation</td>
</tr>
<tr>
<td>OC</td>
<td>Operational Capability</td>
</tr>
<tr>
<td>OCR</td>
<td>Operations Commitment Review</td>
</tr>
<tr>
<td>OO</td>
<td>Object-Oriented</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>
</tbody>
</table>
List of Acronyms (continued)

RFP Request for Proposal
SAIC Science Applications international Corporation
SLOC Source Lines of Code
SoS System of Systems
SoSE System of Systems Engineering
SRDR Software Resources Data Report
SSCM Systems and Software Cost Modeling
SU Software Understanding
SW Software
SwE Software Engineering
SysE Systems Engineering
Sys Engr Systems Engineer
S&SE Systems and Software Engineering
ToC Table of Contents
USD (AT&L) Under Secretary of Defense for Acquisition, Technology, and Logistics
VCR Validation Commitment Review
V&V Verification and Validation
WBS Work Breakdown Structure