

COMP 5104 Quality in OO Software Development

1

Course Outline

2

Learning Objectives

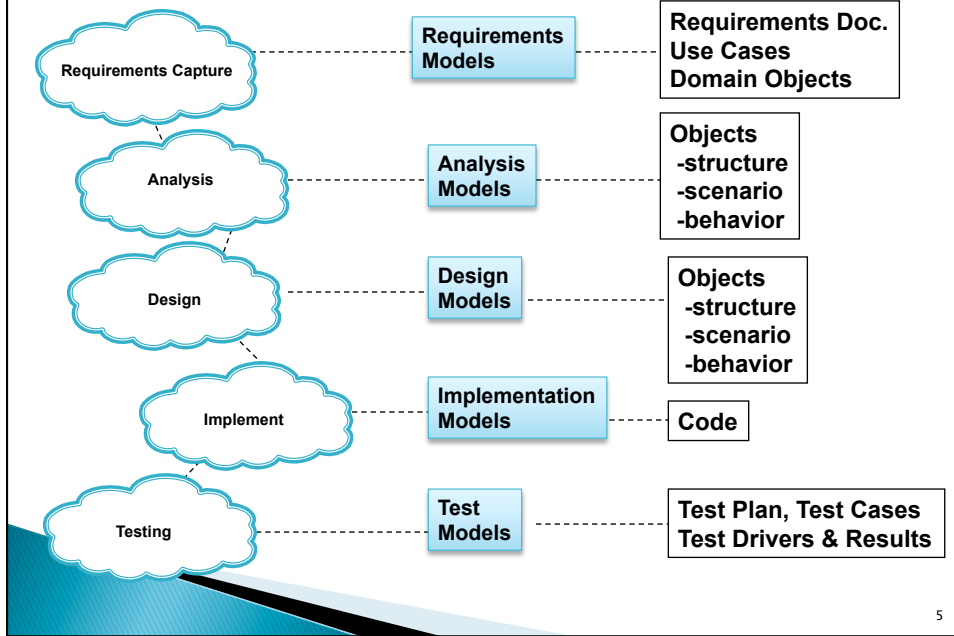
- ▶ Understand the pros and cons of TDD and of refactoring in Java
- ▶ Have a basic understanding of metrics
- ▶ Understand the basics of UML 2.0 and of UCMs
- ▶ Have a basic understanding of a selection of G04 design patterns (with their variants)
- ▶ Understand the pros and cons of state-based MBT (and of SpecExplorer)
- ▶ Understand the basics of modeling using ACL
- ▶ Time permitting, have a basic understanding of generative modeling

3

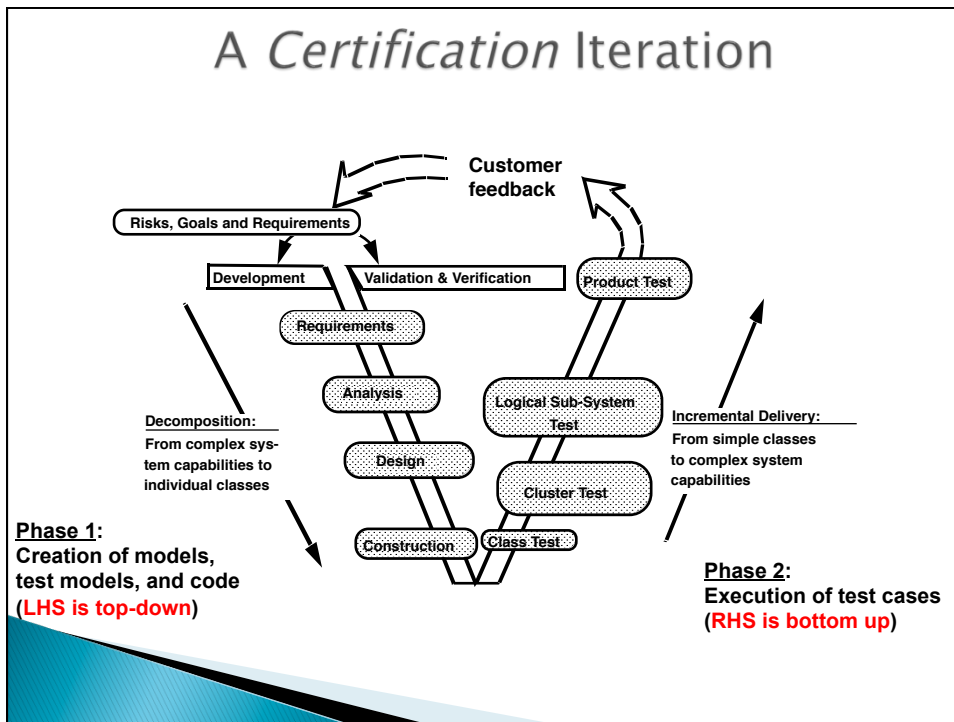
Activities, Process, and Models

4

OO Development Overview



A Certification Iteration



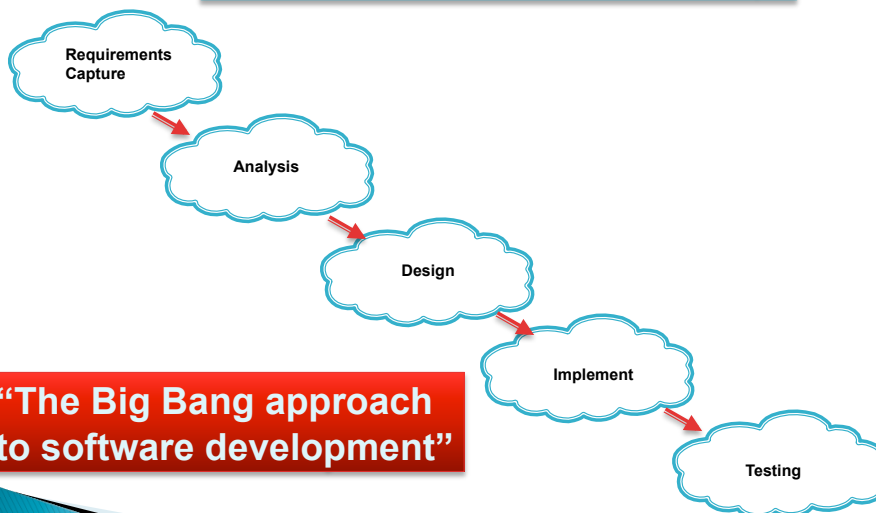
Development Process

- ▶ A development process defines the set of models that are to be used by designers and the order in which these models are used
 - Each model must have a well-defined role in the process
 - When defining the process, the models must be chosen based on the **domain concepts** that need to be modeled:
 - Is OO relevant to the problem at hand?
 - e.g., OO is particularly well suited for event-driven systems
 - Is software evolution a concern or not?
 - Is software quality a concern or not?

7

Waterfall Development

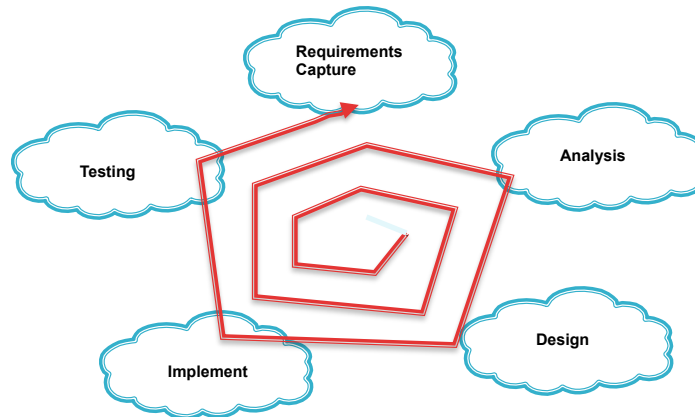
Activities carried out one after the other as steps



8

Incremental-Iterative Development

The Spiral Model



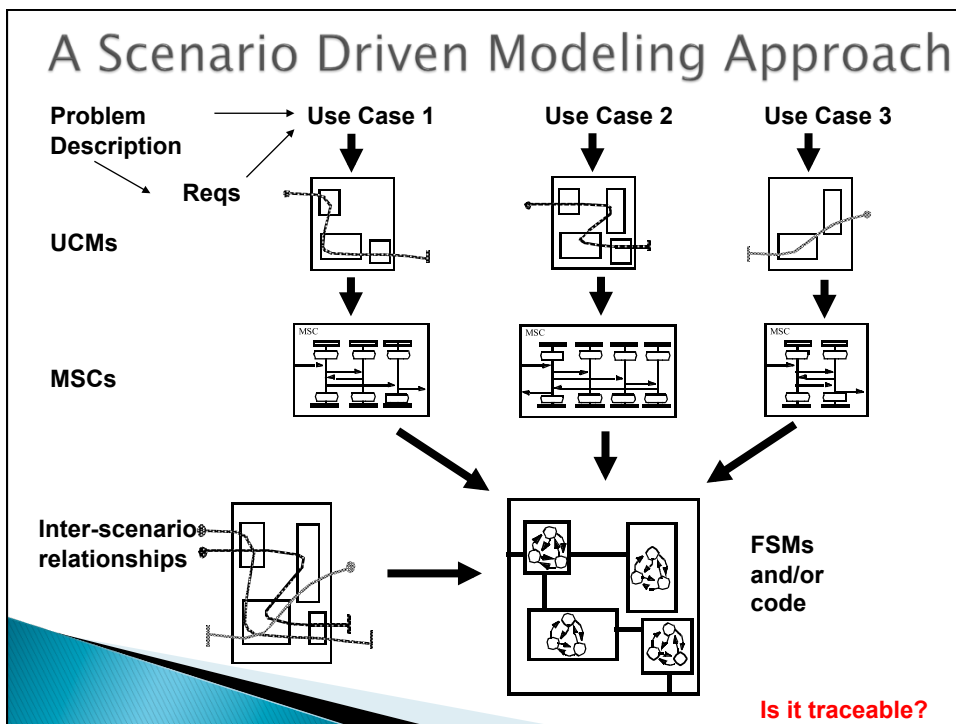
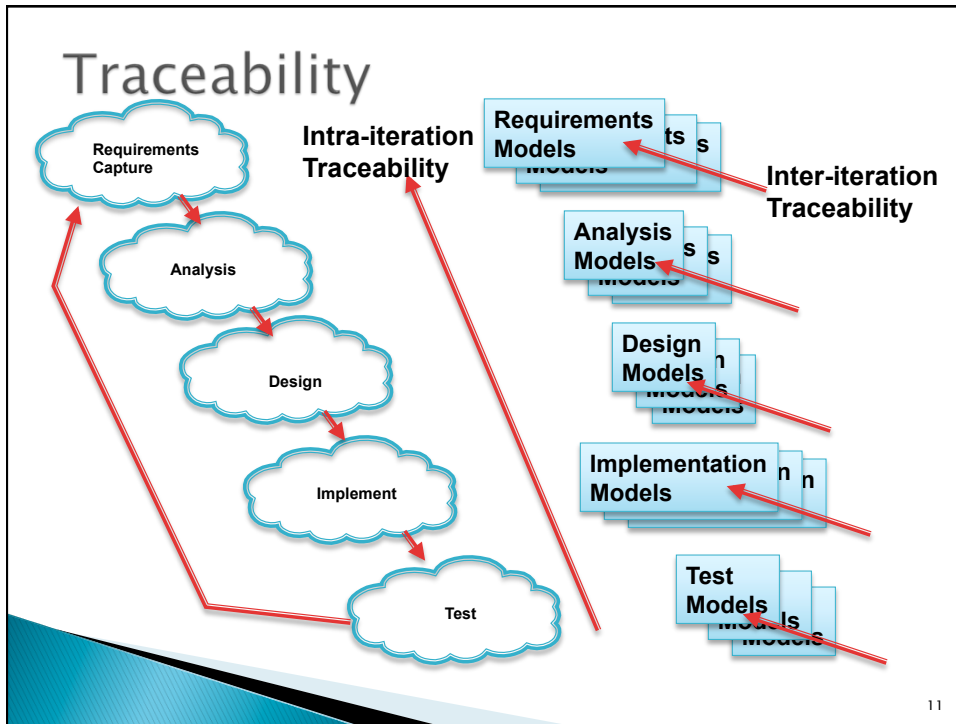
Can have macro and micro iterations
but needs to converge

9

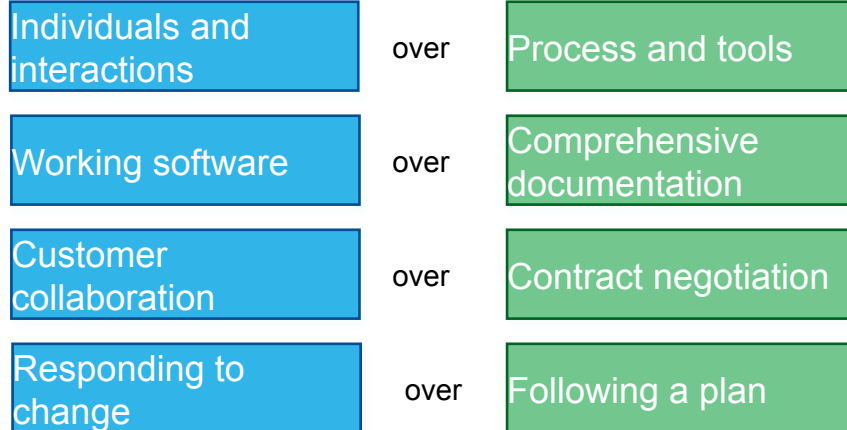
Aiming for Traceability

- ▶ Traceability is required to achieve convergence:
 - We must document the *continuity* that must exist between the work-products of different activities.
 - In turn, continuity enables completeness and consistency checks.
- ▶ Within a particular activity, the work-products must be *consistent*:
 - e.g., the structural, scenario and behavior models must be consistent
 - For example, if an interaction diagram shows an object receiving a message, then the FSM of this object must reflect this possibility
- ▶ Work-products must also be *complete* with respect to the current requirements.

10



The Agile Manifesto—a statement of values



Source: www.agilemanifesto.org

The Core of Agile Modeling

Some Core Principles

- Assume Simplicity
- Expect Incremental Change
- Enabling the Next Effort is Your Secondary Goal
- Model With a Purpose
- Use Multiple Models
- Maximize Stakeholder Investment
- **Value Quality first**
- Get Rapid Feedback
- **Software Is Your Primary Goal**

Some Core Practices

- Active Stakeholder Participation
- Apply the Right Artifact(s)
- **Collective Ownership**
- Create Several Models in Parallel
- Create Simple Content
- Depict Models Simply
- Display Models Publicly
- **Model in Small Increments**
- Model With Others
- **Prove it With Code**
- Use the Simplest Tools