Topics Today

• Q&A on Assignment 3
• Product Sizing: Function Point Analysis (FPA)
• Parametric Cost Estimation: COCOMO Model
• Global Software Development
• Preparation for Quiz 4
Q&A on Assignment 3
Assignment 3

Group forming …


Use course wiki page for submission (not Moodle)
Submit one report per person (i.e., per project)
    The manager (i.e., the product owner) submits his/her report

Deadline: Tuesday (Dec 06) at 23:59
Product Sizing:

Function Point Analysis
• **Functionality** is one aspect of software size. The assumption is that a product with more functionality is larger in size and thus needs more effort to be developed.

• The first function-oriented measure was proposed by Albrecht (1979~1983).
  – He suggested a size measurement approach called the **Function Point (FP)** method.

• Function points (FPs) measure the amount of functionality in a system based upon the **system specification**.

⇒ Estimation before implementation!
Function Point Analysis (FPA)

- Models for estimating Function Points:
  - Albrecht / IFPUG Function Points
  - Mark II Function Points
  - COSMIC Function Points

Number of file types

Numbers of input and output transaction types

Model

‘System size’ (Function Points)
IFPUG Function Point Counting Elements
Albrecht/IFPUG FPs – Elements

Five function types

1. **Internal logical file** (ILF)* types – equates roughly to a data store in systems analysis terms. Created and maintained within the application to be estimated.

2. **External interface file** (EIF) types – where data is retrieved from a data store which is actually maintained by a different application.

3. **External input** (EI) types – input transactions which update internal data stores

4. **External output** (EO) types – transactions which extract and display data from internal data stores. Typically involves creating reports.

5. **External inquiry** (EQ) types – user initiated transactions which provide information but do not update data stores. Typically, the user inputs some data that guides the system to the information the user needs.

* Note: ILF is sometimes called ’Logical Internal File’ (→ LIF)
IFPUG Function Point Counting Big Picture

Step 1a

$\text{#EI} + \text{#EO} + \text{#EQ} + \text{#EIF} + \text{#ILF}$

Step 1b

$\text{Weighted EI} + \text{Weighted EO} + \text{Weighted EQ} + \text{Weighted EIF} + \text{Weighted ILF}$

$= \text{Unadjusted Function Points (UFP)}$

Step 2

$\text{Adjusted FP Count}$

$= \text{Unadjusted and Unweighted Function Count}$

Complexity Adjustment

14 Adjustment Factors

Value Adjustment Factor (VAF)

Unadjusted Function Points

Weighting of functional (technical) complexities
IFPUG Function Points (FP) – Key Elements

- External Inputs (EI)
- External Outputs (EO)
- External Interface Files (EIF)
- Internal Logical Files (ILF)
- External Inquiries (EQ)
IFPUG FP – External Inputs (EI)

**External Inputs – IFPUG Definition:**

- An external input (EI) is an elementary process that processes data or control information that comes from outside the application boundary.
- The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system.

**Examples:**
- Data entry by users
- Data or file feeds by external applications
IFPUG FP – External Outputs (EO)

• **External Outputs – IFPUG Definition:**
  
  – An external output (EO) is an elementary process that sends data or control information outside the application boundary.
  
  – The primary intent of an external output is to present information to a user through processing logic other than, or in addition to, the retrieval of data or control information.
  
  – The processing logic must contain at least one mathematical formula or calculation, create derived data, maintain one or more ILFs, or alter the behavior of the system.
  
  – **Example:**
    
    • Reports created by the application being counted, where the reports include derived information.
IFPUG FP – External Inquiries (EQ)

- **External Inquiries – IFPUG Definition:**
  - An external inquiry (EQ) is an elementary process that sends data or control information outside the application boundary.
  - The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information from an ILF or EIF.
  - The processing logic contains no mathematical formulas or calculations, and creates no derived data.
  - No ILF is maintained during the processing, nor is the behavior of the system altered.

- **Example:**
  - Reports created by the application being counted, where the report does not include any derived data.
IFPUG FP – Internal Logical Files (ILF)

- **Internal Logical Files – IFPUG Definition:**
  - An ILF is a user-identifiable group of logically related data or control information maintained within the boundary of the application.
  - The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.
  - Examples:
    - Tables in a relational database.
    - Flat files.
    - Application control information, perhaps things like user preferences that are stored by the application.
IFPUG FP – Ext. Interface Files (EIF)

• **External Interface files – IFPUG Definition:**
  - An external interface file (EIF) is a user identifiable group of logically related data or control information referenced by the application, but maintained within the boundary of another application.
  - The primary intent of an EIF is to hold data referenced through one or more elementary processes from within the boundary of the application counted.
  - This means an EIF counted for an application must be an ILF in another application.
  - Examples:
    - As for ILF, but maintained in a different system
IFPUG Function Point Counting Big Picture

Step 1a

\[
\text{Unadjusted and Unweighted Function Count} = \# \text{EI} + \# \text{EO} + \# \text{EQ} + \# \text{EIF} + \# \text{ILF}
\]

Step 1b

\[
\text{Unadjusted Function Points (UFP)} = \text{Weighted EI} + \text{Weighted EO} + \text{Weighted EQ} + \text{Weighted EIF} + \text{Weighted ILF}
\]

Step 2

\[
\text{Adjusted Function Points} = \text{Unadjusted Function Points (UFP)} + \text{Value Adjustment Factor (VAF)}
\]

14 Adjustment Factors

Complexity Adjustment
### Albrecht/IFPUG FPs – Complexity Multipliers

<table>
<thead>
<tr>
<th>Function types</th>
<th>Low complexity</th>
<th>Medium complexity</th>
<th>High complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>EI</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>EO</td>
<td>4</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>EQ</td>
<td>3</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>EIF</td>
<td>5</td>
<td>7</td>
<td>10</td>
</tr>
<tr>
<td>ILF</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>
# IFPUG FP Counting: Weighting of Technical Complexity

<table>
<thead>
<tr>
<th>Elements</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>low</td>
</tr>
<tr>
<td>External Inputs (EI)</td>
<td>__ x 3 =</td>
</tr>
<tr>
<td>External Outputs (EO)</td>
<td>__ x 4 =</td>
</tr>
<tr>
<td>External Inquiries (EQ)</td>
<td>__ x 3 =</td>
</tr>
<tr>
<td>External Interface Files (EIF)</td>
<td>__ x 5 =</td>
</tr>
<tr>
<td>Internal Logical Files (ILF)</td>
<td>__ x 7 =</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Unadjusted Function Points (UFP)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Payroll application has:

1. A transaction to add, update and delete employee details → an EI rated to be of medium complexity
2. A transaction that calculates pay details from timesheet data that is input → an EI rated to be of high complexity
3. A transaction that prints out pay-to-date details for each employee → an EO rated to be of medium complexity
4. A personnel file maintained by another system is accessed for name and address details → a simple EIF
5. A file of payroll details for each employee → an ILF rated to be of medium complexity

What would be the FP count for such a system?
## Albrecht/IFPUG FPs – Example /2

<table>
<thead>
<tr>
<th>Complexity Level</th>
<th>Number of Examples</th>
<th>FPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium complexity EI</td>
<td>1</td>
<td>4 FPs</td>
</tr>
<tr>
<td>High complexity EI</td>
<td>1</td>
<td>6 FPs</td>
</tr>
<tr>
<td>Medium complexity EO</td>
<td>1</td>
<td>5 FPs</td>
</tr>
<tr>
<td>Simple complexity EIF</td>
<td>1</td>
<td>5 FPs</td>
</tr>
<tr>
<td>Medium complexity ILF</td>
<td>1</td>
<td>10 FPs</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>30 FPs</strong></td>
</tr>
</tbody>
</table>

- If previous projects delivered 5 FPs per person-day, implementing the above should consume $30/5 = 6$ person-days
IFPUG Function Point Counting: Complexity Assessment Details

Step 2

Adjusted FP Count

Complexity Adjustment

14 Adjustment Factors

Adjusted Function Points

Value Adjustment Factor (VAF)

Step 1b

Unadjusted Function Points (UFP)

Weighting of functional (technical) complexities

Step 1a

Unadjusted and Unweighted Function Count

# EI + # EO + # EQ + # EIF + # ILF =

Weighted EI + Weighted EO + Weighted EQ + Weighted EIF + Weighted ILF =

L A H + L A H + L A H + L A H + L A H =
### IFPUG Function Types – Complexity Assessment

<table>
<thead>
<tr>
<th>Data Function Types</th>
<th>Transaction Function Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Logical Files (ILF)</td>
<td>External Interface Files (EIF)</td>
</tr>
<tr>
<td><strong>Elements evaluated for technical complexity assessment</strong></td>
<td></td>
</tr>
<tr>
<td>REcord Types (RET): User recognizable subgroups of data elements within an ILF or an EIF. It is best to look at logical groupings of data to help identify them.</td>
<td>File Type Referenced (FTR): File type referenced by a transaction. An FTR must be an Internal Logical File (ILF) or External Interface File (EIF).</td>
</tr>
<tr>
<td>Data Element Types (DET): A unique user recognizable, non-recursive (non-repetitive) field containing dynamic information. If a DET is recursive then only the first occurrence of the DET is considered not every occurrence.</td>
<td></td>
</tr>
</tbody>
</table>
IFPUG Function Point Counting Elements
IFPUG Internal Logical File – Complexity Assessment

- Identify number of Record Types (RET)
- Identify number of Data Element Type (DET)
- Determine complexity (→ used to calculate FP count)

<table>
<thead>
<tr>
<th>ILF</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-19</td>
</tr>
<tr>
<td>1</td>
<td>low (7)</td>
</tr>
<tr>
<td>2-5</td>
<td>low (7)</td>
</tr>
<tr>
<td>&gt; 5</td>
<td>average (10)</td>
</tr>
</tbody>
</table>
IFPUG Internal Logical File – Example

ILF: “Employee”

<table>
<thead>
<tr>
<th>ILF</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-19</td>
</tr>
<tr>
<td>1</td>
<td>low (7)</td>
</tr>
<tr>
<td>2-5</td>
<td>low (7)</td>
</tr>
<tr>
<td>&gt; 5</td>
<td>average (10)</td>
</tr>
</tbody>
</table>
### IFPUG External Interface File – Complexity Assessment

- Identify number of Record Types (RET)
- Identify number of Data Element Type (DET)
- Determine complexity (→ used to calculate FP count)

<table>
<thead>
<tr>
<th>EIF</th>
<th>#RET</th>
<th>#DET</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1-19</td>
<td>low (5)</td>
<td>low (5)</td>
</tr>
<tr>
<td></td>
<td>2-5</td>
<td>low (5)</td>
<td>average (7)</td>
<td>high (10)</td>
</tr>
<tr>
<td></td>
<td>&gt; 5</td>
<td>average (7)</td>
<td>high (10)</td>
<td>high (10)</td>
</tr>
</tbody>
</table>
IFPUG External Interface File – Example

Employee Administration (DB)

Payroll Software

2 RET
7 DET

IFPUG External Interface File – Example

<table>
<thead>
<tr>
<th>EIF</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-19</td>
</tr>
<tr>
<td>#RET</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>low (5)</td>
</tr>
<tr>
<td>2-5</td>
<td>low (5)</td>
</tr>
<tr>
<td>&gt; 5</td>
<td>average (7)</td>
</tr>
</tbody>
</table>
IFPUG External Input – Complexity Assessment

- Identify number of File Types Referenced (FTR)
- Identify number of Data Element Type (DET)
- Determine complexity (→ used to calculate FP count)

<table>
<thead>
<tr>
<th>EI</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-4</td>
</tr>
<tr>
<td>#FTR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>low (3)</td>
</tr>
<tr>
<td>2</td>
<td>low (3)</td>
</tr>
<tr>
<td>&gt; 2</td>
<td>average (4)</td>
</tr>
</tbody>
</table>
IFPUG External Input – Example

- Enter a new employee with Monthly Payment:
  - Name,
  - ID,
  - Birth Date,
  - Payment Reference,
  - Salary Level.

<table>
<thead>
<tr>
<th>EI</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-4</td>
</tr>
<tr>
<td>1</td>
<td>low (3)</td>
</tr>
<tr>
<td>2</td>
<td>low (3)</td>
</tr>
<tr>
<td>&gt; 2</td>
<td>average (4)</td>
</tr>
</tbody>
</table>
IFPUG External Input – Example

- Enter a new employee with Weekly Payment:
  - Name,
  - ID,
  - Birth Date,
  - Payment Reference,
  - Hourly Rate,
  - Payment Office.

<table>
<thead>
<tr>
<th>EI</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-4</td>
</tr>
<tr>
<td>#FTR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>low (3)</td>
</tr>
<tr>
<td>2</td>
<td>low (3)</td>
</tr>
<tr>
<td>&gt; 2</td>
<td>average (4)</td>
</tr>
</tbody>
</table>

ILF: Employee Administration (DB)

Employee
- Name
- ID
- Birth Date
- Payment Reference

Weekly Payment
- Hourly Rate
- Payment Office

Monthly Payment
- Salary Level

1 FTR
6 DET
IFPUG External Output – Complexity Assessment

- Identify number of File Types Referenced (FTR)
- Identify number of Data Element Type (DET)
- Determine complexity (→ used to calculate FP count)

<table>
<thead>
<tr>
<th>EO</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-5</td>
</tr>
<tr>
<td></td>
<td>6-19</td>
</tr>
<tr>
<td></td>
<td>&gt; 19</td>
</tr>
<tr>
<td>#FTR</td>
<td>low (4)</td>
</tr>
<tr>
<td>1</td>
<td>low (4)</td>
</tr>
<tr>
<td>2-3</td>
<td>average (5)</td>
</tr>
<tr>
<td>&gt; 3</td>
<td>average (5)</td>
</tr>
</tbody>
</table>
IFPUG External Output – Example

- Report of all Employees containing Names and Birth Dates, sorted by age.

**ILF: Employee**
- Name
- ID
- Birth Date
- Payment Reference

≤ 40 Years
- James Miller, 10-02-1966
- Angela Rhodes, 05-03-1966

≤ 45 Years
- Mike Smith, 23-03-1961

<table>
<thead>
<tr>
<th>EO</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>6-19</td>
</tr>
<tr>
<td>#FTR</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>low (4)</td>
</tr>
<tr>
<td>2-3</td>
<td>low (4)</td>
</tr>
<tr>
<td>&gt; 3</td>
<td>average (5)</td>
</tr>
</tbody>
</table>

1 FTR
2 DET
IFPUG External Inquiry – Complexity Assessment

- Identify number of File Types Referenced (FTR)
- Identify number of Data Element Type (DET)
- Determine complexity (→ used to calculate FP count)

<table>
<thead>
<tr>
<th>EQ</th>
<th>#DET</th>
<th>1-5</th>
<th>6-19</th>
<th>&gt; 19</th>
</tr>
</thead>
<tbody>
<tr>
<td>#FTR</td>
<td>1</td>
<td>low (3)</td>
<td>low (3)</td>
<td>average (4)</td>
</tr>
<tr>
<td></td>
<td>2-3</td>
<td>low (3)</td>
<td>average (4)</td>
<td>high (6)</td>
</tr>
<tr>
<td></td>
<td>&gt; 3</td>
<td>average (4)</td>
<td>high (6)</td>
<td>high (6)</td>
</tr>
</tbody>
</table>
IFPUG External Inquiry – Example

- Report of all employees belonging to Department X containing Names, Birth Dates, and showing the Department Name.
- Files (ILF): Employee, Department
- 2 FTR: Employee, Department
- 3 DET: Name (Employee), Birth Date (Employee), Department Name (Department)

<table>
<thead>
<tr>
<th>EQ</th>
<th>#DET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-5</td>
</tr>
<tr>
<td>1</td>
<td>low (3)</td>
</tr>
<tr>
<td>2-3</td>
<td>low (3)</td>
</tr>
<tr>
<td>&gt; 3</td>
<td>average (4)</td>
</tr>
</tbody>
</table>
IFPUG FP Counting Example – GUI

1 External Inquiry (input side)

External Input (new, 1 FTR, 7 DET, low) = 3 FP
External Input (change, 1 FTR, 7 DET, low) = 3 FP
External Input (delete, 1 FTR, 7 DET, low) = 3 FP
External Inquiry (navigate, 1 FTR, 7 DET, low) = 3 FP

3 External Inputs

12 FP
IFPUG Function Point Counting Big Picture

Step 1a

Step 1b

Step 2

Weighted EI + Weighted EO + Weighted EQ + Weighted EIF + Weighted ILF = Unadjusted Function Points (UFP)

Weighting of functional (technical) complexities

Unadjusted and Unweighted Function Count

# EI + # EO + # EQ + # EIF + # ILF

14 Adjustment Factors

Complexity Adjustment

Value Adjustment Factor (VAF)

Adjusted Function Points

Unadjusted Function Points

Value Adjustment Factor (VAF)
IFPUG Value Adjustment Factor (VAF) /1

- Value Adjustment Factor (VAF) is a weighted sum of 14 Factors (General System Characteristics (GSC) or Technical Complexity Adjustments (TCA)):

<table>
<thead>
<tr>
<th>F1</th>
<th>Data Communications</th>
<th>F2</th>
<th>Distributed Data Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>Performance</td>
<td>F4</td>
<td>Heavily Used Configuration</td>
</tr>
<tr>
<td>F5</td>
<td>Transaction Rate</td>
<td>F6</td>
<td>On-line Data Entry</td>
</tr>
<tr>
<td>F7</td>
<td>End-User Efficiency</td>
<td>F8</td>
<td>On-line Update</td>
</tr>
<tr>
<td>F9</td>
<td>Complex Processing</td>
<td>F10</td>
<td>Reusability</td>
</tr>
<tr>
<td>F11</td>
<td>Installation Ease</td>
<td>F12</td>
<td>Operational Ease</td>
</tr>
<tr>
<td>F13</td>
<td>Multiple Sites</td>
<td>F14</td>
<td>Facilitate Change</td>
</tr>
</tbody>
</table>
IFPUG Value Adjustment Factor (VAF) /3

- Each component is rated from 0 to 5, where 0 means the component is not relevant to the system and 5 means the component is essential.

- The VAF can then be calculated as:

\[
VAF = 0.65 + 0.01 \sum_{j=1}^{14} F_j
\]

- The VAF varies from 0.65 (if all \(F_j\) are set to 0) to 1.35 (if all \(F_j\) are set to 5)
Case Study
(20 min)
FP: Advantages – Summary

• Can be counted before design or code documents exist (but a specification is needed)
• Can be used for estimating project cost, effort, schedule early in the project life-cycle
• Helps with contract negotiations
• Is standardized (though several competing standards exist)
FP: Limitations – Summary

• FP is a subjective measure (→ counting items, complexity weights, adjustment factors)
• Requires a full software system specification
  – But there exist “light-weight” alternatives → Object Point, Use Case Point
• Appropriateness of the basic unit of FP is unclear
• Difficult to apply to maintenance (enhancement) projects
• IFPUG FP not suitable for “complex” software, e.g., real-time and embedded applications
  – Proposed solution: COSMIC FP / Feature Points
Project Time/Effort Estimation:

COCOMO
Estimation Techniques – Main Types

Cost/Schedule Estimation Techniques

- Algorithmic/Parametric Models
  - Constraint Models:
    - SLIM
    - Jensen Model
    - COPMO
    - etc.
  - Empirical Factor Models:
    - COCOMO / COCOMO II
    - ESTIMACS
    - PRICE S
    - Softcost
    - DOTY
    - CheckPoint
    - etc.

- Expert Judgment

- Analogy

- Machine Learning:
  - Case-Based Reasoning
  - Collaborative Filtering
  - Classification Systems
  - etc.

- Other
  - Parkinson’s Law
  - Pricing-to-Win
  - Top-Down Estimation
  - Bottom-Up Estimation

- Delphi-Method
- Planning Poker
- etc.
Parametric Models – Simple Examples

- Estimated effort = Estimated system size / Productivity1
  - E.g., system size estimated in kloc (kilo lines of code)
  - E.g., productivity calculated as average from past projects:
    - Productivity1_i = System size_i [kloc] / Effort_i [person-day]
- Estimated time = Estimated system size / Productivity2
  - Productivity2_i = (System size_i [kloc]) / Time_i [day]
Parametric model example: COCOMO

- COCOMO = Constructive Cost Model
- Size measure (KLOC, FP) & productivity factors as input
COCOMO – Introduction

- **COCOMO (Constructive Cost Model) – 1981**
  - Model based on
    - estimated size of the new system and
    - cost drivers that affect productivity.

- **COCOMO II – 2000**
  - Enhanced version, accounts for changes in software engineering technology, including
    - object-oriented software,
    - software created following spiral or evolutionary development models,
    - software reuse,
    - components off-the-shelf (COTS).

Prof. Barry Boehm, USC
COCOMO – 1981

- The original COCOMO is a collection of three models:
  - Basic model – applied during feasibility study
  - Intermediate model – applied after requirements acquisition
  - Advanced model – applied after design is complete

- All three models take the form:

\[ E = aS^b \times EAF \]
\[ T_{dev} = cE^d \]

where:
- \( E \) is effort in person-months
- \( T_{dev} \) is the development time in months
- \( S \) is size measured in thousands of lines of code (or DSI*)
- \( EAF \) is the effort adjustment factor (equals 1 in the Basic model)
- Factors \( a, b, c \) and \( d \) depend on the development mode.

*DSI = Delivered Source Instructions
Basic COCOMO – Effort Equations

- **Organic mode:** \( PM = 2.4 \ (KDSI)^{1.05} \)
- **Semi-detached mode:** \( PM = 3.0 \ (KDSI)^{1.12} \)
- **Embedded mode:** \( PM = 3.6 \ (KDSI)^{1.20} \)

\[
E_{\text{nom}} = a(\text{Size})^b
\]

\[
EAF = 1
\]

<table>
<thead>
<tr>
<th>Mode</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organic</td>
<td>2.4</td>
<td>1.05</td>
</tr>
<tr>
<td>Semi-detached</td>
<td>3.0</td>
<td>1.12</td>
</tr>
<tr>
<td>Embedded</td>
<td>3.6</td>
<td>1.20</td>
</tr>
</tbody>
</table>

PM = Person-Months

KDSI = Kilo Delivered Source Instructions (~KLOC)
COCOMO II
COCOMO II
Accuracy Funnel

Relative Size Range

Operational Concept
Life Cycle Objectives
Life Cycle Architecture
Initial Operating Capability

Phases and Milestones

Feasibility
Plans/Rqts.
Design
Develop and Test

4x
2x
0.5x
0.25x

x
COCOMO II – Three Sub-Models

- COCOMO II includes a three-stage series of models:
  - **Application Composition Model:** Used during the earliest development phases, usually involving prototyping
  - **Early Design Model:** Used during the next development phases, usually involving exploration of architectural alternatives or incremental development strategies
  - **Post-Architecture Model:** Used once the project has a defined life-cycle architecture, typically providing more accurate and detailed information for adjusting cost drivers
COCOMO II – Early Design Model (EDM)

- The Early Design model equation is:
  \[ E = a \times (KLOC)^b \times EAF \]
  \[ b = 0.91 + 0.01 \sum_{j=1}^{5} SF_i \]

- The effort adjustment factor (EAF) is the product of seven effort multipliers.
- SF\(_i\): Scale Factor i
- \( a = 2.94 \)

- If unadjusted function point count (UFC) is used for sizing, this value is converted to KLOC.
COCOMO II – EDM Effort Multipliers (Cost Drivers)

• The effort adjustment factor (EAF) for the Early Design Model is the product of 7 effort multipliers/modifiers (which are further refined into 17 cost drivers in the Post-Architecture model)

<table>
<thead>
<tr>
<th>Cost Driver</th>
<th>Description</th>
<th>Represent Combined Post-Architecture Cost Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RCPX</td>
<td>Product reliability and complexity</td>
</tr>
<tr>
<td>2</td>
<td>RUSE</td>
<td>Required reusability</td>
</tr>
<tr>
<td>3</td>
<td>PDIF</td>
<td>Platform difficulty</td>
</tr>
<tr>
<td>4</td>
<td>PERS</td>
<td>Personnel capability</td>
</tr>
<tr>
<td>5</td>
<td>PREX</td>
<td>Personnel experience</td>
</tr>
<tr>
<td>6</td>
<td>FCIL</td>
<td>Facilities available</td>
</tr>
<tr>
<td>7</td>
<td>SCED</td>
<td>Schedule pressure</td>
</tr>
</tbody>
</table>
### COCOMO II – EDM Effort Multipliers (Cost Drivers)

<table>
<thead>
<tr>
<th></th>
<th>Extra low</th>
<th>Very low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very high</th>
<th>Extra high</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCPX</td>
<td>0.49</td>
<td>0.60</td>
<td>0.83</td>
<td>1.00</td>
<td>1.33</td>
<td>1.91</td>
<td>2.72</td>
</tr>
<tr>
<td>RUSE</td>
<td></td>
<td>0.95</td>
<td>1.00</td>
<td>1.07</td>
<td>1.15</td>
<td>1.24</td>
<td></td>
</tr>
<tr>
<td>PDIF</td>
<td></td>
<td>0.87</td>
<td>1.00</td>
<td>1.29</td>
<td>1.81</td>
<td>2.61</td>
<td></td>
</tr>
<tr>
<td>PERS</td>
<td>2.12</td>
<td>1.62</td>
<td>1.26</td>
<td>1.00</td>
<td>0.83</td>
<td>0.63</td>
<td>0.50</td>
</tr>
<tr>
<td>PREX</td>
<td>1.59</td>
<td>1.33</td>
<td>1.12</td>
<td>1.00</td>
<td>0.87</td>
<td>0.74</td>
<td>0.62</td>
</tr>
<tr>
<td>FCIL</td>
<td>1.43</td>
<td>1.30</td>
<td>1.10</td>
<td>1.00</td>
<td>0.87</td>
<td>0.73</td>
<td>0.62</td>
</tr>
<tr>
<td>SCED</td>
<td>1.43</td>
<td>1.14</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td></td>
</tr>
</tbody>
</table>

**Default**
COCOMO II – Post-Architecture Model (PAM)

• The Post-Architecture model equation is:

\[ E = a \times (KLOC)^b \times EAF \quad b = 0.91 + 0.01 \sum_{j=1}^{5} SF_i \]

• a = 2.94

• SF\textsubscript{i}; Scale Factor i

• The effort adjustment factor (EAF) is the product of 17 effort multipliers.

• Descriptions of the Effort Multipliers (Cost Drivers) can be found in the COCOMO II Model Definition Manual posted on the course wiki (see Section 3.2)
COCOMO II – PAM Effort Multipliers (Cost Drivers)

Product Factors
- Reliability (RELY)
- Data (DATA)
- Complexity (CPLX)
- Reusability (RUSE)
- Documentation (DOCU)

Personnel factors
- Analyst capability (ACAP)
- Program capability (PCAP)
- Applications experience (APEX)
- Platform experience (PLEX)
- Language and tool experience (LTEX)
- Personnel continuity (PCON)

Platform Factors
- Time constraint (TIME)
- Storage constraint (STOR)
- Platform volatility (PVOL)

Project Factors
- Software tools (TOOL)
- Multisite development (SITE)
- Required schedule (SCED)
COCOMO II – Example Effort Multiplier
Required Software Reliability (RELY)

• Measures the extent to which the software must perform its intended function over a period of time.
• Ask: what is the effect of a software failure?

<table>
<thead>
<tr>
<th>RELY</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>slight inconvenience</td>
<td>low, easily recoverable losses</td>
<td>moderate, easily recoverable losses</td>
<td>high financial loss</td>
<td>risk to human life</td>
<td></td>
</tr>
</tbody>
</table>
COCOMO II – Scale Factors (EDM & PAM)

Scale Factors (SF):

- Precedentedness (PREC) – [0 .. 6.20]
  - Degree to which system is new and past experience applies
- Development Flexibility (FLEX) – [0 .. 5.07]
  - Need to conform with specified requirements
- Architecture/Risk Resolution (RESL) – [0 ... 7.07]
  - Degree of design thoroughness and risk elimination
- Team Cohesion (TEAM) – [0 .. 5.48]
  - Need to synchronize stakeholders and minimize conflict
- Process Maturity (PMAT) – [0 .. 7.80]
  - SEI CMM process maturity rating

\[
b = 0.91 + 0.01 \sum_{j=1}^{5} SF_j
\]
### COCOMO II – Scale Factor Classification

<table>
<thead>
<tr>
<th>Scale Factors (Wi)</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedentedness (PREC)</td>
<td>thoroughly unprecedented</td>
<td>largely unprecedented</td>
<td>somewhat unprecedented</td>
<td>generally familiar</td>
<td>largely familiar</td>
<td>thoroughly familiar</td>
</tr>
<tr>
<td>Development Flexibility (FLEX)</td>
<td>rigorous</td>
<td>occasional relaxation</td>
<td>some relaxation</td>
<td>general conformity</td>
<td>some conformity</td>
<td>general goals</td>
</tr>
<tr>
<td>Architecture/Risk Resolution (RESL)*</td>
<td>little (20%)</td>
<td>some (40%)</td>
<td>often (60%)</td>
<td>generally (75%)</td>
<td>mostly (90%)</td>
<td>full (100%)</td>
</tr>
<tr>
<td>Team Cohesion (TEAM)</td>
<td>very difficult interactions</td>
<td>some difficult interactions</td>
<td>basically cooperative interactions</td>
<td>largely cooperative</td>
<td>highly cooperative</td>
<td>seamless interactions</td>
</tr>
<tr>
<td>Process Maturity (PMAT)</td>
<td>Weighted average of “Yes” answers to CMM Maturity Questionnaire</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* % significant module interfaces specified, % significant risks eliminated

- Sum scale factors $SF_i$ to determine a scale exponent, $b$, with $b = 0.91 + 0.01 \sum SF_i$
COCOMO II – Scale Factor Ratings

- From Class to Value

<table>
<thead>
<tr>
<th>SF</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precedentedness</td>
<td>PREC</td>
<td>6.20</td>
<td>4.96</td>
<td>3.72</td>
<td>2.48</td>
<td>1.24</td>
</tr>
<tr>
<td>Development/Flexibility</td>
<td>FLEX</td>
<td>5.07</td>
<td>4.05</td>
<td>3.04</td>
<td>2.03</td>
<td>1.01</td>
</tr>
<tr>
<td>Architecture/Risk Resolution</td>
<td>RESL</td>
<td>7.07</td>
<td>5.65</td>
<td>4.24</td>
<td>2.83</td>
<td>1.41</td>
</tr>
<tr>
<td>Team Cohesion</td>
<td>TEAM</td>
<td>5.48</td>
<td>4.38</td>
<td>3.29</td>
<td>2.19</td>
<td>1.10</td>
</tr>
<tr>
<td>Process Maturity</td>
<td>PMAT</td>
<td>7.80</td>
<td>6.24</td>
<td>4.68</td>
<td>3.12</td>
<td>1.56</td>
</tr>
</tbody>
</table>

Default

\[
b = 0.91 + 0.01 \sum_{j=1}^{5} SF_i\]
COCOMO II – Impact of SFs on Effort

\[ E = a \times (KLOC)^b \times EAF \]

- \([0 \ldots 6.20]\)
- \([0 \ldots 5.07]\)
- \([0 \ldots 7.07]\)
- \([0 \ldots 5.48]\)
- \([0 \ldots 7.80]\)

\[ b = 0.91 + 0.01 \sum_{i=1}^{5} SF_i \]
Example: Scale Factor TEAM

- Elaboration of the TEAM rating scale:
  - Use a subjective weighted average of the characteristics to account for project turbulence and entropy due to difficulties in synchronizing the project's stakeholders.
  - Stakeholders include users, customers, developers, maintainers, interfacers, and others

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Very Low</th>
<th>Low</th>
<th>Nominal</th>
<th>High</th>
<th>Very High</th>
<th>Extra High</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consistency of stakeholder objectives and cultures</td>
<td>Little</td>
<td>Some</td>
<td>Basic</td>
<td>Considerable</td>
<td>Strong</td>
<td>Full</td>
</tr>
<tr>
<td>Ability, willingness of stakeholders to accommodate other stakeholders' objectives</td>
<td>Little</td>
<td>Some</td>
<td>Basic</td>
<td>Considerable</td>
<td>Strong</td>
<td>Full</td>
</tr>
<tr>
<td>Experience of stakeholders in operating as a team</td>
<td>None</td>
<td>Little</td>
<td>Little</td>
<td>Basic</td>
<td>Considerable</td>
<td>Extensive</td>
</tr>
<tr>
<td>Stakeholder teambuilding to achieve shared vision and commitments</td>
<td>None</td>
<td>Little</td>
<td>Little</td>
<td>Basic</td>
<td>Considerable</td>
<td>Extensive</td>
</tr>
</tbody>
</table>
Case Study

(20 min)
Global Software Development
Global Software Development

Main issue:
- distance matters
Co-located versus global/multi-site

- Co-located: housed within walking distance
  - People re-invent the wheel if they have to walk more than 30 meters, or climb the stairs

- Main question: How to bridge distance in global projects?
  - Communication
  - Coordination
  - Control
Arguments for global software development

• Cost savings (salary)

• Faster delivery (“follow the sun”)

• Larger pool of developers

• Better modularization (of responsibilities)

BUT: Little empirical evidence that these advantages materialize
### Challenges

<table>
<thead>
<tr>
<th></th>
<th>temporal</th>
<th>geographical</th>
<th>sociocultural</th>
</tr>
</thead>
<tbody>
<tr>
<td>communication</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>coordination</td>
<td>X</td>
<td>X</td>
<td>(X)</td>
</tr>
<tr>
<td>control</td>
<td>X</td>
<td>(X)</td>
<td>X</td>
</tr>
</tbody>
</table>

Distance types: temporal, geographical, sociocultural.
Temporal distance challenges

• Communication:
  – Being effective (asynchronous is less effective, misunderstandings, …)

• Coordination:
  – Cost is larger (travels, infrastructure cost, …)

• Control:
  – Delays (wait for next teleconference meeting, send email and wait, search for contact, …)
Geographical distance challenges

• Communication:
  – Effective information exchange (less informal exchange, different languages, different domain knowledge, …)
  – Build a team (cohesiveness, “them and us” feelings, trust, …)

• Coordination:
  – Task awareness (shared mental model, …)
  – Sense of urgency (perception, …)

• Control:
  – Accurate status information (tracking, blaming, …)
  – Uniform process (but different tools and techniques, …)
Sociocultural distance challenges

• Communication:
  – Cultural misunderstandings (corporate, technical, national, …)
• Coordination:
  – Effectiveness (vocabulary, communication style, …)
• Control:
  – Different control/feedback styles (more or less hierarchical; more or less formal)
National culture

• American managers have a hamburger style of management.
  – They start with sweet talk – the top of the bun.
  – Then the criticism is slipped in – the meat.
  – Finally, some encouraging words – the bottom bun.

• With the Germans, all one gets is the meat.

• With the Japanese, all one gets is the bun; one has to smell the meat.
Hofstede’s Cultural Dimensions (National)

- **Power distance (PDI)**
  - status is important versus individuals are equal

- **Collectivism versus individualism (IDV)**
  - Individuals are part of a group, or everyone looks after himself

- **Uncertainty avoidance (UAI)**
  - Strict rules that mitigate uncertainty versus flexibility/pragmatism

- **Femininity versus masculinity (MAS)**
  - Challenges, recognition, materialism (masculine) versus good relationships, cooperation, security, quality of life (feminine)

- **Long-term versus short-term orientation (LTO)**
  - Persistence in pursuing goals, order, capacity for adaptation (LT) versus protecting one’s face, tradition, fulfilling social obligations, rewarding past and present behavior (ST)

- **Indulgence versus restraint (IND)**
  - Indulgence stands for a society that allows relatively free gratification of basic and natural human drives related to enjoying life and having fun. Restraint stands for a society that suppresses gratification of needs and regulates it by means of strict social norms.

https://geert-hofstede.com/national-culture.html
Power distance

- In cultures that endorse high power distance, less powerful accept power relations that are more autocratic and paternalistic.
  - Subordinates acknowledge the power of others simply based on where they are situated in certain formal, hierarchical positions.

- Cultures that endorse low power distance expect and accept power relations that are more consultative or democratic.
  - People relate to one another more as equals regardless of formal positions.

Power Distance Index (PDI) for countries

https://geert-hofstede.com/national-culture.html
Collectivism versus individualism

The degree to which individuals are integrated into groups.

- In individualistic societies, the stress is put on personal achievements and individual rights.
  - People are expected to stand up for themselves and their immediate family, and to choose their own affiliations.

- In collectivist societies, individuals act predominantly as members of a lifelong and cohesive group or organization.
  - People have large extended families, which are used as a protection in exchange for unquestioning loyalty.
Uncertainty avoidance

- Low uncertainty avoidance (UAI):
  - can better cope with uncertainty: they can deal with agile approaches, ill-defined requirements, etc.
- High uncertainty avoidance:
  - favor waterfall, contracts, etc.

- Latin America, Japan: high UAI
- North America, India: low UAI
Exercise
(10 min)
How to overcome distance?

• Common ground
• Coupling of work
• Collaboration readiness
• Technology readiness
Common ground

• How much common knowledge members have, and are aware of
• Common ground has to be established:
  – Traveling, especially at start of project
  – Socialization (kick-off meetings)

• Intense interaction is more important for success than any other factor (e.g., organisational maturity)
Coupling of work

- Tasks that require much collaboration:
  - at same site

- Little interaction required:
  - different sites

- Example: testing or implementing relatively independent subsystems
Collaboration readiness

- Transition to global development organization:
  - Requires changing work habits
  - People must embrace difference as an opportunity
  - Learning new tools
  - Needs incentives for individuals to cooperate
Technology readiness

- Project management tools (workflow management)
- Web-enabled versions of tools
- Remote control of builds and tests
- Web-based project repositories
- Real-time collaboration tools (simple media for simple messages, rich media for complex ones)
- Knowledge management technology (codification AND personalization)
Organizing work in global software development

• Reduce the need for informal communication
  – Usually through organizational means, e.g.:
    • Put user interface people together
    • Use gross structure (architecture) to divide work
      – Conway’s Law: “Organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations.” [1968]
    • Split according to life cycle phases
• Provide technologies that ease informal communication
Summary

• Distance matters

• Main challenges:
  – Deal with lack of informal communication
  – Handle cultural differences
Preparation for Quiz 4
Quiz 4 – Formalities

• Start: Tuesday, Dec 6 at 12:00
• End: Monday, Dec 12 at 16:00 (sharp!)
• 2 attempts, the better one counts
• 15 min time per attempt
• Randomly generated questions (equivalent difficulty)
• 5 questions / 0.6 points each / 3 points maximum

--

• Tip: Don’t start the quiz around midnight (+/- 15 min) – there seems to be maintenance work done which will spoil your quiz attempt!
Quiz 4 – Content and Format

- 5 categories (1 question per category)
  - Release Planning
  - User Story Points
  - Use Case Points
  - Function Point Analysis
  - Cost Estimation with COCOMO II

- 4 Choices with exactly 1 correct answer per question
- Some questions require simple calculations
Example Questions

(10 min)
Quiz 4 – RP Example Question

Which of the following statements about being a good release plan is incorrect:

Answer choices:

a) A good release plan provides maximum business value
b) A good release plan satisfies the most important stakeholders involved
c) A good release plan is feasible with available resources
d) A good release plan may or may not take feature dependencies under consideration
Quiz 4 – USP Example Question

Assume, you have estimated 20 user stories (US1-US20) to have a difficulty/complexity expressed in story points as follows:

- 2 story points: each of US1 to US5
- 3 story points: each of US6 to US10
- 7 story points: each of US11 to US15
- 13 story points: each of US16 to US20

Assume that all user stories have the equal value for the end user. If you have a team of 4 developers and weekly sprints (1 week = 5 days = 40 hours), which user stories would you implement in the sprint if you don’t allow overtime?

Assume that in the past, on average, one developer could implement 1% of the sum of user story points of your current project in one day.

Answer choices: a) ... b) ... c) ... d) ... <List of user stories>
Quiz 4 – UCP Example Question

Assume a use case diagram containing five actors (A1, …, A5) and three use cases (UC1, UC2, and UC3):
- Actors A1 and A2 are of type ‘complex’, all other actors are of type ‘average’
- UC1 and UC3 have 2 transactions each
- UC2 has 4 transactions
- The Technical Complexity Factor equals ‘1’
- All eight Environmental Factors equal ‘3’

What is the number of adjusted Function Points and what is the Project Effort?

Answer choices: a) … b) … c) … d) … <UCP number & Effort>
Quiz 4 – FPA Example Question

An input screen associated with an EI transaction named 'New Customer' contains the following GUI fields and buttons:
- field 1: 'Customer name'
- field 2: 'Contact details'
- field 3: 'Phone number'
- field 4: 'Billing address'
- field 5: 'Shipping address'
- field 6 (offering a pull down menu with 3 pre-defined types): 'Customer type'
- button 1: 'OK'
- button 2: 'Cancel'
- button 3: 'Back'

How many DETs does the EI contain?

Answer choices: a) 11 DETs b) 9 DETs c) 7 DETs d) 12 DETs
Quiz 4 – COCOMO II Example Question

Assume you are using the COCOMO II post-architecture model (with 5 scale drivers and 17 cost drivers) to estimate the development effort for a new software system.

In the following, assume this:
- All cost drivers are set to 'Nominal' before you do the first estimate and afterwards never change.
- All scale drivers are set before you do the first estimate; once they have been set, they never change.

Now assume that one person estimated the size of the software system as 100 FPs (function points), another person estimated it to be 120 FPs.
For each of the FP counts you produce an effort estimate with COCOMO II, i.e. Estimate_1 (using 100 FP) and Estimate_2 (using 120 FP).

How must the scale factors be set in order to guarantee that Estimate_2 < 1.2 * Estimate_1?
Quiz 4 – COCOMO II Example Question (cont’d)

How must the scale factors be set in order to guarantee that $\text{Estimate}_2 < 1.2 \times \text{Estimate}_1$?

Answer choices:

a) All scale drivers must be set to 'Extra High'
b) All scale drivers must be set to 'Very Low'
c) It is impossible to guarantee that $\text{Estimate}_2 < 1.2 \times \text{Estimate}_1$
d) It doesn't matter how you set the scale drivers, it will always be guaranteed that $\text{Estimate}_2 < 1.2 \times \text{Estimate}_1$