Impact of Scope Changes on Software Growth

Dr. Jon Brown
Gail Flynn

Naval Surface Warfare Center, Dahlgren Division
Cost Engineering and Analysis

June 2018
Agenda

• Background
• Software Growth Defined
• Analysis Methods and Results
• Model Description and Results
• Summary and Q&A
Software Growth

- Survey of recent studies measuring software growth
- Most calculated growth using initial and final reported source lines of code (SLOC) or equivalent SLOC (ESLOC)
- Method captures total growth including any growth owing to scope increases

**What is the magnitude of the impact of scope growth on reported software growth?**

**Sources:**
- Average of SRDR Data Compilation Pairs, dated 16 OCT 2017
- ICEAA June 2015 NCCA Software Growth Analysis (SW15) - Logical SLOC only
- SEI DoD SW Factbook, 2017 (CMU/SEI-2017-TR-004)
Software Growth Example

Initial Software Size

400K

400K ESLOC

Final Software Size

800K

800K ESLOC

100% Growth

Direct comparison of final to initial ESLOC includes all sources of growth if not adjusted
Software Growth Example (cont’d)

<table>
<thead>
<tr>
<th>Initial Software Size</th>
<th>Final Software Size</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Initial Scope</strong></td>
<td><strong>Final Scope</strong></td>
</tr>
<tr>
<td>400K ESLOC</td>
<td>800K ESLOC</td>
</tr>
<tr>
<td>400K</td>
<td>500K</td>
</tr>
<tr>
<td><strong>25% Growth</strong></td>
<td><strong>New Scope</strong></td>
</tr>
<tr>
<td>300K</td>
<td>300K</td>
</tr>
</tbody>
</table>

- Adjusting for scope growth would give a truer picture of the actual growth of initial software scope.
- Requires information not currently captured in SRDRs.

To differentiate between the two growth metrics, we need to define some terms.
## Software Size Growth

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Underestimating required SLOC</td>
<td>Size projection errors</td>
<td>Underestimating the amount of new SLOC</td>
</tr>
<tr>
<td>Poor understanding of initial requirements</td>
<td>Requirements volatility</td>
<td>Underestimating the software complexity</td>
</tr>
<tr>
<td>Code reuse optimism</td>
<td>Product functionality changes</td>
<td>Overestimating the expected use of existing SLOC, i.e. modified and unmodified SLOC</td>
</tr>
<tr>
<td>New requirements added during development</td>
<td>Human errors</td>
<td></td>
</tr>
</tbody>
</table>


### Software Size Growth (cont’d)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Underestimating required SLOC</td>
<td>Size projection errors</td>
<td>Underestimating the amount of new SLOC</td>
</tr>
<tr>
<td>Poor understanding of initial requirements</td>
<td>Requirements volatility</td>
<td>Underestimating the software complexity</td>
</tr>
<tr>
<td>Code reuse optimism</td>
<td>Product functionality changes</td>
<td>Overestimating the expected use of existing SLOC, i.e. modified and unmodified SLOC</td>
</tr>
<tr>
<td>New requirements added during development</td>
<td>Human errors</td>
<td></td>
</tr>
</tbody>
</table>

*2007 Software Code Growth**
Software Size Growth (cont’d)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Underestimating required SLOC</td>
<td>Size projection errors</td>
<td>Underestimating the amount of new SLOC</td>
</tr>
<tr>
<td>Poor understanding of initial requirements</td>
<td>Requirements volatility</td>
<td>Underestimating the software complexity</td>
</tr>
<tr>
<td>Code reuse optimism</td>
<td>Product functionality changes</td>
<td>Overestimating the expected use of existing SLOC, i.e. modified and unmodified SLOC</td>
</tr>
<tr>
<td>New requirements added during development</td>
<td>Human errors</td>
<td></td>
</tr>
</tbody>
</table>

**Presented at the 2018 ICEAA Professional Development & Training Workshop - www.iceaaonline.com**

- **Software Growth Definition:**
  - Underestimating required SLOC
  - Poor understanding of initial requirements
  - Code reuse optimism
  - New requirements added during development
Definition of Pure Software Growth

Total Growth

Pure Growth + Scope Growth

- Underestimating required SLOC
- Poor understanding of initial requirements
- Code reuse optimism

- New requirements added during development

Completely unrelated scope additions should be estimated separately and adjusted for in historical data.

Software Growth Definition:
- Underestimating required SLOC
- Poor understanding of initial requirements
- Code reuse optimism
- New requirements added during development
Examples of Pure vs Total Growth

- Four large DoD software programs were selected based on relevance and for availability of data.
- Scope changes were determined using data outside available SRDRs, which included:
  - Monthly or quarterly ESLOC reports
  - Systems Engineering Technical Review briefs
  - Program schedules
  - Software metric reports
  - Identified and interviewed subject-matter experts when possible to validate interpretations of data.
Pure vs Total Growth Program 1

Program Description

- Real time
- Command and control
- Combat Management System (CMS) upgrade
- Software program: ~5000K DSLOC
Pure vs Total Growth Program 1 (cont’d)

Program Description

- Real time
- Command and control
- Combat Management System (CMS) upgrade
- Software program: ~5000K DSLOC

Initial vs. Final Total Software Growth = 79%
Pure vs Total Growth Program 1 (cont’d)

Program Description
- Real time
- Command and control
- Combat Management System (CMS) upgrade
- Software program: ~5000K DSLOC

Large Scope Added = 140K ESLOC

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pure Growth</td>
<td>28%</td>
</tr>
<tr>
<td>Scope Growth</td>
<td>51%</td>
</tr>
<tr>
<td>Total Growth</td>
<td>79%</td>
</tr>
</tbody>
</table>
Pure vs Total Growth Program 2

![Graph showing CDR, IPR, and PDR with an unknown event peak]
Pure vs Total Growth Program 2 (cont’d)

Initial vs. Final Total Software Growth = 20%

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC
Pure vs Total Growth Program 2 (cont’d)

Program Description

- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC

Anomalies: No Obvious Scope Added

| Pure Growth | 20% |
| Scope Growth | 0% |
| Total Growth  | 20% |
Pure vs Total Growth Program 3

Program Description

- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC
Pure vs Total Growth Program 3 (cont’d)

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC

Initial vs. Final Total Software Growth = 61%
Pure vs Total Growth Program 3 (cont’d)

Program Description

- Real time
- Command and control
- CMS upgrade
- Software program: ~4000K DSLOC

Scope Added = 340K ESLOC

Pure Growth  24%
Scope Growth  37%
Total Growth  61%
Pure vs Total Growth Program 4

Program Description

- Real time
- Command and control
- CMS upgrade
- Software program: ~2000K DSLOC

Presented at the 2018 ICEAA Professional Development & Training Workshop - www.iceaaonline.com
Pure vs Total Growth Program 4 (cont’d)

Program Description

- Real time
- Command and control
- CMS upgrade
- Software program: ~2000K DSLOC

Initial vs. Final Total Software Growth = 64%
Pure vs Total Growth Program 4 (cont’d)

Program Description
- Real time
- Command and control
- CMS upgrade
- Software program: ~2000K DSLOC

Pure Growth 46%
Scope Growth 18%
Total Growth 64%

Scope Added = 50K ESLOC
Pure vs Total Growth Comparison

Scope growth is likely a large contributor to total software growth. What is the magnitude of this difference on software cost estimates?
Example Software Development Model: Fixed Method

**Cost Model**

- **Design, Code, Test and Integration (DCTI)**
  - \[ \text{DCTI} = \text{ESLOC} \times (1 + \text{SW Growth}) \times \text{Labor Rate} \]

- **Non DCTI**
  - \[ \text{Non DCTI} = \text{FTEs} \times \text{Labor rate} \]

- **Government**
  - \[ \text{Government} = \text{FTEs} \times \text{Labor rate} \]

- **Software Development Cost Estimate**
  - **Fixed Method:** FTE-Based

**Formulas**

- ESLOC
- SW Growth
- Productivity Rate
- Labor rate
- FTEs
- Non DCTI
- Government

**Presentation Details**

- Presented at the 2018 ICEAA Professional Development & Training Workshop - www.iceaaonline.com
Example Software Development Model: Variable Method

Design, Code, Test and Integration (DCTI)

= ESLOC \times (1 + \text{SW Growth}) \times \text{Labor Rate} \times \text{Productivity Rate}

Non DCTI

= CER \times (\text{DCTI})

Government

= CER \times (\text{Non DCTI + DCTI})

Cost Model

Software Development Cost Estimate

Variable Method: Functionally Correlated
Impact of Pure Growth on Model Results

Given the large impact on a software development estimate, documenting whether pure or total growth is used is critical.

Using total vs. pure software growth can result in 15–40% difference in software development cost.
Impact of Pure Growth on Uncertainty

Risk applied to variable method, with 50% probability

<table>
<thead>
<tr>
<th>Pure Growth</th>
<th>Total Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>A (min)</td>
</tr>
<tr>
<td>Most Likely</td>
<td>B</td>
</tr>
<tr>
<td>High</td>
<td>C</td>
</tr>
</tbody>
</table>

Some Options for Risk Distribution

Option 5 = Option 1 + Risk Event: Scope Increase
Impact of Pure Growth on Uncertainty (cont’d)

Your choice of pure or total software growth and risk boundaries will impact your estimate and should be documented.

Risk applied to variable method, with 50% probability

<table>
<thead>
<tr>
<th>Pure Growth</th>
<th>Total Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>A (min)</td>
</tr>
<tr>
<td>Most Likely</td>
<td>B</td>
</tr>
<tr>
<td>High</td>
<td>C</td>
</tr>
</tbody>
</table>

Some Options for Risk Distribution

Option 5 = + Risk Event: Scope Increase

SW Development Cost Estimate ($M)

5% 30%
Summary

- Pure growth + Scope growth = Total Growth

- Initial vs final comparisons of ESLOC measure total software growth
  - Examples demonstrate that scope growth likely contributes a large amount to total software growth and to variance in the historical dataset

- The choice of pure vs total software growth can impact your software development model 15–40%
  - Given the impact, it is crucial to document your assumption on what is included
  - Using total software growth without adjustment is equivalent to assuming estimate includes software scope growth

- The choice of risk boundaries will impact your software development estimate
  - It is essential to document your risk boundaries and assumptions to support them.
Conclusion

Questions, Answers, and Discussion

NSWCDD V11
Cost & Schedule Engineering & Analysis Branch
Dahlgren Virginia

Dr. Jon Brown
(540) 653-9461
jonathan.d.brown@navy.mil

Gail Flynn
(540) 653-3316
gail.flynn@navy.mil

Presented at the 2018 ICEAA Professional Development & Training Workshop - www.iceaaonline.com
## Model Inputs

### DCTI Inputs

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESLOC</td>
<td>50000 or otherwise</td>
</tr>
<tr>
<td>ESLOC Total Growth</td>
<td>0.79</td>
</tr>
<tr>
<td>ESLOC Pure Growth</td>
<td>0.28</td>
</tr>
<tr>
<td>Productivity Rate (ESLOC/hr)</td>
<td>1.2</td>
</tr>
</tbody>
</table>

### Non-DCTI Inputs

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contractor Fixed (FTEs)</td>
<td>40 or otherwise</td>
</tr>
<tr>
<td>Contractor Variable</td>
<td>DCTI$</td>
</tr>
</tbody>
</table>

### Government Inputs

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Government Fixed</td>
<td>0.25*Fixed Non DCTI</td>
</tr>
<tr>
<td>Government Variable</td>
<td>0.25 * Total Contractor estimate</td>
</tr>
</tbody>
</table>

### Other Inputs

<table>
<thead>
<tr>
<th>Component</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labor rate ($/hr)</td>
<td>200</td>
</tr>
<tr>
<td>Contractor hrs/year</td>
<td>1872</td>
</tr>
<tr>
<td>Government hrs/year</td>
<td>1800</td>
</tr>
</tbody>
</table>

### Risk Inputs

<table>
<thead>
<tr>
<th>Option</th>
<th>Low</th>
<th>PE</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-Pure, PE-Pure, H-Pure</td>
<td>0</td>
<td>0.28</td>
<td>0.56</td>
</tr>
<tr>
<td>L-Pure, PE-Pure, H-Total</td>
<td>0</td>
<td>0.28</td>
<td>1.58</td>
</tr>
<tr>
<td>L-Pure, PE-Total, H-Total</td>
<td>0</td>
<td>0.79</td>
<td>1.58</td>
</tr>
<tr>
<td>L-Total, PE-Total, H-Total</td>
<td>0.1</td>
<td>0.79</td>
<td>1.58</td>
</tr>
</tbody>
</table>