COCOMO Forum

Sizing Breakout Group
Outline

Group (max) Introductions: Motorola–6; The Aerospace Corp.–2; JPL, IBM, GDAIS, MCR Federal; USC-CSE–4

What CSE Has to Offer

Categories Further Discussion
• ? Sizing ... for what purpose
• "Polishing" the LoC counting
• LoC vs. non-LoC counting approaches
• Domain cross-cutting

Cross Cutting Issues

Additional Shared Information

MBASE vs. RUP

One Liner SoWs for PhD Students

Importance Vs. Difficulty

Backup Slides: Topics Brainstormed for Discussion
Introductions

The Breakout Group: Objectives and Proposed Agenda

Participants: Name, Organization: Sizing hot-button

- Peter Hantos, [The] Aerospace Corp.: Backfiring causes more problems than it solves
- Ira Forman, IBM Global Procurement: wants to relate size to $’s
- Robert Dejong, Motorola Toronto: What is the baseline for size
- Danny Ho, Motorola Toronto: Wants help for on size estimation
- Erik Monson, JPL: Logical SLOC vs. Object Oriented Languages
- Ron Kelley, GD AIS: What is a real alternative to LOC
- Cheryl Calimlim, MCR Federal: Here to learn
Introductions (cont.)

More Participants

- Cvetan Redzic, Motorola Arlington Heights: Size estimation

- Jongmoon Baik, Motorola Schaumburg: Sizing methods for UML & SDL (Specification Description Language)

- Rhoda Novak, Aerospace Corp.: How to do RoM sizing earlier in the program life than requirements, interfaces, ....

- Nancy Eickelmann, Motorola Schaumburg:

- Hillel Meyers, Motorola Schaumburg: 4GL and 5GL; normalization(?) back to 3GL

- Jo Ann Lane, San Diego State University: Use-Cases and how to resolve the overlap/inflated estimates (non-repeating elements)
Introductions (cont.)

Still More Participants

- Dr. Barry Boehm, USC CSE:
  COCOMO should join the world of UML and OO, (21st Century?)

- A Winsor Brown, USC CSE:
  Sizing systems of systems

- Yue Chen, USC CSE:
  UML Sizing study; CodeCounters

- Sushil Chaudhry, USC CSE:
  CodeCount Product Line Support Manager
What CSE Has to Offer

**CodeCounters** (all in ANSI Standard C; provided via "CopyLeft" in source code)

- 9 Traditional 3GL languages:
  - Ada, Assembly, C and C++, COBOL, FORTRAN, Java, JOVIAL, Pascal, PL/1

- 7 Newer languages (not ready for prime time):
  - Excel; HTML [has SLOC def]; Javascript; Perl; SQL; Visual Basic; VB Script

- Identified "bugs" in existing codeCounters [need to be followed up on]

- Newer (in development)
  - Web Object counter
  - "New Architecture" counter for PhP and Perl (in Java)
    - "architeceted to add new languages in a month"; degree of testing is not clear
    - How to distribute and support not clear; IPR needs to be clarified

- Documentation: [needs to summarized and made evident]; add "How to generate new code counters"
What CSE Has to Offer (cont.)

Approaches to counting

- UML (Use-Cases)
- MBASE SSRD requirements (loosely on J-16 standard; J-STD-016-1995; 30 September 1995; Trial-Use Standard; Standard for Information Technology; Software Life Cycle Processes Software; Development; Acquirer-Supplier Agreement)
  - Project
  - Capabilities
  - Interface
  - Levels of Service
  - Evolution (must be "architechted" and "designed")

Rosetta stone experience

- COCOMO II versions
- Drivers: Post Architecture and Early Design versions

Lessons learned: Not clear if they have been [well] captured
What CSE Has to Offer (cont.)

In development

- DBaCOCOMOII (for regular user interface)
  - Keep track ALL inputs by "stage" of project
  - Model the changes over time

- COSoSIMO: Constructive System Of Systems Integration Model
Categories Further Discussion

? Sizing ... for what purpose?

"Polishing" the LoC counting

LoC vs. non-LoC counting approaches

Domain cross-cutting
Categories Further Discussion Expansion

? Sizing ... for what purpose?

- CSE for estimation of cost, schedule & **quality** (AKA residual defect)
- Estimation of Cost and Schedule (with an assumed quality)
- [for acquisition] Estimation of x-ilities (product quality level) for comparison/selection of suppliers
- Product sizing
- Functional sizing
- Cost of non-quality
- Price
- Activity (non-code) costing
Categories Further Discussion Expansion (cont.)

"Polishing" the LoC counting

- Counting vs. language
- >3GL counting (4 GL or 5 GL programming language)
  - Rules for inclusion in xGL
- Given that size is used as a proxy for schedule and effort, finding other measurable "proxies" for > 3GL
Categories Further Discussion Expansion (cont.)

LoC vs. non-LoC counting approaches

- Life-cycle phase dependency
- What sizing measure should use and when
- Alternate counting
Categories Further Discussion Expansion (cont.)

Domain cross-cutting

• Rosetta stones
• Sizing vs. Domain
• System of System Integration
Cross Cutting Issues

Experience and Background Level
of student researchers, and advisors or affiliate supporters

- How to get realistic solutions
- How to solve the right problem

Scaling

- Project size
- Crafting vs. systematic, large-scale development
- Number of {CSCIs, components, COTS products, ...}
- Granularity: What are nano, micro, macro, molecular, ... (organism?)

Drivers (Parametric Model) scalability
under impacts of size of product, organization; culture; number of separately "developed" systems; ...

Suppliers and subcontracts
Additional Shared Information

Sizing Calibration.vsd

- Highlighted the need for delta/diff counter
- Need for tracking parametric drivers change over time as well as New, Modified, Reused changes

Historical Data Points

New Data Point

Note: Pre-construction estimation result could be got at LCO or LCA milestone.

Sizing Model Improvement Cycle
Additional Shared Information (cont.)

SizingCalibrationV1.vsd: Multiple sizing and estimating; objects used.

- Pre-construction estimation result could be got at LCO or LCA milestone.

Sizing Model Improvement Cycle

Note: Pre-construction estimation result could be got at LCO or LCA milestone.
MBASE vs RUP

Business case
- Serve and Satisfy Stakeholders
- Risk management
- Domain models; Domain Models
- Documentation
- Schedule
- Performance parameters for Reliability

Process models
- Milestone context
- Evaluation and analysis
- Property models
- Cost Schedule Performance Reliability

Product models
- Planning and control
- Product Evaluation criteria

Success models
- Life cycle Anchor Points; Risk management
- Key practices

MBASE vs. RUP

© 2003 USC Center for Software Engineering
MBASE vs. RUP

Disciplines

Business Modeling
Requirements
Analysis & Design
Implementation
Test
Deployment
Configuration & Change Mgmt
Project Management
Environment

Phases

Inception
Elaboration
Construction
Transition

Iterations

Initial
Elab #1
Elab #2
Const #1
Const #2
Const #N
Trans #1
Trans #2

Time
LCO
LCA
IOC
MBASE vs. RUP (cont.)
MBASE Model Integration Framework

- Business case
  - IKIWISI
  - Stakeholder win-win

- Process models
  - Life cycle anchor points
  - Risk management
  - Key practices

- Success models
  - Process entry/exit criteria

- Planning and control
  - Milestone content

- Property models
  - Evaluation and analysis

- Product models
  - Cost
  - Schedule
  - Performance
  - Reliability

- Product evaluation criteria
# MBASE vs. RUP (cont.)
## MBASE Invariants and Variants

<table>
<thead>
<tr>
<th>Invariants</th>
<th>Variants</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Defining and sustaining a stakeholder win-win relationship through the system's life-cycle.</td>
<td>1. Use of particular success, process, product, or property models.</td>
</tr>
<tr>
<td>2. Using the MBASE Model Integration Framework.</td>
<td>2. Choice of process or product representation.</td>
</tr>
<tr>
<td>4. Using the LCO, LCA, and IOC Anchor Point milestones.</td>
<td>4. Number of spiral cycles or builds between anchor points.</td>
</tr>
<tr>
<td>5. Ensuring that the content of MBASE artifacts and activities is risk-driven.</td>
<td>5. Mapping of activities onto Inception-Elaboration-Construction-Transition phases.</td>
</tr>
<tr>
<td></td>
<td>6. Mapping of staff levels onto activities.</td>
</tr>
</tbody>
</table>
MBASE vs. RUP (cont.)

MBASE Model Integration Process

- Identify and prioritize
- Provide measures for
- Property Models
- Constrain
- Provide parameters for
- Process Models
- Constrain
- Provide parameters for
- Product Models
- Provide measures for
- Success Models

- Describe enterprise context in
- Domain Models
- Set context for
- Serve and Satisfy
- Stakeholders
One Liner SoWs for PhD Students

1. How to size 4GLs and 5GLs (for which SLOC fails); what are their SHLLLOC [Source High Level Language LOC]

2. Investigate backwards compatibility allowing use of existing historical data.

2A. Of LoC Counting Tools: How many versions will be supported.

2B. Of COCOMO II (when to let old data die)

3. Find and validate robust ways of mapping between alternate sizing approaches.

4. How to make compilers do the counting

5. Document the processes at the intersection of "size" estimating and "cost/schedule" estimating
One Liner SoWs for PhD Students (cont.)

6. Analyze the COCOMOII "Local Calibration" by using original size estimate and updated parametric drivers [and refine if necessary?]: identify sources of variation and help predict ranges of early estimates.

7. How evolving "models" interact with estimation:
   Normalize(?) levels of representations in "models"
   • At MBASE Milestones: IRR, LCO and LCA
   • At risk of fullness of definition (tacit vs. explicit knowledge)
   • Validate by investigating applicability
     – At M-GATES at Motorola: M13, M10 & M7 (actuals at M2)
       (15 points to look at; 3 points of estimate and 3 corporate approval/reviews)
     – At Xerox's TTM milestones

8. Given that size is used as a proxy for schedule and effort, find other measurable “proxies” for 4GLs and 5GLs
One Liner SoWs for PhD Students (cont.)

9. Given that size is used as a proxy for schedule and effort, find other measurable size "proxies" for life-cycle phases (including inception, transition, SoSI integration)

10. Investigate existence of a "calculus for sizing" (like relational calculus was to RDBMS)

11. Validate applicability of a variant of Parkinsons Law: (amount of time spent on a sizing a thing [AKA budgetary item] is inversely proportional to cost of the thing)

12. Investigate sizing approaches for estimation of x-ilities (product quality level)
   - for acquisition's for comparison/selection of suppliers
   - for time-to-market trades
   - for/during "warranty/maintenance" periods
One Liner SoWs for PhD Students (cont.)

13. Investigate sizing for (non-coding) activity costing

- replace "wrap-factors" (% for documentation, systems engineering, especially system of system integration)
- new integration and testing tools; Operations and Support tools; etc.
Importance Vs. Difficulty

<table>
<thead>
<tr>
<th>Topic</th>
<th>Importance</th>
<th>Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investigate backwards compatibility allowing use of existing LOCC</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Investigate backwards compatibility allowing use of existing historical data?</td>
<td>High</td>
<td>Medium-High</td>
</tr>
<tr>
<td>How evolving &quot;models&quot; interacts with estimation: Normalize(?)</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Investigate existence of a &quot;calculus for sizing&quot; (like relational calculus was to ROBIS)</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Investigate applicability of a variant of Parkinson's Law: (amount of time spent on a sizing thing [aka venture item] is inversely proportional to cost of the thing)</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>Investigate sizing approaches for estimation of x-kilobit (product)</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

How to size SLL and SLL (for when SLLU tests); what are their Importance vs. Difficulty:

1. Investigate backwards compatibility allowing use of existing LOCC.
2. Investigate backwards compatibility allowing use of existing historical data?
3. How evolving "models" interacts with estimation: Normalize(?)
4. Investigate existence of a "calculus for sizing" (like relational calculus was to ROBIS).
5. Investigate applicability of a variant of Parkinson's Law: (amount of time spent on a sizing thing [aka venture item] is inversely proportional to cost of the thing).

- Importance: High, Medium-High, Low
- Difficulty: High, Medium, Low
Topics Brainstormed for Discussion

No-on FP, especially backfiring

Different counting per Life-cycle phases (for all life-cycle costs)

- Mappings across units

Does OOProgramming change the way one does counting

- Erik: In the small – coding C in the C++, and impact on SLOC
- Ira: C++ is harmful to software engineers
- Ira: More and more functionality from OS and Platform
- Erik: Realtime Java for Flight systems

Cross-checking sizing using multiple approaches
(high level sanity checks for Space Missions Impossible Center)
Topics Brainstormed for Discussion (cont.)

5GL, autocode generation, executable UML: all force us away from traditional approaches (SLOC & FP?)

How is sizing related to the WBS/activity: sizing measure has different impacts

Cone of Uncertainty
- Error Bounds: helping people be consistent

Are there aspects of sizing that can cut across the different domains (IT to airborne)

What is the Calculus of Sizing?

What does CSE offer today relative to sizing
- Counters
- Approaches to counting
- Rosetta stone experience
- Lessons learned
Topics Brainstormed for Discussion (cont.)

3GL :: 4GL :: 5GL [re-clarify the terms for 21\textsuperscript{st} Century]

- 3GL = classic procedural language
- 4GL = Scripting language?
  - RPG?
  - 80's (90's) definition: Mixed procedural, infrastructure (DB) and high functionality UI?
  - All "coding" in text ("interpreted", converted into instructions for computer
- 5GL = coding in Icons?
  - Auto-code generation?
  - Visual languages?
  - DBMSs with icons and controls?
  - Non-traditional programming paradigms:
    - Excel, emacs,

Are there 5.1, 5.2, ... extrapolations
Topics Brainstormed for Discussion (cont.)

Size :: {Quality; Complexity; Reliability and otherilities; development] Schedule Constraints; }

• What is quality?
  – Conformance to specification?
  – "Defects": Faults, errors,
  – Fullfilling "needs" of customer
• Complexity: IKWISI
• Size: estimating vs. measuring
• ilities
Topics Brainstormed for Discussion (cont.)

What's known in the end; what can be estimated early in the lifecycle?

What if you can't change translate into lines of code?

What about backwards compatibility allowing use of existing historical data?

Are there robust ways of mapping between alternate sizing approaches?

Things on CSE's backburner:

- Is productivity the same per line of source code for every different "level" of languages?
- COCOMO II Extrapolation beyond Elaboration and Construction
Topics Brainstormed for Discussion (cont.)

Things on CSE's front burners:

- COSoSIMO: System of Systems Integration (post construction of individual systems)
- Yue's Sizing studies
- Integrated (multiple languages), delta/differencing logical line code counters
  - Delta counts: e.g. changing just condition of a for loop (one of the three components)
  - logical diff
- Incrementally useful (WinWin Spiral in the time box) vs. global, re-orientation