Software Development Methodologies

Lecturer: Raman Ramsin

Lecture 18

Process Patterns
Process Patterns

- Results of applying abstraction to recurring processes and process components

- Create means for developing methodologies through composition of appropriate pattern instances

- Reflect the state of the practice and are based on well-established, refined concepts
Process Patterns: Coplien

- The first recorded reference to the term “Process Pattern” was made by Coplien in his landmark paper in 1994.

- Coplien defined process patterns as “the patterns of activity within an organization (and hence within its project)”.

- Almost all his patterns are relatively fine-grained techniques for exercising better organizational and management practices.

- Do not constitute a comprehensive, coherent whole for defining a software development process.
Process Patterns: Ambler

- Ambler is the author of the only books so far written on object-oriented process patterns.

- Defines a process pattern as “a pattern which describes a proven, successful approach and/or series of actions for developing software”

- Defines an object-oriented process pattern as “a collection of general techniques, actions, and/or tasks (activities) for developing object-oriented software”.
Ambler's Process Patterns: Types

In the ascending order of abstraction level:

1. Task Process Pattern: depicting the detailed steps to execute a specific task of the process.

2. Stage Process Pattern: depicting the steps that need to be done in order to perform a stage of the process. A stage process pattern is usually made up of several task process patterns.

3. Phase Process Pattern: depicting the interaction of two or more stage process patterns in order to execute the phase to which they belong.

In any process, phases are performed in serial order, whereas the stage patterns inside them can be executed iteratively.

Ambler proposes many patterns of each type, complete with detailed steps and guidelines for integrating and shaping the patterns into a comprehensive process.
Ambler's Process Patterns: Task – Example

Technical Review

[Ambler 1998]
Ambler's Process Patterns: Stage – Example

Program

1. Understand Models
2. Write Source Code
3. Document Source Code
4. Reuse Existing Code and Components
5. Optimize Code
6. Synchronize Source Code With Models
7. Prepare Code for Inspections
8. Prepare Integration Plan
9. Integrate and Package
10. "Build" the Software
11. Packaged Application, Source Code

[Ambler 1998]
Ambler's Process Patterns: Phase – Example

Construct

[Ambler 1998]

Department of Computer Engineering

Sharif University of Technology
Object Oriented Software Process (OOSP)

Initiate
- Justify
- Define Initial Management Documents
- Define and Validate Initial Requirements
- Define Infrastructure

Construct
- Model
- Generalize
- Program
- Test in the Small

Deliver
- Test in the Large
- Rework
- Assess
- Release

Maintain and Support
- Support
- Identify Defects and Enhancements

Assure Quality, Manage the Project, Train and Educate, Manage People, Manage Risk, Manage Reuse, Manage Metrics, Manage Deliverables, Manage Infrastructure

[Ambler 1998]
Ambler's Process Patterns: Strengths and Weaknesses

**Strengths**

- Comprehensive and detailed specification document
- Full coverage of generic development lifecycle activities
- Iterative-incremental process
- Full support for umbrella activities
- Requirements-based development
- Based on functional, behavioural, and structural modeling of the problem domain and the system.
**Strengths (Contd.)**

- Accommodates comprehensive modeling at all levels (enterprise to problem domain to system objects; logical to physical).
- Rich modeling-language support (UML), especially in structural and behavioural modeling features.
- Support for formalism (through UML/OCL).
- Traceability supported through use cases.
Ambler's Process Patterns: Strengths and Weaknesses

**Weaknesses**

- Process patterns are not defined as individual patterns, but as components of a specific object-oriented methodology (OOSP);
  - this enhances the tangibility of the patterns but damages their generality and applicability.

- Very complex process (OOSP)
Ambler's Process Patterns: Strengths and Weaknesses

- **Weaknesses (Contd.)**

  - Configurability not addressed
  - Seamlessness damaged due to hitches in model mapping
  - Prohibitive number of models
  - Substantial potential for inconsistency of models
References

