

System & Architecture Modeling

V. Paúl Pauca

Department of Computer Science
Wake Forest University

CSC 331-631
Fall, 2013

Where are we so far?

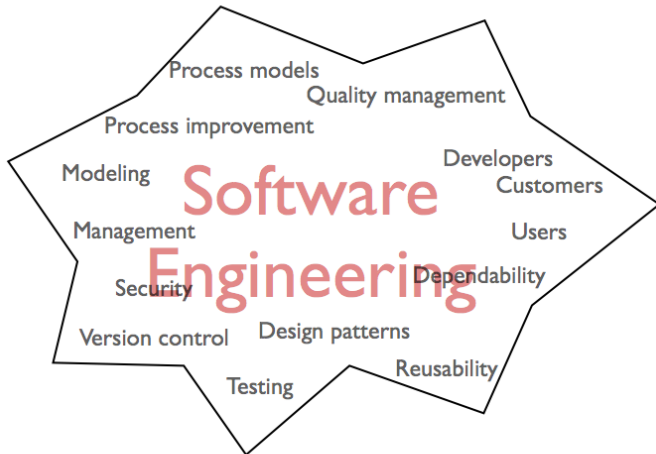
Theory

- **Software processes** (waterfall, incremental dev., etc.)
- **Process activities** (specification, design & implementation, validation, evolution)
- **Dealing with change** (iterative dev., prototyping)
- **Agile vs. plan-driven software dev.** (XP, agile proj. management)

Practice

- **OO fundamentals** (subclassing, polymorphism, etc.)
- **OO programming** (hands-on programming, Makefiles, header vs. implementation files, etc.)
- **OO design** (UML class and sequence diagrams, adapter pattern)
- **Agile development** (project management Trello, SCRUM, product backlog)

Understand big issues



Implementation & coding

Process of developing **abstract models** of a system, where each model presents a **different view or perspective** of the system.

Many ways of creating models:

- Formal (mathematical) models
- Data flow diagrams
- Structured analysis and design (SADT)
- Entity-relationship models (ERM)
- Transition diagrams
- Fence diagrams
- Unified Modeling Language (UML) diagrams
- Model-driven engineering (MDE)

Can be applied in all process activities:

- Software specification (requirements engineering)
- Design and implementation
- Software validation & testing, prototyping
- Software evolution

Different perspectives of the system:

- **External perspective**: context or environment of the system
- **Interaction perspective**: interactions between a system and its environment (or between components of the system)
- **Structural perspective**: organization of the system and its data
- **Behavioral perspective**: dynamic behavior of the system

Modeling through UML

- **UML class diagrams**: structural perspective of the system
- **UML sequence diagrams**: behavioral perspective of the system
- **UML use case diagrams**: interaction perspective
- **UML activity diagrams**: sequential or parallel activities involved in a process
- **UML state diagrams**: how the system state changes with internal/external events

Why do Modeling?

- **Facilitate discussion** about a proposed system
Detail and rigor (abstraction) depends on intended use
- **Document** a proposed system
- **Describe the system** for subsequent implementation

Disadvantages

- Requires expertise in use of UML and other modeling tools
- Important detail may be left out