1.264 Lecture 3

Time and resource estimation

Next class: Read chapters 7, 8. Hand in exercise solution after class.
Form groups for homework.
Hand in today’s exercises on paper.
Choose system implementation goal

• Product, time, cost: you can lock 2 of the 3
• What do you need? Examples (not exhaustive):
  – High certainty in meeting a schedule constraint
    • If schedule is fixed, either product or resources are fixed but not both
  – Runaway prevention (with bad past history)
    • Low level of process maturity, recovery mode
  – Predictability (tied to budget, other programs)
    • Staff and schedule are fixed. Product must flex.
  – Lowest cost
    • Requires good people, good process
  – Desire for free overtime (startups, cheap companies)
• These are not the same problem
  – Choose a method appropriate to the problem
    • Tables next time: fastest possible, most efficient, nominal
Exercise

• What do you need?
  – Change from uniform to regional assortments in retail supply chain in a large retailer
  – Sourcing key inputs from new overseas suppliers for holiday season
  – Implementing a new bus location system for a public transit agency that displays on smart phones and PCs
    • There have been many failures in the past.
  – Implementing electronic health records at a large hospital

• Take 10 minutes:
  – Recommend a goal
  – List top 3 factors or key unknowns to be researched early in the decision
Solution (one of many)

• **What do you need?**
  – Change from uniform to regional assortments in retail supply chain in a large retailer
    • Lowest cost
  – Sourcing key inputs from new overseas suppliers
    • High certainty in meeting schedule
  – Implementing a new bus location system for a public transit agency that displays on smart phones and PCs
    • Runaway prevention
  – Implementing electronic health records at a large hospital
    • Runaway prevention. Difficult to tell doctors what to do.
Estimation

Image by MIT OpenCourseWare.
Estimation steps: software example

- **Estimate size of project:** (product)
  - Methods/behaviors (formerly function points) to be configured, modified, written and/or implemented
  - Lines of code: depends on language, tools (technology)

- **Estimate effort** (person-months) (process/people)

- **Estimate schedule** (calendar-months)

- **Estimate team size as** (person-months / calendar-months)
  - This is just algebra. Team size is not an independent variable.

- Provide estimates in ranges and refine for increasing precision as project progresses
  - Never give a point estimate
  - Use the convergence curve

- This process is also used for non-software projects
  - Non-software metrics harder to find. Track your own.
Project size estimation (product)

• Count:
  – Inputs: Web pages, forms, messages from other systems, XML documents
  – Outputs: Web pages, reports, graphs, messages, XML documents
  – Inquiries: simple Web inputs, generally producing a single output
  – Logical internal files: tables or files in own database
  – External interface files: tables or files used from other systems or databases
# Method (function point) multipliers

## Function-Point Multipliers

<table>
<thead>
<tr>
<th>Program Characteristic</th>
<th>Low Complexity</th>
<th>Medium Complexity</th>
<th>High Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of inputs</td>
<td>X 3</td>
<td>X 4</td>
<td>X 6</td>
</tr>
<tr>
<td>Number of outputs</td>
<td>X 4</td>
<td>X 5</td>
<td>X 7</td>
</tr>
<tr>
<td>Inquiries</td>
<td>X 3</td>
<td>X 4</td>
<td>X 6</td>
</tr>
<tr>
<td>Logical internal files</td>
<td>X 7</td>
<td>X 10</td>
<td>X 15</td>
</tr>
<tr>
<td>External interface files</td>
<td>X 5</td>
<td>X 7</td>
<td>X 10</td>
</tr>
</tbody>
</table>

Image by MIT OpenCourseWare.
### Example of Computing the Number of Function Points

<table>
<thead>
<tr>
<th>Program Characteristic</th>
<th>Function Points</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Low Complexity</td>
</tr>
<tr>
<td>Number of inputs</td>
<td>6 \times 3 = 18</td>
</tr>
<tr>
<td>Number of outputs</td>
<td>7 \times 4 = 28</td>
</tr>
<tr>
<td>Inquiries</td>
<td>0 \times 3 = 0</td>
</tr>
<tr>
<td>Logical internal files</td>
<td>5 \times 7 = 35</td>
</tr>
<tr>
<td>External interface files</td>
<td>9 \times 5 = 45</td>
</tr>
<tr>
<td>Unadjusted function-point total</td>
<td></td>
</tr>
<tr>
<td>Influence multiplier</td>
<td></td>
</tr>
<tr>
<td>Adjusted function-point total</td>
<td></td>
</tr>
</tbody>
</table>

Image by MIT OpenCourseWare.
Influence multipliers

- Data communications
- Distributed processing
- Heavy use
- Performance
- Transaction rate
- Online data entry
- End user efficiency

- Online update
- Complex processing
- Reusability
- Installation ease
- Operational ease
- Multiple sites
- Facilitate change

Rate each element from 0-5
Influence multiplier is $0.65 + 0.01$ (sum of elements), varies between 0.65 and 1.35
## Effort estimation: lines of code (technology)

### Approximate Language Levels

<table>
<thead>
<tr>
<th>Language</th>
<th>Level</th>
<th>Statement per Function Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembler</td>
<td>1.0</td>
<td>320</td>
</tr>
<tr>
<td>C</td>
<td>2.5</td>
<td>125</td>
</tr>
<tr>
<td>C++</td>
<td>6.5</td>
<td>50</td>
</tr>
<tr>
<td>AWK</td>
<td>15.0</td>
<td>25</td>
</tr>
<tr>
<td>Perl</td>
<td>15.0</td>
<td>25</td>
</tr>
<tr>
<td>Lisp</td>
<td>5.0</td>
<td>65</td>
</tr>
<tr>
<td>Ada 83</td>
<td>4.5</td>
<td>70</td>
</tr>
<tr>
<td>Pascal</td>
<td>3.5</td>
<td>90</td>
</tr>
<tr>
<td>Focus</td>
<td>8.0</td>
<td>40</td>
</tr>
<tr>
<td>Oracle</td>
<td>8.0</td>
<td>40</td>
</tr>
<tr>
<td>Sybase</td>
<td>8.0</td>
<td>40</td>
</tr>
<tr>
<td>Paradox</td>
<td>9.0</td>
<td>35</td>
</tr>
<tr>
<td>Modula 2</td>
<td>4.0</td>
<td>80</td>
</tr>
<tr>
<td>dBase IV</td>
<td>9.0</td>
<td>35</td>
</tr>
<tr>
<td>Fortran 77</td>
<td>3.0</td>
<td>110</td>
</tr>
<tr>
<td>GW Basic</td>
<td>3.25</td>
<td>100</td>
</tr>
</tbody>
</table>

Continued...

<table>
<thead>
<tr>
<th>Language</th>
<th>Level</th>
<th>Statement per Function Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quick Basic 3</td>
<td>5.5</td>
<td>60</td>
</tr>
<tr>
<td>Visual Basic 3</td>
<td>10.0</td>
<td>30</td>
</tr>
<tr>
<td>Cobol (ANSI 85)</td>
<td>3.5</td>
<td>90</td>
</tr>
<tr>
<td>Macro assembler</td>
<td>1.5</td>
<td>215</td>
</tr>
<tr>
<td>SAS, SPSS, other statistics packages</td>
<td>10.0</td>
<td>30</td>
</tr>
<tr>
<td>Smalltalk 80; Smalltalk/V</td>
<td>15.0</td>
<td>20</td>
</tr>
<tr>
<td>Excel, Lotus 123, Quattro Pro, other spreadsheets</td>
<td>≈ 50</td>
<td>6</td>
</tr>
</tbody>
</table>

Approximate Language Levels

Image by MIT OpenCourseWare.
Exercise

• If you had a requirements document with:
  – 10 Web input pages,
  – 15 output pages (reports),
  – 20 (internal) database tables, and
  – no inquiries or external files,
• How many function points would it contain?
  – Assume medium complexity for each page, report, table
  – Assume influence multiplier = 1.0
• Technology:
  – If you wrote the system in C, how many lines of code would it have?
  – What if you used perl (a simpler scripting language)?
  – If you could do it in perl instead of C, would you?
• How do you decide what technology to use?
Solution

• If you had a requirements document with 10 Web pages, 15 reports, 20 database tables, and no inquiries or external files, how many function points would it contain?
  – Assume influence multiplier = 1.0
  – About 315 function points, if each item is medium complexity

• If you wrote the system in C, how many lines of code would it have?
  – About 40,000 lines of C
  – What if you used perl?
    • About 8,000 lines of perl

• If you could do it in perl instead of C, would you?
  – Yes. The system is 20% the size and will take <20% of the resources to write (scale diseconomies)

• How do you decide what technology to use?
  – The one that is capable enough to solve the problem and yields the fewest lines.
Hand in

• Upload your answers to the goals and resource estimation exercises
  – On paper. Make sure your name is on it. You may hand in one for your group. Or…
  – Upload to course Web site, in Word, Notepad, or any other common format