Empirical Research Paradigms
Software Cost Estimation
17-939A

Mary Shaw
October MMI

http://spoke.compose.cs.cmu.edu/ser01/

Overview

Effective software project management requires good predictions of effort; this requires estimates of code size

- Requirements
- Desired cost, schedule
- State of technology
- Software attributes
- Personnel attributes
- Project attributes
- Software size
- Computing resources
- Manpower (effort)
- Costs
- Development time
- Elapsed time to milestone
- Software size
- Computing resources


- Proposition: It is often worth paying for information because it helps us make better decisions.
- This paper: Survey of software cost estimation models and techniques.
  - Algorithmic models
  - Analog models
  - Price to win
  - Bottom-up
- Examples:
  - TRW Wolverton model: cost curves
  - Pamam SLM model: KDSI = C (MM/12)1/3 (dev time)4/3
  - Doty model: cost curves: MM = 2.060 KDSI 1.047 (prod of 14 effort mult)
  - RCA PRICS S model: cost, schedule curves; life cycle estimation
- COCOMO: MM = C1 (effort mult) KDSI 1.2, TDEV = C3 MM 1.3
- and summaries of four (then) newer models

COCOMO

- COCOMO-I
  - 3 levels of sophistication, trading precision (and presumably accuracy) for complexity and cost of getting the estimate
  - Basic model:
    - Organic mode (stable, forgiving) MM=2.4 KDSI 1.05
    - Semidetached mode (loosely coupled) MM=3.0 KDSI 1.12
    - Embedded mode (tightly constrained) MM=3.6 KDSI 1.20
  - and similarly estimate schedule from MM
  - Intermediate model: Look at 10-15 cost drivers, derive multiplier
  - Advanced: even more
- COCOMO-II
  - Separate different kinds of development (objects, generators, reuse)
  - Apply adjustments to exponent as well as base
Composable Software Research at Carnegie Mellon

COCOMO II Model Definition (1997)

EM: Enhanced model
Existing software cost estimation models are deficient in dealing with
marketplace sectors,
new development techniques,
the level of information available, and
a project's specific success strategy
of empirical time and effort estimation. An enhanced software cost
estimation is described, capable of providing more accurate analyses /
predictions of that deal with these properties in empirical time and effort
estimation designs. The model has been tested by comparing analyses /
predictions calibrating model parameters with empirically measured
values of scale factors in 83 projects.

COCOMO II

• Successful 2nd generation empirical model
• Balancing accuracy and effort
  > COCOMO I provided trade of accuracy for estimation effort;
  intermediate model proved to be the durable one
  > COCOMO II foregoes that choice, but provides models of
different complexity for different types of projects
• Simple MM=K*size^{1.3x} model elaborated with "cost
drivers"
  > drivers convert subjective through quantitative information to
  weights
  > weights apply both to K and exponent
• additional corrections for reuse, generated code, rework

Projected Software Practitioner Profile

End-user programming
(55,000,000 people)

Projected Software Practitioner Profile

First, Choose Model to Match Sector

Don’t need COCOMO
Use Object Points
Tailor a mix of Application Composition,
Early Design, and Post-Architecture models

First, Choose Model to Match Sector

End-user programming
(55,000,000 people)

End-user programming
(55,000,000 people)
Post-Architecture Model

- Applies after design has progressed enough to get good data
  > use source instructions and/or function points for sizing
  » effort nomimal = A * size^B
  > uses modifiers for reuse, code lost to reqts change, re-engineering
  > has 17 cost drivers that affect the constant A
  » effort adjusted = effort nominal * Prod_i=1..17 (effort-multiplier_i)
  > has 5 factors that affect the exponent B
  » B = 1.01 + .001 * sum_i=1..5 (weight_i)
  > cost drivers and factors come from tables that allow subjective (ordinal-scale) judgments -- than Appendix D converts those to ratio-scale values!

Fenton/Pfleeger: Software Metrics (1997)

<table>
<thead>
<tr>
<th>Scale</th>
<th>Intuition</th>
<th>Preserves</th>
<th>Example</th>
<th>Legitimate transformations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>Simple classification, no order</td>
<td>Differences</td>
<td>Horse, dog, cat</td>
<td>Any one-to-one remapping</td>
</tr>
<tr>
<td>Ordinal</td>
<td>Ranking according to criterion</td>
<td>Order</td>
<td>Tiny, small, medium, big, huge</td>
<td>Any monotonic increasing remapping</td>
</tr>
<tr>
<td>Interval</td>
<td>Differences are meaningful</td>
<td>Size of difference</td>
<td>Temperature in Celsius or Fahrenheit</td>
<td>Linear remappings with offset (ax+b)</td>
</tr>
<tr>
<td>Ratio</td>
<td>Has a zero point</td>
<td>Ratios of values are meaningful</td>
<td>Absolute temperature (Kelvin), values in currency units</td>
<td>Linear remappings without offset (ax)</td>
</tr>
<tr>
<td>Absolute</td>
<td>Exact</td>
<td>All relations</td>
<td>Number of instances</td>
<td>None</td>
</tr>
</tbody>
</table>

COCOMO II

- Good example of empirical model, with extensive fitting, calibration, and guidance about re-fitting for new environments
- Model combines information on ordinal, interval, ratio scales
- Refinements move model from empirical toward structural

Plan for the next few weeks

- Mon 1 Oct
  > Research paradigms and pro forma abstracts. Re-read Newman 94 and Shaw01, look at paper-writing advice on resources page
- Wed 3 Oct, Mon 8 Oct, Wed 10 Oct
  > Four research papers per class, 15-minute presentations
  > Choose a significant paper in your chosen topic area that illustrates both good results and good research strategy
  > Classic papers are prime candidates
- Mon 15 Oct
  > Plan projects, revisit framework for analysis. Re-read Redwine/Riddle
- Wed 17 Oct for 11 classes
  > 1 hour presentation/discussion of topic area
  > 1 hour other topic: classic paper, unify pro forms abstracts
Current interests

<table>
<thead>
<tr>
<th>Topic</th>
<th>Responsible Party</th>
<th>Sounding Board</th>
</tr>
</thead>
<tbody>
<tr>
<td>SW analysis</td>
<td>Vahe Poladian</td>
<td>George Fairbanks, Dean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sutherland, Tim</td>
</tr>
<tr>
<td>Maintenance</td>
<td>Dean Sutherland</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(analytic for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>maintenance)</td>
</tr>
<tr>
<td>Reliability</td>
<td>Beth Latronico</td>
<td></td>
</tr>
<tr>
<td>Reverse engineering</td>
<td>Elissa Newman</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bridgette Spitznagel</td>
</tr>
<tr>
<td>SE for security</td>
<td>Owen Cheng</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vahie Poladian, Tim</td>
</tr>
<tr>
<td>Tools &amp; environments</td>
<td>George Fairbanks</td>
<td></td>
</tr>
<tr>
<td>Middleware</td>
<td>Bridgette Spitznagel</td>
<td>George Fairbanks,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tim Halloran</td>
</tr>
<tr>
<td>Testing</td>
<td>Paul Li, Annie Lu</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(for reliability)</td>
</tr>
<tr>
<td>SW architecture</td>
<td>Charles Shelton</td>
<td></td>
</tr>
<tr>
<td>At large</td>
<td></td>
<td>Beth Latronico</td>
</tr>
</tbody>
</table>

Presentation Slots 10/3 to 10/10

- Wed 3 Oct
  > Shaw: Software Cost Estimation [Boehm97, supplemental Boehm84]
  > Tim Halloran: Configuration management [Rockland97, Tichy85]
  > Dean Sutherland: Slicing in Maintenance [Gallagher&Lyle91]
  > Annie Luo: Testing Tools [Hamlet&Voss <need citation>,
    supplemental Gelperin/Hetzel <need citation>]

- Mon 8 Oct
  > Elissa Newman: Program Slicing [Weiser81]
  > Owen Cheng:
  > George Fairbanks: Statemate [Harel87]
  > Bridgette Spitznagel:

- Wed 10 Oct
  > Vahe Poladian:
  > Beth Latronico:
  > Charles Shelton: Programming in the Large [DeRemer/Koen76]
  > Paul Li:

Newman's Pro Forma Templates for Engineering

EM: Enhanced model
Existing model-type models are deficient in dealing with properties of solution strategy. An enhanced model-type is described, capable of providing more accurate analyses/predictions of properties in solution strategy designs. The model has been tested by comparing analyses/predictions with empirically measured values of properties.

ES: Enhanced solution
Studies of existing artifact-type have shown deficiencies on property. An enhanced design for an artifact-type is described, based on solution strategy. In comparison with existing solutions, it offers enhanced levels of property, according to analyses based on model-type. These improvements have been confirmed/demonstrated in tests of a working artifact-type based on the design.

ET: Enhanced tool
The effectiveness of model-type/solution strategy in supporting the design of artifact-type has been demonstrated. An enhanced tool/method is described for the design of artifact-type based on model-type/solution strategy. Examples are provided confirming the effectiveness of its support for model-type/solution strategy.

Additional Pro Forma Templates for HCI

RS: Radical solution
A radical solution to the problem of problem definition is described, based on solution strategy. In comparison with existing normal solutions it offers advantages, which have been demonstrated in preliminary tests, but it leaves a number of side effects to be addressed including list of side effects. Strategies are suggested for addressing these side effects.

XH: Experience and/or Heuristic
Studies reported here of application supported by supporting technology generate a number of findings concerning issues, including list-of-findings. They indicate that requirement is/is not met by design-heuristic.

ESNA: Existing Solution Applied to New Area [Latronico]
The technique existing technique has been successfully applied in area-x with benefits for area-x. It is hoped that technique will provide similar benefits in area-y. An experimental evaluation of technique for area-y has been carried out using system from array-y. Benefits for area-y have been shown to hold (optional: if technique is altered by alteration list). Limitations remaining to the application of technique to area-y include limitation list.