Class Announcements

• Mid Term Exam - Feedback
• Next Assignment – Due 10/20 Tuesday
• Presentations by Teams for the Analysis Model Tuesday 10/20
Recap – 1 to 5 Chapters

1. UML Notation
   1. Use Case
   2. Class Diagrams
   3. Interaction or Sequence Diagrams
   4. Machine or State Diagrams
   5. Activity Diagrams

2. Requirement Elicitation
   1. Use Case, Initial Objects, Relationship between objects
   2. Non-Functional Requirements
   3. Requirement Analysis Document

3. System Analysis
   1. Entity, Boundary and Control Objects
   2. Analysis Object Models, Dynamic Object Models
   3. Use case mapping, Associations, Aggregations, Attributes
   4. Managing Change
Function Point  Cost Model
Function Point – Cost Model or Analysis (FPC)

- Developed by Alan Albrecht Of IBM – 1979.
- FPC is a method used to measure various cost in a software application.
- FPC measure the functionality from the user point of view
- FPC is based on User Input and the Output expected by system
- FPC is a simple tool takes parameters to calculate the over all software cost
Steps involved in FPC Method

• Identify system boundary
• Boundary defines the functions included in the FPC
## FPC – Weighting Factor Estimate

<table>
<thead>
<tr>
<th>Measurement Parameters</th>
<th>Count</th>
<th>Simple</th>
<th>Average</th>
<th>Complex</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of User Inputs</td>
<td>X</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Number of User Outputs</td>
<td>X</td>
<td>4</td>
<td>5</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Number of User Inquires</td>
<td>X</td>
<td>3</td>
<td>4</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>Number of Internal Files</td>
<td>X</td>
<td>7</td>
<td>10</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>Number of External Interfaces of files</td>
<td>X</td>
<td>5</td>
<td>7</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td><strong>Grand Total (FP)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>=</strong></td>
</tr>
</tbody>
</table>
# FPC – Rating Estimate of categories

<table>
<thead>
<tr>
<th>Category</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Does the system require reliable backup and recovery?</td>
<td>Rating</td>
</tr>
<tr>
<td>2 Are data communications required?</td>
<td>Rating</td>
</tr>
<tr>
<td>3 Are there distributed processing functions?</td>
<td>Rating</td>
</tr>
<tr>
<td>4 Is performance critical?</td>
<td>Rating</td>
</tr>
<tr>
<td>5 Will the system run in an existing, heavily utilized operational environment?</td>
<td>Rating</td>
</tr>
<tr>
<td>6 Does the system require on-line data entry?</td>
<td>Rating</td>
</tr>
<tr>
<td>7 Does the on-line data entry require the input transaction to be built over multiple screens or operations?</td>
<td>Rating</td>
</tr>
<tr>
<td>8 Are the master files updated on-line?</td>
<td>Rating</td>
</tr>
<tr>
<td>9 Are the inputs, outputs, files or inquiries complex?</td>
<td>Rating</td>
</tr>
<tr>
<td>10 Is the internal processing complex?</td>
<td>Rating</td>
</tr>
<tr>
<td>11 Is the code designed to be reusable?</td>
<td>Rating</td>
</tr>
<tr>
<td>12 Are conversion and installation included in the design?</td>
<td>Rating</td>
</tr>
<tr>
<td>13 Is the system designed for multiple installations in different organizations?</td>
<td>Rating</td>
</tr>
<tr>
<td>14 Is the application designed to facilitate change and ease of use by the user?</td>
<td>Rating</td>
</tr>
</tbody>
</table>

**Ratings Scale:**
- 0 – No
- 1 – Incidental
- 2 – Moderate
- 3 – Average
- 4 – Significant
- 5 - Essential

**Total sum of all category ratings**
FPC – Model Formula

FPC = Grand Total FP * [0.65 + 0.01*(sum of all category ratings)]
SYSTEM DESIGN
System Design – System Decomposition

• Migration from Analysis to System Design.
• Define *Design Goals*
• Design Initial *Sub System* Decomposition.
• *Refine* the Sub System to address the Design Goals.
System

System Design

Object Design

Implementation
Activities of System Design

This phase will produce the following

1. Design Goals
2. Software Architecture
3. Boundary Use Cases, Exceptions, hardware configurations
Design Goal:

• Design Goals come from Non-Functional Requirements.
• Trade Off decisions made.
• Sub System Decomposition is the bulk of the System Design.
• Developers divide the system into manageable parts to reduce complexity.
• Each Sub-System is assigned to a smaller teams.
What is a Sub System?

• A **subsystem** is a *replaceable* part of the system with well-defined interfaces that *encapsulates* the state and behavior of its *contained* classes.

• To reduce the complexity of the **Solution Domain**, a System is divided into smaller systems *called* **Sub System**.
More about Sub Systems....

• Service : Is a related operations that share a common purpose.

• Define the Sub System in terms of the services they provide.
Questions?
References

• Use Cases Combined with BOOCH, OMT UML Process and Products
  - Putnam P Texel, Charles B Williams

• Object-Oriented Software Engineering Using UML Patterns, and JAVA
  - Bernd Bruegge & Allen H. Dutoit

• Software Engineering 9th Edition
  - Ian Sommerville