COCOTS
(COnstructive COTS)
Software Integration
Cost Model: An Overview

Mr. Christopher M. Abts (USC)
Dr. Elizabeth (Betsy) Bailey (Software Metrics)
Dr. Barry W. Boehm (USC)
USC Center for Software Engineering
CSS '98, Irvine, CA
October 23, 1998

Copyright 1997, 1998
University of Southern California
Outline

• Model Development History and Support
• Problem Context
• COTS Software Integration Cost Sources
• COCOTS vs. COCOMO Cost Sources
• COTS Assessment
• COTS Tailoring
• COTS Glue Code Development and Test
• COTS Volatility Effects on Application Development Cost
• Total COTS Integration Cost Estimate
• Prospective COCOTS Follow-ons
• Conclusions

Model Development History and Support

• USAF/ESC Effort
  – March 1996 through June 1997
    • Initial Glue Code Model Definition, Experimental Calibration
• FAA Effort
  – Phase 1 (July to October, 1997)
    • Glue Code Model Redefinition, Experimental Calibration
  – Phase 2 (October 1997 to July 1998)
    • Glue Code Model Refinement
    • Assessment, Tailoring, and Volatility Models Defined
  – Phase 3 (July 1998 to December 1998)
    • Further Data Collection & Model Refinement, Calibration
    • Goal: calibrated model available by end 1998
• ONR Effort
  – January 1998 through 1999
    • Further Refinement of Models, including activity analysis & effort distribution
    • Data Collection & Calibration
    • Determination of How Best to Associate COCOTS with COCOMO II

Coordination of FAA and ONR Data Collection Being Pursued with Help of DoD
Problem Context: What is (and Isn’t) COTS?
- terms from recent Ground System Architectures Workshop

• COTS: Commercial Off-the-Shelf
• GOTS: Government Off-the-Shelf
• HOTS: Hot Off-the-Shelf
• NOTS: Not Off-the-Shelf
• ROTS: Research Off-the-Shelf

Problem Context: COTS Phenomena, Pitfalls and Practices

• You have no control over a COTS product’s functionality or performance.
• Most COTS products are not designed to interoperate with each other.
• You have no control over a COTS product’s evolution
• COTS vendor behavior varies widely
Problem Context: Modeling

- New COCOTS Modeling Problem
- COTS Infrastructure
  - COCOMO II: PVOL, PEXP
- COTS Tools
  - LTEX, TOOL

Cost Modeling Currently Addressed

COTS Software Integration Cost Sources*

1. COTS Assessment
   - Initial Filtering
   - Final Selection
2. COTS Tailoring
3. COTS Application Glue Code Development and (System) Test
4. COTS Volatility Effects on Application Development Cost

*Initial COCOTS Focus: Software Development; Operations & Maintenance to be addressed later
COCOMO Cost Sources
(No COTS in System)

Application Code Development Integration and Test Without The Use of COTS in System

COCOMO vs. COCOTS Cost Sources
(COTS in System)

Application Code Development Integration and Test Separate from COTS (md:ct)

TIME

STAFFING
Objectory Management Checkpoints

<table>
<thead>
<tr>
<th>Major Milestones</th>
<th>Inception</th>
<th>Elaboration</th>
<th>Construction</th>
<th>Transition</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCO</td>
<td>Iteration 1</td>
<td>Iteration 2</td>
<td>Iteration 3</td>
<td>Iteration 4</td>
</tr>
<tr>
<td>LCA</td>
<td>Iteration 5</td>
<td>Iteration 6</td>
<td>Iteration 7</td>
<td>Iteration 8</td>
</tr>
<tr>
<td>IOC</td>
<td>Full Release</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Strategic focus on global concerns of the entire software project

Minor Milestones

Tactical focus on local concerns of current iteration

Status Assessments

Periodic synchronization of stakeholder expectations

Objectory Information Set Evolution

<table>
<thead>
<tr>
<th>Engineering Stage</th>
<th>Manufacturing Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception</td>
<td>LCO</td>
</tr>
<tr>
<td>Elaboration</td>
<td>LCA</td>
</tr>
<tr>
<td>Construction</td>
<td>IOC</td>
</tr>
<tr>
<td>Transition</td>
<td></td>
</tr>
</tbody>
</table>

Feasibility Iterations

Architecture Iterations

Usable Iterations

Product Releases

RATIONAL
COTS Integration Cost Sources:

1) Assessment

Initial Filtering Effort

Total Effort = \( \left( \text{# COTS Candidates} \right) \left( \frac{\text{Average Filtering Effort}}{\text{Candidate}} \right) \)

Final Selection Effort

Total Effort = \( \sum \left( \text{# COTS Candidates} \right) \left( \frac{\text{Average Assessment Effort for Attribute in Given Domain}}{\text{Candidate}} \right) \)

- List of attributes refined in collaboration with Dr. Elizabeth Bailey
- Effort/candidate is project-dependent, within domain guidelines

---

**COTS Integration Cost Sources: Assessment Attributes**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Correctness</th>
<th>Understandability</th>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Correctness</td>
<td>Documentation quality</td>
<td>Portability</td>
</tr>
<tr>
<td>Simplicity</td>
<td>Testability</td>
<td>Functionality</td>
<td></td>
</tr>
<tr>
<td>Availability/Robustness</td>
<td>Availability</td>
<td>Ease of use</td>
<td>Functionality</td>
</tr>
<tr>
<td>Fail safe</td>
<td>Fail safe</td>
<td>Usability/Human Factors</td>
<td>Price</td>
</tr>
<tr>
<td>Fault tolerance</td>
<td>Fault tolerance</td>
<td>Version Compatibility</td>
<td>Recurring costs</td>
</tr>
<tr>
<td>Input error tolerance</td>
<td>Input error tolerance</td>
<td>Downward compatibility</td>
<td></td>
</tr>
<tr>
<td>Redundancy</td>
<td>Redundancy</td>
<td>Upward compatibility</td>
<td>Maturity</td>
</tr>
<tr>
<td>Reliability</td>
<td>Reliability</td>
<td>Product Maturity</td>
<td></td>
</tr>
<tr>
<td>Robustness</td>
<td>Robustness</td>
<td>Inter-component Compatibility</td>
<td>Vendor Maturity</td>
</tr>
<tr>
<td>Safety</td>
<td>Safety</td>
<td>Compatibility with other components</td>
<td></td>
</tr>
<tr>
<td>Performance</td>
<td>Performance</td>
<td>Interoperability</td>
<td>Vendor Support</td>
</tr>
<tr>
<td>Security</td>
<td>Security</td>
<td>Flexibility</td>
<td>Response time for critical problems</td>
</tr>
<tr>
<td>(Access related)</td>
<td>(Access related)</td>
<td>Extendability</td>
<td>Support</td>
</tr>
<tr>
<td>(Sabotage related)</td>
<td>(Sabotage related)</td>
<td>Warranty</td>
<td></td>
</tr>
<tr>
<td>Flexibility</td>
<td>Flexibility</td>
<td>User Training</td>
<td>User Training</td>
</tr>
<tr>
<td>Execution performance</td>
<td>Execution performance</td>
<td>Installation/Upgrade ease</td>
<td>User training</td>
</tr>
<tr>
<td>Interconnectivity capacity</td>
<td>Interconnectivity capacity</td>
<td>Installation Ease</td>
<td>User training</td>
</tr>
<tr>
<td>Precision</td>
<td>Precision</td>
<td>Upgrade/Refresh ease</td>
<td>Vendor Concessions</td>
</tr>
<tr>
<td>Memory performance</td>
<td>Memory performance</td>
<td>Willingness to escrow source code</td>
<td></td>
</tr>
<tr>
<td>Response time</td>
<td>Response time</td>
<td>Willingness to make modifications</td>
<td></td>
</tr>
<tr>
<td>Throughput</td>
<td>Throughput</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Center for Software Engineering
### COTS Integration Cost Sources:

#### 2) Tailoring

Total Effort = \[ \sum_{\text{Tailoring Complexity Levels}} \left( \frac{\text{# COTS Candidates Tailored at Complexity Level}}{\text{Average Effort at Tailoring Complexity Level in Domain}} \right) \]

---

**Five tailoring effort complexity levels:**

- Very Low, Low, Nominal, High, Very High
- Differentiated based on number tailored parameters, difficulty of needed scripts, API iterations, etc.

#### Tailoring Dimensions of Tailoring Difficulty

<table>
<thead>
<tr>
<th>Dimension</th>
<th>COTS Integration Cost Sources:</th>
<th>2) Tailoring - Dimensions of Tailoring Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter Specification</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>$\geq$ 50 parameters to be initialized</td>
<td>$\geq$ 100 parameters to be initialized</td>
</tr>
<tr>
<td>Script Writing</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Menu driven; 1 to 5 line scripts; 1 to 5 scripts needed</td>
<td>Menu driven; 6 to 10 line scripts; 6 to 10 scripts needed</td>
</tr>
<tr>
<td>TO Report &amp; GUI Screen Specification &amp; Layout</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>Automated or standard templates; 1 to 5 screen specifications</td>
<td>Automated or standard templates; 6 to 15 screen specifications</td>
</tr>
<tr>
<td>Availability of COTS Tailoring Tools</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

---

Center for Software Engineering
COTS Integration Cost Sources:
3) Glue Code Development and Test

Total Effort = A • [(size)(1+breakage)]^B \prod (effort multipliers)

- A - a linear scaling constant
- Size - of the glue code in SLOC or FP
- Breakage - of the glue code due to change in requirements and/or COTS volatility
- Effort Multipliers - 13 parameters, each with settings ranging VL to VH
- B - an architectural scale factor with settings VL to VH

Personnel Drivers
1) ACIEP - COTS Integrator Experience with Product
2) ACIPC - COTS Integrator Personnel Capability
3) AXCIP - Integrator Experience with COTS Integration Processes
4) APCON - Integrator Personnel Continuity

COTS Component Drivers
5) ACPMT - COTS Product Maturity
6) ACSEW - COTS Supplier Product Extension Willingness
7) ACPFX - COTS Product Interface Complexity
8) ACPS - COTS Supplier Product Support
9) ACPFD - COTS Supplier Provided Training and Documentation

Application/System Drivers
10) ACREL - Constraints on Application System/Subsystem Reliability
11) AACPX - Application Interface Complexity
12) ACPFR - Constraints on COTS Technical Performance
13) ASPRT - Application System Portability

Nonlinear Scale Factor
1) AAREN - Application Architectural Engineering
COTS Integration Cost Sources:
4) Increased Application Effort Due to COTS Volatility

Approximate Model:

Total Effort = (Application Effort) \times \left( \frac{\text{BRAK COTS}}{100} \right) \times \text{(EAF) COTS}

Detailed Model with COCOMO II Parameters:

Total Effort = (Application Effort) \times \left( \frac{1 + \text{BRAK COTS}}{1 + \text{BRAK}} \right) ^ {1.01 + \Sigma} \times \text{(EAF) COTS}

BRAK COTS: % application code breakage due to COTS volatility
BRAK : % application code breakage otherwise
Σ : COCOMO II scale factor
EAF : Effort Adjustment Factor (product of effort multipliers)

COTS Integration Cost Sources:
4) Increased Application Effort Due to COTS Volatility
- COCOMO II Scale Factors

<table>
<thead>
<tr>
<th>Scale Factor</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>Precededness</td>
<td>thoroughly unprecedented</td>
<td>Largely unprecedented</td>
<td>somewhat unprecedented</td>
<td>generally familiar</td>
<td>largely familiar</td>
<td>thoroughly familiar</td>
</tr>
<tr>
<td>Development</td>
<td>rigorous</td>
<td>Occasional Relaxation</td>
<td>some relaxation</td>
<td>general conformity</td>
<td>some conformity</td>
<td>general goals</td>
</tr>
<tr>
<td>Flexibility</td>
<td>Architecture/Risk Resolution</td>
<td>little (20%)</td>
<td>some (40%)</td>
<td>often (60%)</td>
<td>generally (75%)</td>
<td>mostly (90%)</td>
</tr>
<tr>
<td>Team Cohesion</td>
<td>some difficult interactions</td>
<td>Basically cooperative interactions</td>
<td>largely cooperative interactions</td>
<td>highly cooperative interactions</td>
<td>seamless interactions</td>
<td>N/A</td>
</tr>
<tr>
<td>Process Maturity</td>
<td>Chaos</td>
<td>CMM Level 1</td>
<td>CMM Level 2</td>
<td>CMM Level 3</td>
<td>CMM Level 4</td>
<td>CMM Level 5</td>
</tr>
</tbody>
</table>

*percentage of module interfaces specified, percentage of significant risks eliminated.
Total COTS Integration Cost Estimate

Total Integration Effort (in Person-Months) =
Assessment Effort + Tailoring Effort + Glue Code Effort + Volatility Effort

where
Assessment Effort = Filtering Effort + Final Selection Effort

Total integration Cost =
(Total Integration Effort) • ($$/Person-Month)

Prospective COCOTS Follow-ons

• Extensive data collection and conditioning
• Recalibration and iteration of the model within current structure
• Experimental usage and refinement, including exploration of other cost drivers and model forms
• Modeling of schedule estimation and activity distribution
• Integration with COCOMO II estimation model
• More extensive model implementation
• Modeling other COTS related costs
  – Licenses, training, maintenance, hardware
Modeling Other COTS Related Costs

• Largely a (unit cost) * (# units) framework
  – Unit costs vary by quantity, platform, time
• Need to consider time-phasing of acquisition, implementation, operations & maintenance
• Biggest challenge will be complex, dynamic COTS price structures

Extended COCOTS Model

\[
\text{Cost (t)} = \sum \text{Cost [SW development]} + \text{Cost [SW maintenance] (t)} + \text{Cost [SW COTS integration]} + \text{Cost [SW COTS integ. maint.] (t)} + \text{Cost [SW COTS] (t)} + \text{Cost [HW COTS] (t)}
\]

-- COCOMO II, others
-- COCOMO II, others
-- COCOTS
-- COCOTS
-- (see chart following)
-- (see chart following)
Estimating Cost of Software COTS

\[ \text{Cost [SW COTS]}(t) = \]
\[ \text{Cost[acquisition office]}(t) + \]
\[ \text{Cost[licenses]}(t) \quad (= \text{Cost[#licenses,#features,#platforms]}(t)) + \]
\[ \text{Cost[implementation]}(t) \quad (= \text{Cost[training]}(t) + \text{Cost[install]}(t)) + \]
\[ \text{Cost[Op. & Mnt.]}(t) \quad (= \text{Cost[maint. lic.’s.]}(t) + \text{Cost[support]}(t)) \]

Estimating Cost of Hardware COTS

\[ \text{Cost[HW COTS]}(t) = \]
\[ \text{Cost[acquisition office]}(t) + \]
\[ \sum_i \left( \text{Cost[acquistion]}_i(t) + \text{Cost[implementation]}_i(t) + \text{Cost[O&M]}_i(t) \right) \]

\[ \text{for} \]
\[ i = \text{processors, storage, workstation, communications} \]
Conclusions

- COCOTS provides solid framework for estimating software COTS integration cost
  - needs further data, calibration, iteration
  - current spreadsheet model could be used experimentally
- COCOTS can be extended to cover other COTS related costs
  - biggest challenge will be complex, dynamic COTS price structures