Software Development Cost Modeling and Estimation through a UNSW Lens

Ross Jeffery
UNSW Creating Technology

- Microkernel-based embedded operating systems
  - Used in QUALCOMM chipsets – approaching 1 billion deployments
David Weiss, when drawing on the work of Thomas Kuhn, notes, “for software engineering to become a discipline we must know…what the critical variables in software development are, and how to use the results of our measurements to improve our development processes” [25]. The work of Barry Boehm has been seminal in the continuously evolving software development paradigm shift referred to by Weiss in this article, particularly in the areas of software development cost modeling and estimation.
“The research of Louis Pasteur, whose studies of bacteriology were carried out at the behest of the French wine industry, characterizes the work of scientists who, like Bohr, search for fundamental knowledge, but unlike Bohr, select their questions and methods based on potential relevance to real world problems.”

Contents

1. Introduction

2. Early Cost Modeling

3. Software Engineering Economics

4. Elapsed Time Compression

5. Simulation and calibration

6. Systems Estimation

7. Conclusions
2. The Early Days in Cost Modeling

- Programming productivity
  Nelson, Johnson, Chrysler, Lawrence
- Modeling – productivity/size relationship
- Issues:
  Lack of consistent definition of variables
  Absence of social and organizational factors
  Absence of program factors like complexity, quality, documentation completeness, etc.
### “Peopleware” – DeMarco & Lister (1987)

<table>
<thead>
<tr>
<th>Effort estimate prepared by:</th>
<th>Average productivity</th>
<th>Number of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmer alone</td>
<td>8.0</td>
<td>19</td>
</tr>
<tr>
<td>Supervisor alone</td>
<td>6.6</td>
<td>23</td>
</tr>
<tr>
<td>Programmer and supervisor</td>
<td>7.8</td>
<td>16</td>
</tr>
</tbody>
</table>

“Peopleware” – DeMarco & Lister (1987)

<table>
<thead>
<tr>
<th>Effort estimate prepared by:</th>
<th>Average productivity</th>
<th>Number of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmer alone</td>
<td>8.0</td>
<td>19</td>
</tr>
<tr>
<td>Supervisor alone</td>
<td>6.6</td>
<td>23</td>
</tr>
<tr>
<td>Programmer and supervisor</td>
<td>7.8</td>
<td>16</td>
</tr>
<tr>
<td>Systems Analyst</td>
<td>9.5</td>
<td>21</td>
</tr>
</tbody>
</table>

“Peopleware” – DeMarco & Lister (1987)

<table>
<thead>
<tr>
<th>Effort estimate prepared by:</th>
<th>Average productivity</th>
<th>Number of projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmer alone</td>
<td>8.0</td>
<td>19</td>
</tr>
<tr>
<td>Supervisor alone</td>
<td>6.6</td>
<td>23</td>
</tr>
<tr>
<td>Programmer and supervisor</td>
<td>7.8</td>
<td>16</td>
</tr>
<tr>
<td>Systems Analyst</td>
<td>9.5</td>
<td>21</td>
</tr>
<tr>
<td>(No estimate)</td>
<td>12.0</td>
<td>24</td>
</tr>
</tbody>
</table>

3. And then: Software Engineering Economics

• Variable definition
• Program factors
• Organizational factors
• Integrated software support environment
4. Schedule Compression

- COCOMO 1981
- RCA 1978
- Putnam 1978

and others……

- Some model inconsistencies, uncertain data support, Brooks law robustness.
Case 1. Avoiding Brooks law

SCATS is a computer system that monitors and controls traffic signals in real-time.

SCATS is an acronym for Sydney Co-ordinated Adaptive Traffic System. It was first developed by the New South Wales Roads and Traffic Authority (RTA).

The system is used in more than 80 cities around the world.
Empirical Observations

• The background – Brooks Law (adding more people to a late project will make it even later)

• Empirical observation - The RTA was an exception to Brooks law (and approx. 20% higher productivity than any other organization of the >30 studied)– why?

• Empirical observation - The study of 47 software projects showed elapsed time compression and extension with both increased and reduced productivity. Another exception to Brooks Law - why?
Case 2 - Data from 47 industry projects concerning schedule and effort  (Jeffery - IEEE TSE)

Effort ratio
Actual/expected

Elapsed time ratio
Actual/expected

Elapsed time compression  ↔  Elapsed time expansion

Elapsed time ratio
Actual/expected

1.0  ↔  2.1

Effort ratio
Actual/expected

4.2  ↔  1.0
Compression models and data

![Graph showing effort and elapsed time ratios for different models (RCA, Putnam, COCOMO).]
5. Theory development - Staffing model, size and productivity relationships (He Zhang - Jason)

Qualitative abstract structure of staffing process
Results

• A theoretical model using modeling, behavior trees, and simulation show that the empirically observed behavior is possible.

• The model can then be used to plan and predict the effects on schedule and effort of project staffing patterns.

• In practice - “Information hiding” in the form of staff quarantine was used at the RTA. The key variable is staff numbers and how they can be deployed. Prerequisite – architectural design that allows staff quarantine.

• A case of observations, laws and theory.
Observations, laws and theories

- **Observations** – facts or impressions, they tell us “what”.
- **Laws** – repeatable observations, they tell us “how”.
- **Theories** – Explain and order or observations, they tell us “why”.  
  (Endres and Rombach, 2003)
Calibration

• Lo, Webby and Jeffery - a high correlation between effort and the numbers of different types of widgets ($R^2 = 0.863$, $p < 0.001$).

• But we also found that the model exhibited a high error level when applied in a different development environment without calibration.

• 1990 - The need for clear unambiguous metrics definition was highlighted and the necessity of model calibration for the MIS context investigated was shown.
Calibration

• 2001 – with Ruhe and Wieczorek
• All of these results provide strong validation for the argument made in S. E. Economics for calibration of cost models to the local environment and also calibration to account for different definitions used for terms such as “source instructions", “development" and so on
Importance of calibration

Turning Point

Increasing returns to scale

Decreasing returns to scale

Effort

Size
Case 3: Lucent technologies 5ESS Switch Code

- The setting – 5ESS switching software.
- Similar success with inspections as NASA.
Empirical results

- Practically all of the discovered defects were found by two/three people from the large team.

- A finding supported by inspection experiments conducted using student subjects in Australia.

- These experiments also showed the positive impact of product decomposition strategies on inspection performance.
Size measurement – function points

• Consistent a priori size measure
• Counter issue
• Counter experience in application domain
• Counting rules
Data Analysis – Keung et al.

• The literature exhibits a high degree of instability in its findings.
• Findings that (1) many of the data sets used in prior work are poor at differentiating between different estimators and therefore studies comparing different techniques need to make use of the stronger datasets, (2) neural nets and simple linear regression performed worse than analogy as learners, and (3) data preprocessing has a significant impact on learner effectiveness.
Some industry comment

• “For 90% of projects we can estimate effort within 10% of actual. But it's the other 10% of projects that cause us problems”

• For the 90% of projects in a data rich, stable organization with stable application requirements and stable development environments, cost modeling and cost estimation should not be a significant issue. The question today is: “How common is this scenario?”
Returning to the bigger picture

Slower development cycles
Waterfall methodologies

Focus on cycle time
RAD, Spiral.....

Agility in development management – but lots has changed
An Industrial observation

• An American corporation employing over 140,000 people in total.
• “We're doing some cool things -- advanced agile/lean methodologies on the process side and we're starting to do a lot of mobile development on the product side.

We're using all the Atlassian tools and we do a lot of test automation (unit, behavioral, load test) to the point where we're trying to get to a "continuous delivery" process ... the notion that within 8 hours after check in, a piece of code would have been fully vetted automatically in a QA environment and ready for production deployment.”

Email communication July 2010
The state of practice

• **Management**
  Agile – penetration of SCRUM as a flexible project management method. Kanban, Pairing, etc....

• **Technology** support for development. Supported peer review of code and other documents. Test support and automation. Tools support for continuous integration. New deployment technologies

• **Reuse**
  Component assets, open source, and architecture supported services.
State of the Practice

• We now work on complex systems of systems; we deploy systems on the cloud; and we deliver very large applications in embedded systems.
State of the art

- Component development
- Services orientation
- Automated testing and continuous integration
- Agile development teams
- Metrics collection and reporting
- Tools and technologies rapidly changing
- Adaptive systems
- **Systems of Systems**
- Cloud
- Social Networks
- Internet of things
Defence Concerns

Table 1  Projects of Concern February 1st 2011

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN10</td>
<td>Collins Class Submarine Sustainment and Projects</td>
</tr>
<tr>
<td>JOINT 2043 Phase 3A</td>
<td>High Frequency Modernisation (HFMOD) - communications and data exchange capability for sea, air and land forces</td>
</tr>
<tr>
<td>AIR 5333</td>
<td>‘Vigilare’ - Aerospace surveillance and command and control system</td>
</tr>
<tr>
<td>AIR 5276 Phase 8B</td>
<td>Electronic Support Measures upgrade for AP-3C Orion aircraft</td>
</tr>
</tbody>
</table>

The response:

Defence “Systems of Systems Integration – Capability Scoping Study” 2011
The Way Forward

• In part it will be technologies like COSYSMO:

which estimates the “quantity of systems engineering labor required for conceptualizing, designing, testing, and deploying large-scale software and hardware projects”.

NICTA Copyright 2011
From imagination to impact
7. Conclusions

1. The research space in software and systems process is too large for broad scale solely academic research inroads. (And the PhD-only model is limited).
2. The evidence on impact suggests that the historical lead times for research to practice are now too long.
3. Successful impact has been achieved more often in industry/academic collaborative research into technologies. These may be process technologies.
4. There is evidence that “laws” of process can be found in industry/academic collaborative research.
5. The state-of-the-art can be achieved in industrial practice through collaborative use-inspired research.
Conclusions

• “Just as it was in 1981, Barry Boehm continues to lead the field in the area of cost modeling and cost estimation, for software, systems, and systems of systems.”
Questions?
<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN10</td>
<td>Collins Class Submarine Sustainment and Projects</td>
</tr>
<tr>
<td>AIR 5077</td>
<td>‘Wedgetail’ Airborne Early Warning and Control aircraft</td>
</tr>
<tr>
<td>Phase 3</td>
<td></td>
</tr>
<tr>
<td>SEA 1448</td>
<td>Anti-Ship Missile Defence radar upgrades for ANZAC Class Frigates</td>
</tr>
<tr>
<td>Phase 2B</td>
<td></td>
</tr>
<tr>
<td>JOINT 2043</td>
<td>High Frequency Modernisation (HFMOD) – communications and data exchange capability for sea, air and land forces</td>
</tr>
<tr>
<td>Phase 3A</td>
<td></td>
</tr>
<tr>
<td>AIR 5333</td>
<td>‘Vigilare’ – Aerospace surveillance and command and control system</td>
</tr>
<tr>
<td>JOINT 129</td>
<td>‘Tactical Unmanned Aerial Vehicles – airborne surveillance for land forces</td>
</tr>
<tr>
<td>Phase 2</td>
<td></td>
</tr>
<tr>
<td>LAND 121</td>
<td>‘Overlander’ replacement field vehicles, trailers and modules for land</td>
</tr>
<tr>
<td>Phase 3</td>
<td>forces (‘Medium Heavy’ class of vehicles only)</td>
</tr>
<tr>
<td>JOINT 2070</td>
<td>Lightweight torpedo replacement for ANZAC and ADELAIDE Class Frigates</td>
</tr>
<tr>
<td>AIR 5402</td>
<td>Multi-Role Tanker Transport aircraft – Air to Air Refuelling Capability</td>
</tr>
<tr>
<td>AIR 5276</td>
<td>Electronic Support Measures upgrade for AP-3C Orion aircraft</td>
</tr>
<tr>
<td>Phase 8B</td>
<td></td>
</tr>
<tr>
<td>AIR 5418</td>
<td>Joint Air to Surface Standoff Missiles</td>
</tr>
<tr>
<td>Phase 1</td>
<td></td>
</tr>
</tbody>
</table>

Projects of Concern February 1\(^{st}\) 2011