Project Estimation

Estimation approaches:
- "similar" projects
- decomposition techniques
- empirical models

Estimating from similar projects:
- must account for project similarities, and differences, (technical, functional, size, staff, etc.)
- accuracy of estimates depends on large pool of accurate data from previous projects
Decomposition Techniques

1. Break the project down by some quantifiable attribute (may require some architecture/design work)
2. Determine the resources required for each attribute element (may base on past project experience)
3. Sum the resource requirements
4. Most commonly used: LOCs and Function Points

We need considerable data on LOC/person-month and FP/person-month to estimate accurately
Challenges with LOC Estimates

- How accurate are our estimates of LOC?
- What constitutes a line of code?
- What do we do about deleted lines of code (as may occur during maintenance)?
- What behavior do we instill in by using LOC?
- What are the effects of language on LOC?
- If coding represents 10-20% of overall project development time, is LOC an appropriate granule from which to estimate (degree of uncertainty)?
- We may have to perform considerable design work to estimate LOC accurately
Function Points

Function Points are computed from a sum-of-products of system characteristics.

May be done early in the project with minimal system design and decomposition.

Function point calculation is language independent.

Does not penalize creative implementations that may require fewer-than-expected LOC.
## Function Point Computation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Count</th>
<th>Weighting Factor</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td># user inputs</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td># user outputs</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td># user inquiries</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td># files</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td># ext. ifcs</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

### Function Point Total:
Three Point Estimated Value Estimation

Technique used to weight the expected value of an estimation variable (e.g., LOC)

Used a weighted average of sums from the optimistic, most likely, and pessimistic estimates

\[ EV = \left( S_{\text{opt}} + 4S_m + S_{\text{pess}} \right) / 6 \]
For each FP complexity estimator, we may use the three point estimated value to arrive at a more accurate estimate.

Function points may be extended to include:
- algorithmic complexity (Feature Points)
- function and control capabilities (3D Function Points)
Process-Based Estimates

Most common technique used to estimate project resource needs (especially people & time needs)

Overall Approach:
- decompose each software engineering stage into set of tasks and activities for which the effort may be estimated
- sum overall time and resources required

Tools such as Microsoft Project and Timeline are helpful
Empirical Models

Based on curve fitting (regression analysis) from large pool of collected data

Overall format: \( E = A + B \times (ev)^c \)

Where:
- \( E \) is the effort in person-months
- \( A, B, \) and \( C \) are empirically derived constants
- \( ev \) is the estimation variable (LOC or FP)

Empirical models must be calibrated for the local environment and types of projects
COCOMO Model
(COst CONstructive MOdel)

Developed by Barry Boehm in 1981

Three forms:
* **Organic**—small, simple projects with rigid requirements implemented by small teams with experience
* **Semi-detached**—intermediate size & complexity, some fluid requirements, team has mixed experience
* **Embedded**—system is constrained by hardware, software, & operational constraints
COCOMO (continued)

Basic formula:
\[ E = a_b KLOC^{b(b)} \]

where:
- \( E \) is the effort in person-months
- \( a_b \) and \( b(b) \) are constants depending on the model

\[ D = c_b E^{d(b)} \]

where:
- \( D \) is the development time in months
- \( c_b \) and \( d(b) \) are constants, depending on the model
Regardless of the favored estimation method, it is recommended that more than one approach be used.

Estimates should be developed independently and compared once they are completed.

Significant differences in estimates should signal a need to re-evaluate the project, assumptions, and models used.

Estimates are always more accurate when historical data is used to calibrate the models.