Software Estimation Experiences at Xerox

Dr. Peter Hantos
Office Systems Group, Xerox
Theme

Is making bad estimates a crime?
No, but it is certainly not victimless …
Objectives

- Focus on recent piloting efforts
  - On corporate/business group level
  - On project level

- Address issues from a wide variety of angles:
  - Theoretical
  - Practical
  - Cultural
  - Management
  - Other, “soft” issues
Agenda

- Objectives
- A Corporate-Level project
- Business Group and Project-Level pilot experiences
- Cultural and Management barriers
- Sizing issues
- Conclusions
- Acknowledgement
- References
Supporting the Schedule Slip-Rate Metrics

- Slip-Rate Definitions:

  - Cycle Time = Actual End Date – Actual Start Date + 1

  - Planned Duration = Planned End Date – Actual Start Date + 1

  - Schedule Slip Rate = \( \frac{Actual \ End \ Date - Planned \ End \ Date + 1}{Planned \ Duration} \)
Slip-Rate Issues

✔ Software Slip-Rate Issues:
  - We have to support the overall product benchmarking process, but also have to provide specific guidelines to software project planners
    • Used COCOMO-II as a research tool
    • Promoted COCOMO-II as primary estimation tool

✔ Product* Slip-Rate Issues:
  - It is difficult to make the connection between product slip, hardware slip and software slip
    • Used Anchor Points to provide solution

* Dominant Xerox products are viewed as Software-Intensive Systems
Hardware-Software processes

Business Process

1. Define Market Attack Plan & Technology
2. Define Product & Deliver Technology
3. Design Product
4. Demonstrate Product

Subsystem Development (Model Shop)
Subsystem Development (Pilot Plant)
Assembly & system Integration
Integration with target hardware and testing
Mfg. Startup

Hardware Process

Concept and Technology
Solid Model Design
AP
AP
AP

Software Process

INCEPTION ELABORATION CONSTRUCTION TRANSITION D & M
Data Elements for S/W Slip-rate

- **Cycle Time**
- **Effort to Find and Fix Software Defects**
- **Customer-Reported Software Defects**

**Stage Effort**

- **Start Date (LCA):** Actual; Committed Features
- **End Date (SDR):** Planned and Actual; Delivered and Renegotiated Features

**Total Effort**
Benchmark Schedules Directive

✓ The Corporation requires the use of benchmark schedules to determine planned program duration:

– Product benchmark schedules are driven by a composite newness/complexity index*, comprehending technology, hardware and software. Reference Database is confidential.

*Index and Database developed by the PRTM management consulting group
Specific S/W issues

✓ “Newness” emphasizes “precedentedness”
✓ Higher level of reuse-intent is implicitly equated with precededentedness
✓ SLOC definition deals only with new code, actual reuse is not considered
✓ Internal software complexity (as it is defined in COCOMO) is not considered
COCOMO-based Recommendations

✔️ Instead of simply considering new code, use the COCOMO “Equivalent KLOC”*

✔️ Put emphasis on complexity instead of precedentedness:
  – See productivity ranges in COCOMO-II**:
    – PREC 1.4 (deviation = 0.046)
    – CPLX 2.4 (deviation = 0.014)

* Page 22, REF [2]
** Page 171, REF [2]
COCOMO for Benchmarking

✓ Initial stage: “Average” benchmark
   - All drivers are set to nominal, except for
     • \textit{TIME} = H \quad \text{Due to real-time requirements}
     • \textit{STOR} = VH \quad \text{Low UMC is always a major constraint}
     • \textit{PLEX} = L \quad \text{It is always hard to keep up with the pressure from h/w and OS vendors both on servers and clients}

✓ For “Best In Class” benchmark
   - Increase/change the following drivers:
     • \textit{RUSE, ACAP, PCAP, PCON, APEX, PLEX, LTEX, TOOL, SCED}
   - Increase \textit{PMAT}
     • (Note that CMM Level-2 translates into “nominal” driver rating, and Xerox development teams are expected to operate on at least CMM Level-2)
Still needs more research …

✓ …on estimating Elaboration effort/schedule
  • Early Design Model is not really applicable for the difficult and prevalent parts of the systems

✓ … on estimating Transition effort/schedule
  • COQUALMO might be used as a starting point…
  • …but the h/w – s/w interdependencies are not captured, consequently a more complex defect introduction and removal model would have to be developed
Business Group and Project Level

✓ Cultural/Management Barriers
  – Inability to manage expectations
  – Improper positioning of pilot project(s)
  – Unresolved size metrics concerns
Misinterpretation of the CMM

- Core issue: The CMM Goals can be satisfied without assuring performance quality targets for process execution
- It is enough to define, document and follow processes, however weak they are
- Shooting for the Grade – “Why do more if less is enough to be assessed on Level-x”
Sizing Issues

✓ Conceptual
  - Almost impossible to get consensus on what size metrics to use
  - Backfiring FP into SLOC is highly inaccurate
  - High programmer sensitivity due to potential misuse of SLOC
  - *Collectors* of data and *users* of data are different
  - Definition of DM in COCOMO is vague and inconsistent
  - Definition of CM in COCOMO is inconsistent
  - Deleted LOC (and design?) are intentionally not accounted for in COCOMO

✓ Practical
  - Counting LOC, particularly measuring Modified Code Size (CM) for COCOMO is not as simple as we were led to believe by metrics gurus
Measuring Modified Code Size

Factors to consider

1. Directories of old and new code have to be accessed simultaneously
2. Tagging mechanism is needed to identify Adapted Code Modules
3. A “DIF”-type tool is needed (Caveat: OS dependent) to compare files
4. Default tool capability: showing modifications visually, on screen, and on the physical lines only
5. Results have to be segmented to “untouched”, and “modified+new”
6. A CodeCount – type analyzer has to applied to both segments to determine the equivalent Logical Lines of Code
7. DIF and CodeCount have to be integrated with the SCM system

Issues

1. Having CodeCount is a very small fraction of the needed solution
2. Integration of all of these tools with the SCM tools is a big undertaking
3. Due to tools and vendor diversity all possible solutions have to be highly customized, and a general solution is not really feasible.
Conclusions

- Metrics problems in general are more political than technical in nature.
- There are substantial cultural and management barriers to the use of sophisticated Cost Estimation Methods, or even to start a small-scale pilot.
- Success of a Cost Estimation pilot is highly dependent on having an accepted metric and stable method for sizing.
Acknowledgement

We would like to acknowledge Mr. Don Reifer’s dedicated support on behalf of CSE/USC to the piloting teams and also thank him for the workshops he led in Rochester.
References