SOFTWARE PROJECT MANAGEMENT AND COST ESTIMATION

Object points (function + code)

- Count number of
  - screens
  - reports
  - 3GL components (Java, C++ classes)
- For each use following weighting based on complexity
- Object type simple Media Difficult
- Screen 1 2 3
- Report 2 5 8
- 3GL components 10

Function points verses Object points

- Function points
  - Established standard
  - Much legacy estimation data available
  - Supported by many tools
  - Can be calculated very early on, requirements stage
- Object points
  - Newer
  - Easier to calculate
  - Provides allowance for re-use

Cost Estimation

- From size estimation (FP, OP or KLOC)
- Map to cost using cost estimation model
- Two error factors
  - Original estimation error
  - Cost derivation error
- Or
- Use direct estimation
  - E.g. poker planning

Factors affecting productivity

- Application domain experience
- Process quality
- Project size
- Technology support
- Working environment

Estimation techniques

- Algorithmic cost modelling e.g. Constructive Cost Modelling (CoCoMo)
- Expert judgement
- Estimation by analogy
- Parkinson's Law
- Pricing to win

Constructive Cost Modelling (CoCoMo)

COCOMO

- Barry W. Boehm 1981
- 63 projects at TRW Aerospace
- From 2,000 to 100,000 lines
- COCOMO II 2000
  - University of Southern California
  - University of California Irvine
  - COCOMO™ II Affiliates' Program

The COCOMO method

- Input
  - Conduct of the project (e.g. design model)
  - Staff available
  - Hardware and CASE tools involved
  - Nature of the product
- Output estimates
  - Size of the system (LOC and function points)
  - Project schedules and team factors
  - Cost and staffing profiles.

(1) Application Composition

- Application composition
  - For: prototype system using scripting, SQL etc.
  - Uses: object points
- PM = (NOP x (1 - % reused/100)) / PROD
- PM : Person months
- NOP: Total number object points
- PROD : Object point productivity (4 low 50 high)

(2) Early Design

- Early design
  - For: initial effort estimation
  - Uses: function points
  - Effort = A x $Size^B$ x M
  - A : constant, found to be about 2.94
  - Size : thousands of lines of code (derived from SLOC)
  - B : measure of product novelty
  - M: product of 7 values each between 1 to 6

Re-use model (auto gen code)

- Reuse - variant (A)
- For: integration projects using reusable or automated code generation
- Uses: lines of code reused, or generated
- For automatically generated code:
- PM = (ASLOC x AT/100) / ATPROD

Re-use model (new code)

- Reuse - variant (B)
- Where new written code is required:
- ESLOC = ASLOC x (1 – AT/100) AAM

- Estimate of error for new code

Post-architecture

- For: overall development effort
- Uses: number of lines of source code
- $PM = A \times Size^{B} \times M$

Project duration and staffing

- Calendar time (TDEV) can be calculated:
  $TDEV = 3 \times (PM)^{(0.33+0.2*(B-1.01))}$

  Where PM is the effort computation and B the exponent as computed earlier.
- Staffing (PM) estimates are affected by the communication problem, and as noted before, is not linear.

Summary

- Cost Estimation Modelling
  - Relies on expert judgement
  - Requires much effort
  - Produces subject results
- Alternatives
  - Group estimation
  - Analogy estimation
  - Pricing to win