Current and Future Challenges for Systems and Software Cost Estimation

Barry Boehm, USC-CSSE
29th COCOMO-SSCM Forum
October 21, 2014
Summary

Current and future trends create challenges for systems and software cost estimation

- Mission challenges: emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems

- USC, NPS/AFIT, DoD Systems Engineering Research Center researching ways to address challenges
  - Beginning with space systems (COSATMO models)
  - Extendable to other DoD domains

- Forum includes related COCOMO-family workshops
  - Wednesday AM: COSYSMO 3.0
  - Thursday AM: COCOMO III
Software Estimation: The Receding Horizon

IDPD: Incremental Development Productivity Decline
MBSSE: Model-Based Systems and Sw Engr.
COTS: Commercial Off-the-Shelf
SoS: Systems of Systems

Relative Productivity

Estimation Error

Unprecedented | Preceded | Component-based | COTS | Agile | SoS. Apps, Widgets, IDPD, Clouds, Security, MBSSE

Time, Domain Understanding
Current and Future Estimation Challenges

• Emergent requirements
  – Cannot prespecify requirements, cost, schedule, EVMS
  – Need to estimate and track early concurrent engineering

• Rapid change
  – Long acquisition cycles breed obsolescence
  – Need better models for incremental development

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

• Model, COTS, service-based, Brownfield systems
  – New phenomenology, counting rules

• Major concerns with affordability
  – US DoD: Better Buying Power 3.0
US DoD: Better Buying Power 3.0
Current draft about to become DoD policy

- Achieve affordable programs
- Achieve dominant capabilities while controlling lifecycle costs
- Incentivize productivity in industry and government
- Incentivize innovation in industry and government
- Eliminate unproductive processes and bureaucracy
- Promote effective competition
- Improve tradecraft in acquisition of services
- Improve the professionalism of the total acquisition workforce
Rapid Change Creates a Late Cone of Uncertainty
– Need evolutionary/incremental vs. one-shot development

Uncertainties in competition, technology, organizations, mission priorities
Incremental Development Productivity Decline (IDPD)

- **Example: Site Defense BMD Software**
  - 5 builds, 7 years, $100M; operational and support software
  - Build 1 productivity over 300 LOC/person month
  - Build 5 productivity under 150 LOC/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
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
Multi-Mission Systems Costing

• Product Line Engineering
  – Identify multi-mission commonalities and variabilities
  – Identify fully, partially sharable commonalities
  – Develop plug-compatible interfaces for variabilities

• Product Line Costing (COPLIMO) Parameters
  – Fractions of system fully reusable, partially reusable and cost of developing them for reuse
  – Fraction of system variabilities and cost of development
  – System lifetime and rates of change

• Product Line Life Cycle Challenges
  – Layered services vs. functional hierarchy
  – Modularization around sources of change
  – Version control, COTS refresh, and change prioritization
  – Balancing agility, assurance, and affordability
Risk-Driven Scalable Spiral Model: Increment View

Unforeseeable Change (Adapt)

Rapid Change

Agile Rebaselining for Future Increments

Deferrals

Short, Stabilized Development of Increment N

Increment N Transition/Operations and Maintenance

Future Increment Baselines

Future V&V Resources

Current V&V Resources

High Assurance

Verification and Validation (V&V) of Increment N

Unforeseeable Change (Plan)

Short Development Increments

Stable Development Increments

Foreseeable Change (Plan)

Short Development Increments

Continuous V&V
Summary

• Current and future trends create challenges for ground system cost estimation
  – Mission challenges: emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems

DoD Systems Engineering Research Center researching ways to address challenges
  – Beginning with space systems (COSATMO models)
  – Extendable to other DoD domains

• Workshop objectives
  – Understand, prioritize ground system cost estimation needs, opportunities
  – Identify sources of expertise, data
COSATMO Concept

- Focused on current and future satellite systems
  - Accommodating rapid change, evolutionary development, Net-Centric SoSs, Families of systems, DI2E SWASe’s
    - Software, Widgets, Assets, Services, etc.
  - Recognizes new draft DoDI 5000.02 process models
    - Hardware-intensive, DoD-unique SW-intensive, Incremental SW-intensive, Accelerated acquisition, 2 Hybrids (HW-, SW-dominant)
  - Supports affordability analyses (total cost of ownership):
    - Covers full life cycle: definition, development, production, operations, support, phaseout
    - Covers full system: satellite(s), ground systems, launch
    - Covers hardware, software, personnel costs
- Extensions to cover systems of systems, families of systems
- Several PhD dissertations involved (as with COSYSMO)
  - Incrementally developed based on priority, data availability
COSATMO Tentative Model

- **Total satellite system cost =**
  
  System engineering cost
  + Satellite software cost
  + Satellite vehicle hardware development and production cost
  + Launch cost
  + Initial ground software cost
  + Initial ground facility cost
  + Operation & support cost

- **Model as sum of submodels relates to models in COCOMO family**
COCOMO Family of Cost Models

Software Cost Models


Other Independent Estimation Models

COCOTS 2000, COSYSMO 2005, COSoSIMO 2007

Software Extensions


Legend:
- Model has been calibrated with historical project data and expert (Delphi) data
- Model is derived from COCOMO II
- Model has been calibrated with expert (Delphi) data

Dates indicate the time that the first paper was published for the model

2014/02/26
COSATMO Submodel Starting Points

- System engineering: COSYSMO, perhaps with add-ons
- Satellite vehicle hardware development and production: Current Aerospace hardware cost model(s); exploring extensions of COSYSMO for hardware cost estimation
- Satellite vehicle, ground system software development: COCOMO II, COCOTS, perhaps with add-ons
- Launch model: similarity model, based on vehicle mass, size, orbit
- Ground system equipment, supplies: construction, unit-cost, services cost models
- Operation & support: labor-grade-based cost models, software maintenance models
Summary

• Current and future trends create challenges for systems and software cost estimation
  – Mission challenges: emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems

• USC, NPS/AFIT, DoD Systems Engineering Research Center researching ways to address challenges
  – Beginning with space systems (COSATMO models)
  – Extendable to other DoD domains

• Forum includes related COCOMO-family workshops
  Wednesday AM: COSYSMO 3.0
  – Thursday AM: COCOMO III
COSYSMO 3.0 Context

- COSYSMO 1.0 focused on basic-project SE costs
  - 4 size drivers: #rqts, interfaces, scenarios, key algorithms
  - Weighted by complexity and added together
  - 14 cost drivers: 8 technical, 6 personnel-related
  - Calibrated to 50+ project data points from 7 companies
  - Adopted by Galorath, Price Systems, Softstar Systems

- COSYSMO 2.0 added SE-with-reuse effects to 1.0
  - Calibrated to 40+ BAE Systems project data points

- COSYSMO-REVL added rqts-volatility effects to 1.0
  - Calibrated to 25 Boeing project data points

- COSYSMO 3.0 proposes to harmonize 2.0 and REVL
  - And adding SE-for-reuse, SE-for-SoS interoperability effects
  - And revisiting COSYSMO 1.0 size and cost drivers
  - Also exploring COSYSMO for system development costing
COSYSMO 3.0 Directions
(Adapted from ARR slides [8])

Harmonize existing COSYSMO family models:

• Several factors affecting the COSYSMO cost model have been shown to be valuable in increasing estimation accuracy (terminology from [5]):
  – Reuse (simple model--SEWR) [3]
  – Reuse (with SEFR) [1]
  – Requirements volatility (SERV) [4]

The rating scales for these could be integrated into a comprehensive COSYSMO model.

Enhancement planned for inclusion:

• System-of-system considerations are hypothesized to affect system engineering costs:
  – Interoperability considerations [6]
Enhancements under discussion:

• Explore a model for total development cost based primarily on the COSYSMO parameters (Cole)
• Reduce the number of Effort Multipliers (Roedler)
Harmonized COSYSMO 3.0

Top-Level Model

\[ PM_{C3} = A_{C3} \times (Size_{C3})^{E_{C3}} \times EM_{C3, j} \]

\[ \sum_{j=1}^{14+} \]

Elements of the Harmonized COSYSMO 3.0 model:

- Calibration parameter \( A \)
- Interoperability
- Size model
  - eReq submodel
  - Partial development submodel
- Exponent (E) model
  - SF submodel
  - REVL submodel
- Effort multipliers EM
  - 14 unchanged EMs
  - SEFR
  - Interoperability
- Multi-subproject model
Harmonized COSYSMO 3.0 Interoperability Model

- Lane & Valerdi [6] propose that interoperability be considered a cost influence in the COSYSMO family.

- Motivation: if a system is part of a system-of-systems, then that context is reflected in interoperability requirements on the system.

- Two ways this influence could be manifested are proposed:
  - Method 1: Add a new effort multiplier
  - Method 2: Adjust the easy/medium/difficult rating scale for system interfaces (part of the Size model)

- Both Methods are shown in this presentation; presumably only one would be retained in COSYSMO 3.0.
Harmonized COSYSMO 3.0
Size Model

$$\text{Size}_{C3} = \frac{1}{\text{Prods}} \sum_{\text{Prods}} e\text{Req}(\text{Type}(\text{Prod}), \text{Difficulty}(\text{Prod})) \times$$

$$\text{PartialDevFactor}(\text{Phase}_{\text{Start}}(\text{Prod}), \text{Phase}_{\text{End}}(\text{Prod}))$$

- Prod is one of the four system engineering products that determines size in COSYSMO family (per [2]):
  - System requirement
  - System interface
  - System algorithm
  - Operational scenario

- For simplicity in model explanation, each individual Prod is considered separately

- There are two submodels:
  - Equivalent nominal requirements ("eReq")
  - Partial development
• A new, 15th effort multiplier is “System Engineering for Reuse (SEFR)”
  – I.e., is the project developing intermediate and final system engineering results to be reused on later projects?
    • Reuse for product line is one example
  – Inspired by [1]

• Assumes there is an added cost for SEFR

• Starting point for rating scale (as suggested by Boehm) is COCOMO II RUSE:
  – Low: Not for reuse
  – Nominal: Reused across project
  – High: Reused across program
  – Very High: Reused across product line
  – Extra High: Reused across multiple product lines
Adjustment for interoperability (Method 1):

- “Interoperability” might be a new, 16th effort multiplier
- Table 2 of [6] proposes this rating scale, depending on whether the project is for an existing system or a new system:

<table>
<thead>
<tr>
<th>Type of Development</th>
<th>Type of Development (based upon LISI levels)</th>
<th>Level</th>
<th>Level</th>
<th>Level</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Isolated</td>
<td>Connected</td>
<td>Functional standards employed</td>
<td>Domain standards employed</td>
<td>Enterprise standards employed</td>
</tr>
<tr>
<td>Existing systems</td>
<td>Functional standards employed</td>
<td>Domain standards employed</td>
<td>Enterprise standards employed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>New system (s)</td>
<td>System-specific data</td>
<td>Documented data</td>
<td>Aligned static data</td>
<td>Aligned dynamic data</td>
<td>Harmonized data</td>
</tr>
</tbody>
</table>
Summary

• Current and future trends create challenges for systems and software cost estimation
  – Mission challenges: emergent requirements, rapid change, net-centric systems of systems, COTS, clouds, apps, widgets, high assurance with agility, multi-mission systems

• USC, NPS/AFIT, DoD Systems Engineering Research Center researching ways to address challenges
  – Beginning with space systems (COSATMO models)
  – Extendable to other DoD domains

• Forum includes related COCOMO-family workshops
  – Wednesday AM: COSYSMO 3.0
  – Thursday AM: COCOMO III
Observations

• COCOMO II challenged by different development strategies
• 2000 calibration dataset is over 14 years old
• Productivity appears to be *increasing* over time
• Levels of reported process maturity increasing in Software Engineering data
• Productivity appears to decline with multiple incremental development
COCOMO II Challenges

1995: one-size-fits-all model for 21st century software

1999: poor fit for schedule-optimized projects; CORADMO

2000: poor fit for COTS-intensive projects: COCOTS

2003: need model for product line investment: COPLIMO

2003: poor fit for agile projects: Agile COCOMO II (partial)

2012: poor fit for incremental development: COINCOMO
COCOMO II Data by 5-Year Periods

- 1970-1974: 12 projects
- 1975-1979: 36 projects
- 1980-1984: 0 projects
- 1985-1989: 17 projects
- 1990-1994: 22 projects
- 1995-1999: 105 projects
- 2000-2004: 102 projects
- 2005-2009: 47 projects
COCOMO II Data: Productivity Trends
COCOMO II Data: Process Maturity Trends
Workshop Topics

1. Consider incorporating Software Application Domains
2. Discuss additional model forms
3. Review current set of cost drivers
# Super-Domains and AFCAA Productivity Types

<table>
<thead>
<tr>
<th>Super Domain</th>
<th>Productivity Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-Time (RT)</td>
<td>1 Sensor Control and Signal Processing</td>
</tr>
<tr>
<td></td>
<td>2 Vehicle Control</td>
</tr>
<tr>
<td></td>
<td>3 Vehicle Payload</td>
</tr>
<tr>
<td></td>
<td>4 Real Time Embedded-Other</td>
</tr>
<tr>
<td>Engineering (ENG)</td>
<td>5 Mission Processing</td>
</tr>
<tr>
<td></td>
<td>6 Executive</td>
</tr>
<tr>
<td></td>
<td>7 Automation and Process Control</td>
</tr>
<tr>
<td></td>
<td>8 Scientific Systems</td>
</tr>
<tr>
<td></td>
<td>9 Telecommunications</td>
</tr>
<tr>
<td>Mission Support (MS)</td>
<td>10 Planning Systems</td>
</tr>
<tr>
<td></td>
<td>11 Training</td>
</tr>
<tr>
<td></td>
<td>12 Software Tools</td>
</tr>
<tr>
<td></td>
<td>13 Test Software</td>
</tr>
<tr>
<td>Automated Information System (AIS)</td>
<td>14 Intelligence and Information Systems</td>
</tr>
<tr>
<td></td>
<td>Software Services</td>
</tr>
<tr>
<td></td>
<td>Software Applications</td>
</tr>
</tbody>
</table>
Additional Model Forms

• **Keep COCOMO II models?**
  – Application Composition
  – Early-Design
  – Post-Architecture

• **Should COCOMO III be backwards compatibility to COCOMO 81 & COOCMO II?**

• **New parameters, e.g.,**
  – to indicate the type of processes that are planned for the development e.g.: plan-driven, rapid development, architected agile, formal methods, COTS integration.
COCOMO II Cost Driver Review

- New cost driver values based on post-2000 data points
- Review cost drivers for
  - Relevance?
  - Additions / deletions?
- Which cost drivers need a better rating selection system that reduces rating subjectivity