Software Engineering
Design Process

Copyright © 1998 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited
Today’s Lecture

1. Intro to Software Engineering
2. Inexact quantities
3. Error propagation
4. Floating-point numbers
5. Design process
6. Teamwork
7. Project planning
8. Decision making
9. Professional Engineering
10. Software quality
11. Software safety
12. Intellectual property
Design Models and Processes

An **engineering design** is a model of the product or structure to be engineered. The model is used to

- Evaluate suitability of proposed product/system
- Communicate proposed product to others

An **engineering design process** describes a set of steps for constructing an engineering design.
Agenda

• SE process vs. Engineering design process

• Software Engineering processes
  • Waterfall model
  • Concurrent engineering
  • Spiral model
  • Agile methods
Engineering Design Process

1. Recognition of Need
2. Definition of the Design Problem
3. Design Criteria and Constraints
4. The Design Loop
   - Synthesis
   - Analysis
   - Decision-Making
5. Optimization
6. Evaluation
7. Communication (Drawings, reports)

Engineering Development Process

1. Recognition of Need

2. Definition of the Design Problem

3. Design Criteria and Constraints

4. The Design Loop
   - Synthesis
   - Analysis
   - Decision-Making

5. Optimization

6. Evaluation

7. Communication (Drawings, reports)

8. Manufacturing

9. Quality Control

10. Field/customer service
1. Waterfall Model

1. Requirements
2. Specification
3. Architectural Design
4. Detailed Design
5. Implementation
6. Testing
7. Maintenance
Waterfall Model

1. Requirements
   Understand the problem

2. Specification
   Characterize acceptable solution

3. Architectural Design
   Decompose solution into subsystems

4. Detailed Design
   Design subsystems

5. Implementation
   Make code-level decisions

6. Testing
   Test that implementation is acceptable

7. Maintenance
   Fix bugs, add features
Waterfall Model

1. Requirements
2. Specification
3. Architectural Design
4. Detailed Design
5. Implementation
6. Testing
7. Maintenance
Eng. Process vs. SE Process

1. Recognition of Need
2. Definition of the Design Problem
3. Design Criteria and Constraints
4. The Design Loop
   - Synthesis
   - Analysis
   - Decision-Making
5. Optimization
6. Evaluation
7. Communication (Drawings, reports)
8. Manufacturing
9. Quality Control
10. Field/customer service

1. Requirements
2. Specification
3. Architectural Design
4. Detailed Design
5. Implementation
6. Testing
7. Maintenance
2. Concurrent Engineering

Requirements
Specification
Design
Implementation
Testing

\( time \)
3. Spiral Model

- Requirements/Specification
- Coding
- Design
- Testing
4. Agile Methods

3. Spiral Model

- Requirements/Specification
- Design
- Coding
- Testing

4. Agile methods (e.g., Extreme Programming)

- Test Case Definition
- Pair Programming
- Refactor
- Design
- Testing
4. Agile Methods (Extreme Programming)

Test Case Definition

Pair Programming

Testing

Refactor

Design
# Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td>Rapid development</td>
<td>High quality</td>
</tr>
<tr>
<td>Project Size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Developers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td>Rapid development</td>
<td>High quality</td>
</tr>
<tr>
<td>Project Size</td>
<td>Smaller teams and products</td>
<td>Larger teams and products</td>
</tr>
<tr>
<td>Developers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td>Rapid development</td>
<td>High quality</td>
</tr>
<tr>
<td>Project Size</td>
<td>Smaller teams and products</td>
<td>Larger teams and products</td>
</tr>
<tr>
<td>Developers</td>
<td>Agile, knowledgeable, talented, collocated, collaborative</td>
<td>Plan-oriented, adequately skilled, have access to external knowledge</td>
</tr>
<tr>
<td>Project Requirements</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Product Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td>Rapid development</td>
<td>High quality</td>
</tr>
<tr>
<td>Project Size</td>
<td>Smaller teams and products</td>
<td>Larger teams and products</td>
</tr>
<tr>
<td>Developers</td>
<td>Agile, knowledgeable, talented, collocated, collaborative</td>
<td>Plan-oriented, adequately skilled, have access to external knowledge</td>
</tr>
<tr>
<td>Project Requirements</td>
<td>Emergent, rapidly changing</td>
<td>Knowable early, largely stable</td>
</tr>
<tr>
<td>Product Architecture</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Brief Comparisons

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Agile Methods</th>
<th>Plan-driven Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Objective</td>
<td>Rapid development</td>
<td>High quality</td>
</tr>
<tr>
<td>Project Size</td>
<td>Smaller teams and products</td>
<td>Larger teams and products</td>
</tr>
<tr>
<td>Developers</td>
<td>Agile, knowledgeable, talented, collocated, collaborative</td>
<td>Plan-oriented, adequately skilled, have access to external knowledge</td>
</tr>
<tr>
<td>Project Requirements</td>
<td>Emergent, rapidly changing</td>
<td>Knowable early, largely stable</td>
</tr>
<tr>
<td>Product Architecture</td>
<td>Designed for current requirements</td>
<td>Designed for current and foreseeable requirements</td>
</tr>
</tbody>
</table>

Summary

1. Waterfall Model

1. Requirements
2. Specification
3. Architectural Design
4. Detailed Design
5. Implementation
6. Testing
7. Maintenance

2: Concurrent Engineering

Requirements
Specification
Design
Implementation
Testing
time

3: Spiral Model

Requirements/Specification
Testing
Design
Coding

4: Agile Methods

Test Case Definition
Testing
Refactor
Design
Pair Programming
Announcements

• (Optional) draft design report due Thursday 4:30 p.m.

• No web review next week.