Software Cost Estimation Meets Software Diversity

Barry Boehm, USC
STC 2017 Keynote
September 26, 2017
Outline

Sources of Software Diversity
- A Short History of Software Estimation Accuracy
- Process, Product, Property, and People Drivers

Options for Software Cost Estimation
- Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites

Best Fits of Estimation-Types to Diversity-Types
- Extensions of ICSM Common Cases

Charting Your Path to Improved Estimates
A Short History of Software Estimation Accuracy

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 Precedent Component-based COTS Agile SoS. Apps, Widgets, IDPD, Clouds, Security, MBSSE

A B C D

Time, Domain Understanding

Copyright © USC-CSSE 2017
Future Software Process Diversity

- **Sequential Phases**
  - Waterfall, V-Model

- **Sequential Increments**
  - Most agile methods: XP, Scrum, Crystal, SAFE
  - Pre-Planned Product Improvement (P3I)

- **Continuous reprioritization**
  - Kanban, DevOps

- **Evolutionary Definition and Development**
  - Incremental Commitment Spiral, Rational Unified Process

- **Fully concurrent: Open Source**
# ICSM Common Case Examples

<table>
<thead>
<tr>
<th>Accounting Application</th>
<th>Simple Customer Business App</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size/Complexity:</strong> Small/low</td>
<td><strong>Size/Complexity:</strong> Small/low</td>
</tr>
<tr>
<td><strong>Typical Change Rate/Month:</strong> Low</td>
<td><strong>Typical Change Rate/Month:</strong> Medium to high</td>
</tr>
<tr>
<td><strong>Criticality:</strong> High</td>
<td><strong>Criticality:</strong> Medium</td>
</tr>
<tr>
<td><strong>NDI Support:</strong> NDI-driven architecture</td>
<td><strong>NDI Support:</strong> No COTS, development and target environment well-defined</td>
</tr>
<tr>
<td><strong>Organizational Personnel Capability:</strong> NDI-experienced, medium to high</td>
<td><strong>Organizational Personnel Capability:</strong> Agile-ready, domain experience high</td>
</tr>
<tr>
<td><strong>Software Strategy:</strong> COTS</td>
<td><strong>Software Strategy:</strong> Architected agile</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cellphone Feature</th>
<th>Security Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Size/Complexity:</strong> Medium/medium</td>
<td><strong>Size/Complexity:</strong> Small/low</td>
</tr>
<tr>
<td><strong>Typical Change Rate/Month:</strong> Medium to high</td>
<td><strong>Typical Change Rate/Month:</strong> Low</td>
</tr>
<tr>
<td><strong>Criticality:</strong> Low</td>
<td><strong>Criticality:</strong> Extra high</td>
</tr>
<tr>
<td><strong>NDI Support:</strong> No COTS, development and target environment well-defined</td>
<td><strong>NDI Support:</strong> No COTS, development and target environment well-defined</td>
</tr>
<tr>
<td><strong>Organizational Personnel Capability:</strong> Agile-ready, domain experience high</td>
<td><strong>Organizational Personnel Capability:</strong> Strong formal methods experience</td>
</tr>
<tr>
<td><strong>Software Strategy:</strong> Agile</td>
<td><strong>Software Strategy:</strong> Formal methods</td>
</tr>
</tbody>
</table>
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
  - 20,000 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
Future Software Product Diversity

• Developed, Reused, Generated Software
  – Source Lines of Code (SLOC), Function Points (FP)
  – Reused: Equivalent SLOC
  – Generated: Model Directives

• Product Line Definition and Development
  – Reused, Modified, Generated SLOC or FP

• Non-Developmental Items (NDI), Cloud Services
  – NDI: Commercial Off-the-Shelf (COTS), Open Source
  – Costing: Assessment, Tailoring, Glue Code, New-Release Adaptation

• Domain Languages: Business, Supply Chain, Space

• Datasource-Driven: Selection Criteria
Reuse at HP’s Queensferry Telecommunication Division

Time to Market (months)

Year

<table>
<thead>
<tr>
<th>Year</th>
<th>86</th>
<th>87</th>
<th>88</th>
<th>89</th>
<th>90</th>
<th>91</th>
<th>92</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>

Non-reuse Project
Reuse project
Multi-Mission Support 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
The Basic COPLIMO Model
- Constructive Product Line Investment Model

• Based on COCOMO II software cost model
  – Statistically calibrated to 161 projects, representing 18 diverse organizations

• Based on standard software reuse economic terms
  – RCR: Relative cost of reuse
  – RCWR: Relative cost of writing for reuse

• Avoids overestimation
  – Avoids RCWR for non-reused components

• Provides experience-based default parameter values

• Simple Excel spreadsheet model
  – Easy to modify, extend, interoperate
# Basic COPLIMO Output Summary

### Summary of Inputs:

<table>
<thead>
<tr>
<th>Input</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVPROD</td>
<td>300</td>
</tr>
<tr>
<td>AVSIZE</td>
<td>50000</td>
</tr>
<tr>
<td>UNIQ%</td>
<td>40</td>
</tr>
<tr>
<td>ADAP%</td>
<td>30</td>
</tr>
<tr>
<td>RUSE%</td>
<td>30</td>
</tr>
<tr>
<td>RCR-UNIQ</td>
<td>100</td>
</tr>
<tr>
<td>RCR-ADAP</td>
<td>40</td>
</tr>
<tr>
<td>RCR-RUSE</td>
<td>5</td>
</tr>
<tr>
<td>RCWR</td>
<td>1.7</td>
</tr>
</tbody>
</table>

(Note: Do not change above values!)
(Change from "Input" sheet.)

### 7 year Product Line Effort Savings:

The chart shows the product line development cost estimation, with net development effort savings calculated.

### Table of Results:

<table>
<thead>
<tr>
<th># of Products</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique SLOC</td>
<td>0</td>
<td>20000</td>
<td>40000</td>
<td>60000</td>
<td>80000</td>
<td>100000</td>
<td>120000</td>
<td>140000</td>
</tr>
<tr>
<td>Adapted SLOC</td>
<td>0</td>
<td>15000</td>
<td>30000</td>
<td>45000</td>
<td>60000</td>
<td>75000</td>
<td>90000</td>
<td>105000</td>
</tr>
<tr>
<td>Reused SLOC</td>
<td>0</td>
<td>15000</td>
<td>30000</td>
<td>45000</td>
<td>60000</td>
<td>75000</td>
<td>90000</td>
<td>105000</td>
</tr>
<tr>
<td>Total Non-PL SLOC</td>
<td>0</td>
<td>50000</td>
<td>100000</td>
<td>150000</td>
<td>200000</td>
<td>250000</td>
<td>300000</td>
<td>350000</td>
</tr>
<tr>
<td>Non-PL Effort (PM)</td>
<td>0</td>
<td>166.667</td>
<td>333.333</td>
<td>500</td>
<td>666.667</td>
<td>833.333</td>
<td>1000</td>
<td>1166.667</td>
</tr>
<tr>
<td>1-Product Equiv. SLOC</td>
<td>0</td>
<td>71000</td>
<td>26750</td>
<td>26750</td>
<td>26750</td>
<td>26750</td>
<td>26750</td>
<td>26750</td>
</tr>
<tr>
<td>1-Product Equiv. Effort</td>
<td>0</td>
<td>236.667</td>
<td>89.1667</td>
<td>89.1667</td>
<td>89.1667</td>
<td>89.1667</td>
<td>89.1667</td>
<td>89.1667</td>
</tr>
<tr>
<td>Cum. Equiv. PL SLOC</td>
<td>0</td>
<td>71000</td>
<td>97750</td>
<td>124500</td>
<td>151250</td>
<td>178000</td>
<td>204750</td>
<td>231500</td>
</tr>
<tr>
<td>Cum. PL Effort</td>
<td>0</td>
<td>236.667</td>
<td>325.833</td>
<td>415</td>
<td>504.167</td>
<td>593.333</td>
<td>682.5</td>
<td>771.667</td>
</tr>
<tr>
<td>PL Effort Savings</td>
<td>0</td>
<td>-70</td>
<td>7.5</td>
<td>85</td>
<td>162.5</td>
<td>240</td>
<td>317.5</td>
<td>395</td>
</tr>
<tr>
<td>PL Reuse Investment</td>
<td>0</td>
<td>70</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Return on Investment</td>
<td>N/A</td>
<td>-1</td>
<td>0.10714</td>
<td>1.21429</td>
<td>2.32143</td>
<td>3.42857</td>
<td>4.53571</td>
<td>5.642857</td>
</tr>
</tbody>
</table>
Persistence of Legacy Systems

• Before establishing new-system increments
  – Determine how to undo legacy system

1939’s Science Fiction World of 2000

Actual World of 2000
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

Legacy Business Services

Contract Services
- Deliverables Management
- Subcontracting
- Billing

Project Services
- Staffing
- Work Breakdown Structure
- Scheduling
- Change Tracking
- Progress Tracking
- Earned Value Management
- Reqs, Configuration Management
Result of Legacy Re-engineering

Legacy Business Services

Contract Services
- Contract Financial Services
  - Billing
  - Subcontract payments
- Contract Non-Financial Services
  - Deliverables mgmt.
  - Terms compliance
- General Financial Services
  - Accounting
  - Budgeting
  - Earned value
  - Payroll
- General Non-Financial Services
  - Progress tracking
  - Change tracking

Project Services
- Project Financial Services
  - WBS
  - Expenditure categories
- Project Non-Financial Services
  - Scheduling
  - Staffing
  - Reqs CM
Future Software Properties Diversity

• Dependability
  – Reliability, Availability, Safety, Security

• Changeability
  – Adaptability, Maintainability, Modifiability, Repairability

• Mission Effectiveness
  – Response Time, Throughput, Accuracy, Usability, Scalability, Interoperability

• Life Cycle Efficiency (Cost-Effectiveness)
  – Development and Maintenance Cost, Schedule; Reusability
Response Time Rqt. Impact on Cost

- **$100M**
- **$50M**

### Required Architecture:
- Custom; many cache processors

### Original Architecture:
- Modified Client-Server

### Graph:
- Original Spec
- After Prototyping

<table>
<thead>
<tr>
<th>Response Time (sec)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>After Prototyping</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Copyright © USC-CSSE 09/26/2017
Better, Cheaper, Faster: Pick Any Two
COCOMO II Model Results

- For 100-KSLOC set of features
- Can “pick all three” with 77-KSLOC set of features
Future Software **People** Diversity

- **Desired Software People Capabilities**
  - Software System Analysis
  - Software System Development
  - Application Domain Experience
  - Software Languages and Tools Experience
  - Software Process Maturity
  - Team Cohesion
  - Low Personnel Turnover
  - Familiarity with Apps, Widgets, Social Media, Data Analytics, Multimedia, Virtual Reality
Outline

• Sources of Software Diversity
  – A Short History of Software Estimation Accuracy
  – Process, Product, Property, and People Drivers

→ Options for Software Cost Estimation
  – Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites

• Best Fits of Estimation-Types to Diversity-Types
  – Extensions of ICSM Common Cases

• Charting Your Path to Improved Estimates
Estimation-Type Options

- Expert-Judgement; Stakeholder Consensus
  - Planning Poker, Wideband Delphi, Bottom-Up
- Analogy: Previous Projects; Yesterday’s Weather
  - Agile COCOMO II, Case-Based Reasoning, Causal Modeling
- Parametric Models
  - COCOMO/COSTAR, Knowledge Plan, SEER, SLIM, True-S
- Resource-Limited
  - Cost or Schedule as Independent Variable (CAIV, SAIV)
- Reuse-Driven: Equivalent Size
  - Adjusted for %Design, Code, Test Modified, Understandability
- Product Line
  - % Development for Reuse; % Development with Reuse
Outline

• Sources of Software Diversity
  – A Short History of Software Estimation Accuracy
  – Process, Product, Property, and People Drivers

• Options for Software Cost Estimation
  – Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites

  Best Fits of Estimation-Types to Diversity-Types
  – Extensions of ICSM Common Cases

• Charting Your Path to Improved Estimates
Best Fits of Estimation-Types to Diversity-Types

• Pure Agile: Planning Poker, Agile COCOMO II

• Architected Agile
  – COSYSMO for architecting; Planning Poker, CAIV-SAIV for sprints, releases; IDPD for large systems

• Formal Methods: $/SLOC by Evaluated Assurance Level

• NDI/Services-Intensive: Oracle, SAP, other ERP
  – RICE Objects: (R)eports, (I)nterfaces, (C)onversions, (E)nhancements
  – COCOTS, Value-Added Function Points, Agile for portions

• Hybrid Agile/Plan-Driven
  – Expert Delphi, Parametric Models, Agile for portions; IDPD

• Systems of Systems
  – COSYSMO for Integrator; Hybrid Agile/Plan-Driven for component systems

• Family of Systems: COPLIMO

• Brownfield: Experiment for refactoring; above for rebuilding
Proliferation of Estimation Types
Thanks to Capers Jones

- **Source Lines of Code (SLOC)**
  - Physical/Logical; Executable/nonexecutable; New/reused; Programmed/generated/translated; Added/modified/deleted

- **Function points (FP)**
  - Original IBM; IFPUG 2,3,4; Fast; COSMIC; Mark II, FISMA, NESMA; Unadjusted/adjusted; RICE Objects

- **SLOC/FP backfire ratios**
  - SPR, QSM, DAVIDS, Gartner Group

- **Agile sizing**
  - Story points (Planning Poker, T-shirt size); ideal person-weeks

- **Risky: high variability**
  - Number of requirements/shalls; nonfunctional requirements (SNAP points); UML diagram counts
Outline

• Sources of Software Diversity
  – A Short History of Software Estimation Accuracy
  – Process, Product, Property, and People Drivers

• Options for Software Cost Estimation
  – Expert Judgement/Consensus; Size-Based; Productivity-Based; Component-Based; Process-Based; Composites

• Best Fits of Estimation-Types to Diversity-Types
  – Extensions of ICSM Common Cases

Charting Your Path to Improved Estimates
Charting Your Path to Improved Estimates

- Identify your most critical future improvement areas

- Identify, experiment with best candidate estimation methods in most critical areas

- Experiment with available methods for others; evaluate further improvement needs

- Build up, analyze experience base, use to steer path
COCOMO II Experience Factory: I

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

COCOMO 2.0

Rescope

Cost, Sched, Risks

Ok?

N

Yes

No
COCOMO II Experience Factory: II

System objectives: fcn’y, perf., quality

Corporate parameters: tools, processes, reuse

COCOMO 2.0

Rescope

Execute project to next Milestone

Milestone plans, resources

Milestone expectations

Ok?

Ok?

Yes

Yes

No

No

Revise Milestones, Plans, Resources

M/S Results

Revised Expectations

Yes

No

Done?

End

Cost, Sched, Risks

Yes

Copyright © USC-CSSE
COCOMO II Experience Factory: III

Corporate parameters: tools, processes, reuse

System objectives: fcn’y, perf., quality

COCOMO 2.0

Rescope

Cost, Sched, Risks

Ok?

Yes

No

Execute project to next Milestone

Milestone plans, resources

Ok?

Yes

M/S Results

No

Revised Expectations

Yes

Done?

Yes

End

No

Revised Milestones, Plans, Resources

Accumulate COCOMO 2.0 calibration data

Recalibrate COCOMO 2.0

Yes

Milestone expectations

No
COCOMO II Experience Factory: IV

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

COCOMO 2.0

Rescope

Execute project to next Milestone

Revise Milestones, Plans, Resources

Evaluate Corporate SW Improvement Strategies

Recalibrate COCOMO 2.0

Accumulate COCOMO 2.0 calibration data

Cost, Sched, Risks

Milestone plans, resources

Milestone expectations

Yes

Yes

Yes

No

No

No

Yes

Yes

Done?

End
Backup Charts
USC-CSSE Modeling Methodology
- concurrency and feedback implied

1. Determine Model Needs
2. Analyze existing literature
3. Perform Behavioral analyses
4. Define relative significance, data, ratings
5. Perform expert-judgment Delphi assessment, formulate a priori model
6. Gather project data
7. Determine Bayesian A-Posteriori model
8. Gather more data; refine model
Step 6: Gather, Analyze Project Data

• **Best to pilot data collection with early adopters**
  – Identifies data definition ambiguities
  – Identifies data availability problems
  – Identifies need for data conditioning

• **Best to collect initial data via interviews**
  – Avoids misinterpretations
    - Endpoint milestones; activities included/excluded; size definitions
  – Uncovers hidden assumptions
    - Schedule vs. cost minimization; overtime effort reported
Initial Data Analysis May Require Model Revision

- Initial COCOTS model adapted from COCOMO II, with different parameters
  - Effort = A* (Size)^B* \prod (Effort Multipliers)
- Amount of COTS integration glue code used for Size
- Data analysis showed some projects with no glue code, much effort
  - Effort devoted to COTS assessment, tailoring
COCOTS Effort Distribution: 20 Projects

Mean % of Total COTS Effort by Activity (+/- 1 SD)

- Assessment: 49.07% (±7.57%)
- Tailoring: 50.99% (±7.48%)
- Glue Code: 61.25% (±0.88%)
- System Volatility: 20.27% (±11.31%)

% Person-months

-20.00% -10.00% 0.00% 10.00% 20.00% 30.00% 40.00% 50.00% 60.00% 70.00%
Revised COCOTS Model

• COCOMO-like model for glue code effort
• Unit cost approach for COTS assessment effort
  – Number of COTS products to assess
  – Number of attributes to assess, weighted by complexity
• Activity-based approach for COTS tailoring effort
  – COTS parameters setting, script writing, reports layout, GUI tailoring, protocol definitions
New Glue Code Submodel Results

- New calibration results
  - Excluding projects with very large, very small amounts of glue code
    - [0.5 - 100 KLOC]: Pred (.30) = 9/17 = 53%
    - [2 - 100 KLOC]: Pred (.30) = 8/13 = 62%
  - Previous calibration results:
    - [0.1 - 390 KLOC]: Pred (.30) = 4/13 = 31%

- Pred(.30) = percent of projects with estimates within 30% of actuals