Lecture 0
Software Project Estimation

Purpose: Cover basic concepts and practices for software project estimation

What is Estimation?  Spec → Effort + Schedule + Cost

Poor Estimation is Common
- Bias -- overly optimistic about self, others
- Ill-defined estimation process
- Lack of experience
- Pressure to please (say what is expected by management)
- Lack of historical data on which to base estimate
- Misuse of estimation models

Estimation Principles
- Employ multiple estimators
- Use multiple techniques
- Require justification (basis of estimate) for each estimation
- Employ mathematical or facilitated process to combine multiple estimates
- Associate probabilities with estimates

Software Productivity
- Five Major Factors [Basili ’78]
  - People -- size and expertise of individuals, teams
  - Problem -- specific project requirements, application domain
  - Process -- software management and development methods
  - Product -- complexity and required qualities of the product (reliability)
  - Resources -- facilities and environment used by developers

- The BIG 29 [Walston & Felix, ’77]

- Empirical Models reflect specific projects, experiences, applications
  Beware of empirical models -- human/organizational behavior is highly variable

Estimation Life Cycle

1. Problem/Need Statement
   - Fuzzy requirements
Limited opportunity to clarify requirements
- Origin of uncertainty

2. **Scope problem**
- Imperative to state assumptions
- Sketch solution proposal
- Assume operating and user environment
- Disclaimers about boundaries

3. **Estimate size**
- Effort is proportional to size
- Cost models are exponential: \( \text{Effort} = A \times \text{Size}^B \), for constants \( A, B \).
- **Size Measures**
  - Source Lines of Code (SLOC) -- universal measure
  - Function Points (measure of inherent capability delivered by software)
  - Uncertainty about SLOC initially high, decreases over life cycle
  - FP known earlier in life cycle
  - FP \( \rightarrow \) SLOC conversions known based on languages

- **Estimation Techniques**
  - Experience is paramount
  - Project history database essential
    - component and whole-project development data
    - technical software metrics
    - project metrics

- **Decomposition Techniques**
  - Software decomposition tree
  - Granularity depends on available time, experience, historical data on hand
  - Assign size (or effort) estimate to component or decompose further
  - Assemble estimate from leaves of decomposition tree
  - Adjust estimate to account for integration of components higher in tree

- **Analogy Techniques**
  - Locate historical data on similar components
  - Adjust estimate for differences to historical component

- **Consensus Building**
  - Delphi method -- anonymous estimation with facilitated variance reduction
  - Weighted averaging: expected = \( [ \text{LO} + 4 \times \text{AVG} + \text{HI}] / 6 \)
4. Estimate effort and optimal schedule/duration
   - Process
     - Select cost model based on
       - size metric
       - project type
     - Apply cost model
     - Refine model parameters
     - Repeat estimate
   - Walston & Felix Cost Model
     - empirical model from 60 projects (29 project characteristics)
     - Effort: \( E = 5.2 \times L^{0.91} \)
     - Duration: \( D = 4.1 \times L^{0.36} = 2.47 \times E^{0.35} \)
     - Staff Size: \( S = 0.54 \times E^{0.06} \)
     - Documentation Pages: \( P = 49 \times L^{1.01} \)
   - COCOMO (Constructive Cost Model, Boehm)
     - 3 types of projects
       - Organic -- simple, small teams, experienced, familiar domain
       - Semi-detached -- moderate, mixture of skill/experience/familiarity
       - Embedded -- software embedded in hardware, little user interface. Difficult.
     - 3 levels of model
       - **Basic** model -- 2 coefficients per project type
       - **Intermediate** model -- effort estimation incorporates *effort multipliers* to account for *cost drivers* that make projects easier or harder. Different effort coefficients.
       - **Detailed** model -- not covered.
   - Basic COCOMO
     - Effort (labor months): \( E = a \times L^b \)
     - Duration (calendar months): \( D = c \times E^d \)
     - A different \((a,b,c,d)\) tuple for each project type.
       - Organic: \((2.4, 1.05, 2.5, 0.38)\)
       - Semidetached: \((3.0, 1.12, 2.5, 0.35)\)
       - Embedded: \((3.6, 1.20, 2.5, 0.32)\)
   - Intermediate COCOMO
     - 15 cost drivers (project characteristics) that may adjust effort
       - Each cost driver become multiplier in range 0.9-1.4
       - Product of all cost drivers provides *effort adjustment factor (EAF)*
     - Effort: \( E = a \times L^b \times EAF \)
Putnam Cost Model (large projects)
- \( L = C \times E^{1/3} \times D^{4/3} \),
  - \( C \): state of technology, development environment (2000-11000),
  - \( D \): duration (calendar years),
  - \( E \): effort (labor years),
- \( E = L^3 / (C^3 \times D^4) \)

**Warning:** Do not over compress the schedule: minimum duration.
- Small extension to schedule can yield large savings in effort
- Conversely, small compressions can drastically increase effort.
- Boehm: *There exists a limit beyond which a project can not reduce schedule by buying more personnel and equipment. This limit occurs at ~ 75% of nominal schedule.*

Use Putnam model to check for schedule sensitivity.
- Compression from \( D \) to \( 2D \) => \( 1/16 \)th the effort.

SLIM -- automated estimation, planning tool.
- Based on Putnam model, Norden curve
- Outputs
  - Ranges for estimates based on uncertainty factors
  - Month by month distribution of effort
- Modeler/planner capabilities
  - model calibration
  - characterization of software and project properties
  - software sizing
- Internals
  - Putnam cost model
  - PERT scheduling
  - Linear programming to satisfy/solve project constraints

ESTIMACS
- Inputs: function points
- Outputs: effort, staffing, risk, hardware capacity

5. Develop schedule
- Goal: tasks + start-date + end-date
- Approach: exploit parallelism in project activities
- Use models to give gross duration, staffing
- Work Break-Down method
  - Process breakdown vs Product breakdown
  - Separate accounting per node in the breakdown tree
- Parallelism not shown clearly
- Development Task Driven
  - Task list
  - Task precedence
  - Activity network (PERT chart)
  - Critical path
- Refine of schedule
  - revise tasks
  - revise staffing
- Present schedule as Gantt Chart

6. Establish duration
   - Goal: determine the calendar time from project start to project finish
   - Constraint: often this comes first, as a given

7. Determine cost
   - Goal: estimate the cost of following the schedule
   - Factors: Cost of money – inflation, interest on $$ borrowed to pay employees
   - Realities: Cost of MISSED SCHEDULE -- may be loss of on-time bonuses or late penalties.