POST DEPLOYMENT SOFTWARE SUPPORT ESTIMATING METHODS

Presented to:
11th International Forum on COCOMO and Software Cost Modeling
9-11 October 1996

Presented by:
David Patterson
Sherry Stukes
Henry Appgar
Management Consulting & Research, Inc.
4165 E. Thousand Oaks Blvd., Suite 235
Thousand Oaks CA 91362-3810

OUTLINE

• Background
• Definition
• Commercial Models Comparison
• Software Life Cycle Cost Drivers
• Software Support Metrics
• Conclusions
BACKGROUND

- Current DoD systems are software intensive
- Emphasis is placed on estimating software development cost
- Software maintenance can represent 75% of the total software life cycle cost
- Difficult to estimate due to:
  - the uncertainty of maintenance environment
  - lack of calibrated software maintenance estimating models

DEFINITION

Government
Post Deployment Software Support (PDSS): The total cost of maintaining a software program for its entire life once it has been delivered to its operational site. This includes all activities related to:

- Error correction
- Adaptation to requirements
- Perfective improvement
- Enhancements

- Defect correction
- Making correction to overlooked requirements
- Improving the functionality of the software program
- Growth of the software program
DEFINITIONS
Software Support Organizations

- AFSPC estimates maintenance cost based on:
  - available budget
  - number of people available
  - prioritization of maintenance items

- SM-ALC estimates maintenance cost based on:
  - size of development effort
  - percentage of code to be modified
  - number of people available

DEFINITIONS
Commercial Model Developers

- REVIC: uses a percentage of development cost on an annual basis to project the magnitude of the maintenance effort. Environmental adjustment factors, specifically required reliability (RELY) and modern programming practices (MODP) are cost drivers to the software maintenance effort.

- SASET: uses a percentage of development cost on an annual basis to project maintenance effort. A unique set of complexity attributes is used to reflect the maintenance environment.

- SEER-SEM: uses size, support personnel capabilities and experience, and number of support years to factor the development effort and derive a maintenance cost.
DEFINITIONS
Commercial Model Developers (continued)

- PRICE S: uses unique sub-models to estimate maintenance, enhancement and growth/modification. These sub-models are based on size, complexity, and the productivity factor of the maintenance team.

- SLIM: uses a parameter, mean time to defect, to estimate only the maintenance portion of software support. New capability or enhancements are not estimated. Mean time to defect uses two input values: operational run time and units, to generate maintenance cost.

- SoftCost: uses a factor, annual change traffic, from the nominal development effort along with environmental adjustments to estimate the maintenance effort. The model estimates only the maintenance effort, but it could be calibrated to account for enhancement and/or growth.

- CheckPoint: uses a support scenario, the number of support years, and looks at how much code is new, changed, or deleted to estimate the effort for only software maintenance. The model can be calibrated to account for enhancement and/or growth.
COMMERCIAL MODELS COMPARISON

<table>
<thead>
<tr>
<th>SUPPORT Attributes</th>
<th>REV-</th>
<th>SASE</th>
<th>IDES</th>
<th>PROFE</th>
<th>SLM</th>
<th>SoPC</th>
<th>ChordPac</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance</td>
<td>X*</td>
<td>X*</td>
<td>X</td>
<td>X*</td>
<td>X*</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Agreements (as-is)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enhancements</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Panacea (Closed)</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* Use percentages for initial change of software. This percentage could be used in a survey for other software.

SOFTWARE LIFE CYCLE COST DRIVERS

- Development (25%)
  - Lines of code
  - Development complexity
  - Programming language
  - New design
  - Reuse

- Maintenance (75%)
  - Same as for development, plus:
  - Number of operating locations
  - Service life
  - Quality
  - Expected growth and enhancements

5
### SOFTWARE SUPPORT METRICS

- **Number of Years in Maintenance**
- **Size**
  - SLOC Added
  - SLOC Modified
  - SLOC Deleted
- **COTS**
  - Number of COTS Packages Added
  - Size of COTS Packages
- **Schedule (MM/YY)**
  - Start Date
  - Stop Date
- **Number of Operational Sites**
- **Documentation**
  - Number of Revisions
  - Date of Last Revision
- **Quality**
  - Number of Software Problem Reports
  - Number of Defects
  - Average time to fix a single defect
- **% of Engineering Change Orders per Year**
- **Effort**
  - Average Annual Software Maintenance
  - Average Number of Hours per Person Month

### AFCAA STUDY

- **Post Deployment Software Support Estimating Methods**
  - Estimating methods used to determine resources for newly deployed software systems
  - Estimating methods used to determine resources for software systems that have already been deployed
  - Process for allocating resources among multiple software systems
  - Process for determining types and levels of maintenance
  - Description of activities performed in software support organizations
AFCAA STUDY (continued)

- Software Growth Data Collection
  - Estimated and actual software maintenance staff hours
  - Descriptive information
  - Data will be used for developing growth factors and CERs in the future

CONCLUSIONS

- We need to improve our estimation of software maintenance
- There is a disconnect between commercial estimating models and real world software support
- Many commercial models use a factor of development cost
- More effort needs to be expended in:
  - developing factors
  - developing CERs
  - distinguishing between statistical maintenance cost and level of effort