
Brad Clark
USC Center for Systems and Software Engineering

Ray Madachy
Naval Postgraduate School

29th International Forum on COCOMO and Systems/Software Cost Modeling
October 22, 2014
Overview

• Forthcoming open access e-book:


• Will be downloadable from
  – softwarecost.org
  – USC Center for Systems and Software Engineering
  – DoD Systems Engineering Research Center (SERC)
  – Other sites
Contributors

• Barry Boehm, University of Southern California
• Bradford Clark, Software Metrics Inc.
• Joseph Dean, Air Force Cost Analysis Agency
• Cheryl Jones, US Army Armament Research Development and Engineering Center
• Raymond Madachy, Naval Postgraduate School
• John McGarry, US Army Armament Research Development and Engineering Center
• Wilson Rosa, Naval Center for Cost Analysis
• Thomas Tan, University of Southern California Los Angeles, California
Acknowledgments

• Numerous workshop participants during the research

• External reviewers
  – Jo Ann Lane, University of Southern California
  – Daniel Ligett, Softstar Systems
  – Daniel Nussbaum, Naval Postgraduate School
  – David Seaver, National Security Agency
  – Richard Selby, Northrop Grumman
  – Daniel Strickland, Missile Defense Agency
  – David Zubrow, Software Engineering Institute

• Cost estimation tool companies providing support
  – Galorath Systems
  – PRICE Systems
  – Quantitative Software Management
  – Softstar Systems
Table of Contents

1 Introduction
2 Software Resources Data Report
3 Metrics Definitions
4 Data Assessment
5 Cost Estimating Relationships
6 Modern Estimating Challenges
7 Estimation Process
Appendix A - Cost Model Descriptions
Appendix B - MIL-STD-881C WBS Mapping
Appendix C - Unified CodeCount
Appendix D – Acronyms
Bibliography
Index
Purpose

• This metrics manual helps analysts and decision makers develop accurate, easy and quick early software cost estimates for different types of systems and operating environments oriented for the United States Department of Defense (DoD) and government agencies in related domains.

• The intent is to improve quality and consistency of early software estimating methods across government cost agencies and program offices through guidance, standardization, and knowledge sharing.

• We have analyzed empirical software cost data for these types of programs and are transitioning the results back in this open access manual.

• Furthermore we describe our processes for data normalization, analysis, derivation of Cost Estimating Relationships (CERs) and productivity benchmarking.
Dataset Summary

• Analysis based on 317 SRDRs reported from recent DoD projects during 2004-2013.
• Normalized Effort vs. Equivalent Size in SRDRs:
Usage Scenarios

• Primary usage for government analysts who are validating and cross-checking software cost estimates provided by contractors.
  – Existing CERs can provide sanity checks without full detailed information on the future programs. Productivity benchmarks are also useful for comparison purposes.
  – In some cases, new CERs need to be developed when existing CERs are inadequate.

• CERs and productivity benchmarks are also important to contractors who provide SRDR data.
  – In most cases, the people responsible for reporting the SRDR are the same preparing the cost proposals for source selection. Publicized CERs and benchmarks will help them crosscheck their cost proposal estimates against government benchmarks.
Cost Estimating Approach

• Collect software cost metrics data
  – The data used in this manual is from DoD’s Software Resource Data Report (SRDR)

• Prepare the data for analysis with normalization
  – This step heavily relies on a set of standard metric definitions
  – Data is then normalized to the standard set of metric definitions

• Create cost estimating relationship (CER) models from the data
  – Segmenting the data into groups
  – Productivity benchmark
  – Formulae with parameters relating size, effort and schedule

• Address challenges in using CERS with modern software development practices.

• Evolve the CERS based on comparing estimates with actual costs on future completed systems.
Estimation and Metrics Processes

Defense Acquisition Management System

A  
Materiel Solution Analysis

B  
Technology Development

C  
Engineering and Manufacturing Development

Production and Deployment  
Operations and Support

Contractor-provided SRDRs

Simple CER Estimates

Multi-Parameter CER Estimates

Program Estimation Process

Initiation and Research  
Assessment  
Analysis  
Documentation and Presentation

Simple CERs  
Multi-Parameter CERs

Software Cost Estimation Metrics Process

Collect Data  
Prepare Data  
Create CERs  
Address Challenges

Evolve CERs

Metrics Definitions  
Cost Model Descriptions
Software Resources Data Report

• Overview
• Collecting Organization
• Repository
• Reporting Frequency
• SRDR Content
• Further SRDR Resources
Metrics Definitions

- Overview
- Product Size Measures
- SLOC Count Definitions
- ESLOC Summary
- Functional Size Measures
- Development Effort
- Development Duration
Metrics Definitions Examples

• Software Size Types

<table>
<thead>
<tr>
<th>Size Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>New software created for the first time.</td>
</tr>
<tr>
<td>Adapted</td>
<td>Pre-existing software that is used as-is (Reused) or changed (Modified).</td>
</tr>
<tr>
<td>Reused</td>
<td>Pre-existing software that is not changed with the adaption parameter settings:</td>
</tr>
<tr>
<td></td>
<td>• Design Modification % (DM) = 0%</td>
</tr>
<tr>
<td></td>
<td>• Code Modification % (CM) = 0%.</td>
</tr>
<tr>
<td>Modified</td>
<td>Pre-existing software that is modified for use by making design, code and / or test changes:</td>
</tr>
<tr>
<td></td>
<td>• Code Modification % (CM) &gt; 0%.</td>
</tr>
</tbody>
</table>

• SLOC Count Definitions

<table>
<thead>
<tr>
<th>Statement Type</th>
<th>Logical</th>
<th>NCSS</th>
<th>Physical</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>In</td>
<td>Out</td>
<td>In</td>
<td>Out</td>
</tr>
<tr>
<td>Executable</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Nonexecutable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Declarations</td>
</tr>
<tr>
<td>Compiler directives</td>
</tr>
<tr>
<td>Comments</td>
</tr>
<tr>
<td>Blank lines</td>
</tr>
</tbody>
</table>
Data Assessment

• Gather Collected Data
• Inspect each Data Record
• Determine Data Quality Levels
• Correct Missing or Questionable Data
• Normalize Size and Effort Data
• Convert Raw SLOC into Equivalent SLOC
Data Assessment Examples

- **Data Quality Rating Scale**

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>1</td>
<td>if size data present</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>if no size data</td>
</tr>
<tr>
<td>Size Count Type</td>
<td>1</td>
<td>if size is Logical SLOC</td>
</tr>
<tr>
<td></td>
<td>0.7</td>
<td>if size is Non-Commented Source Statements</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>if size is Physical Lines (Comment and Source Statements)</td>
</tr>
<tr>
<td></td>
<td>0.4</td>
<td>if size is Total Lines (all lines in file: blank, comment, source)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>if no size data</td>
</tr>
<tr>
<td>ESLOC Factors:</td>
<td>1</td>
<td>if modification factors are provided for Auto-gen, Modified &amp; Reuse code counts from out-</td>
</tr>
<tr>
<td>(See Section 3.2.2.2)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Proxy DM, CM and IM Values**

<table>
<thead>
<tr>
<th>PT</th>
<th>Code Type</th>
<th>#</th>
<th>DM</th>
<th>CM</th>
<th>IM</th>
<th>AAF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Mean</td>
<td>Median</td>
<td>Mean</td>
<td>Median</td>
</tr>
<tr>
<td>SCP</td>
<td>Auto-Gen</td>
<td>0</td>
<td>0 ± 0</td>
<td>0</td>
<td>0 ± 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reused</td>
<td>18</td>
<td>0.51 ± 0.21</td>
<td>0.42</td>
<td>0.66 ± 0.4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Modified</td>
<td>7</td>
<td>0.66 ± 0.4</td>
<td>1</td>
<td>0.66 ± 0.4</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Auto-Gen</td>
<td>0</td>
<td>0 ± 0</td>
<td>0</td>
<td>0 ± 0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Reused</td>
<td>8</td>
<td>0.17 ± 0.23</td>
<td>0.1</td>
<td>0.95 ± 0.07</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Modified</td>
<td>14</td>
<td>0.95 ± 0.07</td>
<td>1</td>
<td>0.95 ± 0.07</td>
<td>1</td>
</tr>
</tbody>
</table>
Cost Estimating Relationships

• Overview
• Data Segmentation
• SRDR Metric Definitions
• Cost Estimating Relationships (CERs)
• Software CERs by AppType Across All Environments
• Productivity Benchmarks
Data Analysis Objectives

• Make collected data useful to oversight and management entities
  – Provide guidance on how to condition data to address challenges
  – Segment data into different Application Types and Operating Environments
  – Analyze data for simple Cost Estimating Relationships (CER) within each domain
  – Develop rules-of-thumb for missing data
Operating Environment

- Represents the platform that the software operates in.
- Complexity of a software also driven by environment
- It is important to determine the appropriate environment before analyzing your software project:

<table>
<thead>
<tr>
<th>Environment</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Ground</td>
<td>Fixed Ground Site (MGS)</td>
</tr>
<tr>
<td>Ground Surface Vehicles</td>
<td>Manned (MGV)</td>
</tr>
<tr>
<td></td>
<td>Unmanned (UGV)</td>
</tr>
<tr>
<td>Sea Systems</td>
<td>Manned (MSV)</td>
</tr>
<tr>
<td></td>
<td>Unmanned (USV)</td>
</tr>
<tr>
<td>Aircraft</td>
<td>Manned (MAV)</td>
</tr>
<tr>
<td></td>
<td>Unmanned (UAV)</td>
</tr>
<tr>
<td>Missile / Ordnance (M/O)</td>
<td>Unmanned (UOM)</td>
</tr>
<tr>
<td>Spacecraft</td>
<td>Manned (MSC)</td>
</tr>
<tr>
<td></td>
<td>Unmanned (USC)</td>
</tr>
</tbody>
</table>
Application Type

- Complexity is influenced by its Application Type
- Application Types are groups of application domains that are environment independent, technology driven and characterized by:

1. Required software reliability
2. Data processing requirements
3. Product complexity
4. Integration complexity
5. Real-time operating requirements
6. Platform volatility
7. Target system volatility
8. Special display requirements
9. Development re-hosting
10. Quality assurance requirements
11. Security requirements
12. Assurance requirements
13. Required testing level

- Determining the appropriate productivity type is critical as it allows you to select the most appropriate model and benchmark.
Application Types

- Sensor Control and Signal Processing (SCP)
- Vehicle Control (VC)
- Vehicle Payload (VP)
- Real Time Embedded (RTE)
- Mission Processing (MP)
- Process Control (PC)
- System Software (SYS)
- Planning Software (PLN)

- Scientific Software (SCI)
- Training Software (TRN)
- Telecommunications (TEL)
- Software Tools (TOOL)
- Test Software (TST)
- Intelligence & Information Software (IIS)
Example CER

- Intelligence and Information Software (IIS) CER:

\[ PM = 1.27 \cdot KESLOC^{1.18} \]
CER Examples

• CERs by Application Types

<table>
<thead>
<tr>
<th>AppType</th>
<th>Effort Equation</th>
<th>N</th>
<th>R²</th>
<th>MAD</th>
<th>PRED</th>
<th>EKSLOC Min</th>
<th>EKSLOC Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCP</td>
<td>( PM = 74.37 + KESLOC^{1.71} )</td>
<td>36</td>
<td>***</td>
<td>.69</td>
<td>.31</td>
<td>1</td>
<td>162</td>
</tr>
<tr>
<td>VP</td>
<td>( PM = 3.15 \cdot KESLOC^{1.38} )</td>
<td>16</td>
<td>***</td>
<td>.27</td>
<td>.5</td>
<td>5</td>
<td>120</td>
</tr>
<tr>
<td>RTE</td>
<td>( PM = 34.32 + KESLOC^{1.52} )</td>
<td>52</td>
<td>***</td>
<td>.61</td>
<td>.46</td>
<td>1</td>
<td>167</td>
</tr>
<tr>
<td>MP</td>
<td>( PM = 3.48 \cdot KESLOC^{1.17} )</td>
<td>48</td>
<td>.88</td>
<td>.49</td>
<td>.58</td>
<td>1</td>
<td>207</td>
</tr>
<tr>
<td>SYS</td>
<td>( PM = 16.01 + KESLOC^{1.37} )</td>
<td>60</td>
<td>***</td>
<td>.37</td>
<td>.53</td>
<td>2</td>
<td>215</td>
</tr>
<tr>
<td>SCI</td>
<td>( PM = 21.09 + KESLOC^{1.36} )</td>
<td>39</td>
<td>***</td>
<td>.65</td>
<td>.18</td>
<td>1</td>
<td>171</td>
</tr>
<tr>
<td>IIS</td>
<td>( PM = 1.27 \cdot KESLOC^{1.18} )</td>
<td>37</td>
<td>.90</td>
<td>.35</td>
<td>.65</td>
<td>1</td>
<td>180</td>
</tr>
</tbody>
</table>

• Productivity benchmarks by Operating Environment

<table>
<thead>
<tr>
<th>OpEnv</th>
<th>N</th>
<th>Min</th>
<th>Max</th>
<th>LCI</th>
<th>Mean</th>
<th>UCI</th>
<th>SD</th>
<th>CV</th>
<th>Q1</th>
<th>Med.</th>
<th>Q3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVU</td>
<td>6</td>
<td>5</td>
<td>134</td>
<td>50</td>
<td>85</td>
<td>120</td>
<td>44</td>
<td>52%</td>
<td>52</td>
<td>76</td>
<td>125</td>
</tr>
<tr>
<td>OVU</td>
<td>16</td>
<td>0.3</td>
<td>189</td>
<td>88</td>
<td>133</td>
<td>177</td>
<td>90</td>
<td>68%</td>
<td>51</td>
<td>120</td>
<td>226</td>
</tr>
<tr>
<td>AVM</td>
<td>50</td>
<td>0.32</td>
<td>208</td>
<td>108</td>
<td>133</td>
<td>158</td>
<td>91</td>
<td>68%</td>
<td>81</td>
<td>129</td>
<td>184</td>
</tr>
<tr>
<td>MVM</td>
<td>67</td>
<td>2</td>
<td>54</td>
<td>159</td>
<td>203</td>
<td>247</td>
<td>184</td>
<td>91%</td>
<td>112</td>
<td>193</td>
<td>717</td>
</tr>
<tr>
<td>GSF</td>
<td>116</td>
<td>0.61</td>
<td>215</td>
<td>182</td>
<td>205</td>
<td>227</td>
<td>123</td>
<td>60%</td>
<td>118</td>
<td>229</td>
<td>301</td>
</tr>
</tbody>
</table>
Dataset Productivity Distribution

![Histogram of Dataset Productivity Distribution](image)
Example CER Uncertainty Analysis

- IIS CER prediction errors and Standard Error Variation
Modern Estimating Challenges

• Rapid Change, Emergent Requirements, and Evolutionary Development
• Net-centric Systems of Systems (NCSoS)
• Model-Driven and Non-Developmental Item (NDI)-Intensive Development.
• Ultrahigh Software Systems Assurance
• Legacy Maintenance and Brownfield Development
• Agile and Kanban Development.
• Putting It All Together at the Large-Project or Enterprise-Level
Estimation Process

- Overview
- Estimation Purpose
- Program Definition
- Estimation Scope
- Data Collection and Normalization
- Estimate Creation
- Sensitivity Analysis
- Risk and Uncertainty Analysis
- Estimate Documentation and Packaging
Estimation Process based on GAO

Initiation & Research

Define the estimate's purpose
Develop the estimating plan

Assessment

Define the program
Determine the estimating structure
Identify ground rules and assumptions
Obtain the data
Develop the point estimate and compare to an independent estimate

Documentation & Presentation

Update estimate to reflect actual costs / changes
Present estimate to management for approval
Document the estimate

Analysis

Conduct risk and uncertainty analysis
Conduct sensitivity analysis

Appendix A - Cost Model Descriptions

• Introduction
• Cost Models
  – COCOMO II
  – True Planning
  – SEER-SEM
  – SLIM
• Cost Model Input Factors
  – Software Size
  – Software Cost Drivers
• Cost Model Lifecycles and Work Breakdown Structures

<table>
<thead>
<tr>
<th>COCOMO II</th>
<th>SEER-SEM</th>
<th>True Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCALE DRIVERS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Precededness</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>Development Flexibility</td>
<td>none</td>
<td>Development Team Complexity</td>
</tr>
<tr>
<td>Architecture/Risk Resolution</td>
<td>none</td>
<td></td>
</tr>
<tr>
<td>Team Cohesion</td>
<td>none</td>
<td>Organization Productivity</td>
</tr>
<tr>
<td>Process Maturity</td>
<td>none 1</td>
<td>- CMM Level</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODUCT ATTRIBUTES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required Software Reliability</td>
<td>Specification Level - Reliability</td>
<td>Operating Specification</td>
</tr>
<tr>
<td>Data Base Size</td>
<td>none</td>
<td>Code Size non Executable</td>
</tr>
<tr>
<td>Product Complexity</td>
<td>- Complexity (Staffing) - Application Class Complexity</td>
<td>Functional Complexity</td>
</tr>
<tr>
<td>Required Reusability</td>
<td>- Reusability Level Required - Software Impacted by Reuse</td>
<td>Design for Reuse</td>
</tr>
<tr>
<td>Documentation Match to Lifecycle Needs</td>
<td>none</td>
<td>Operating Specification</td>
</tr>
</tbody>
</table>
Future Work

• More SRDR data is collected each year. The SRDR data requirements are periodically revised to collect more and higher quality data.
  – The intent is to keep this manual relevant with future editions incorporating new information from contributors.

• Much future work is identified in the current manual.

• This manual will be hosted and maintained at http://softwarecost.org
  – Readers can find errata, updates to its content, and a place to submit suggestions and comments.