

# CPSC 310 – Software Engineering

## Course Review



## Software Process

Agile

SCRUM

Waterfall /  
V Cycle

Spiral

Testing

Requirements

Design

Modular

Pattern

Refactoring

User Interface

Quality

Collaborative Development

# CPSC310 TOPICS

# Software Process

Agile

SCRUM

Waterfall /  
V Cycle

Spiral

Testing

Design

Modular

Pattern

Refactoring

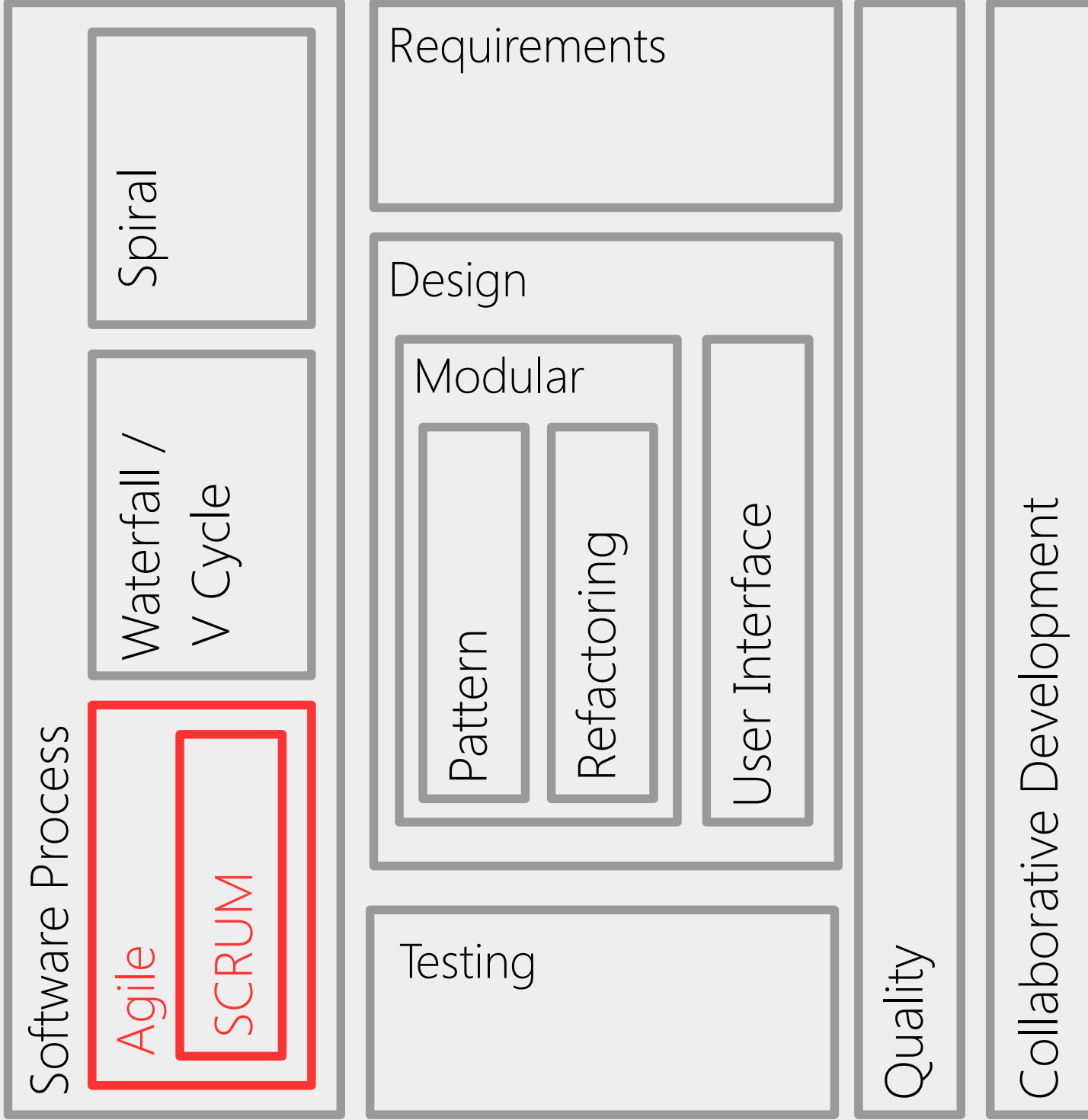
User Interface

Requirements

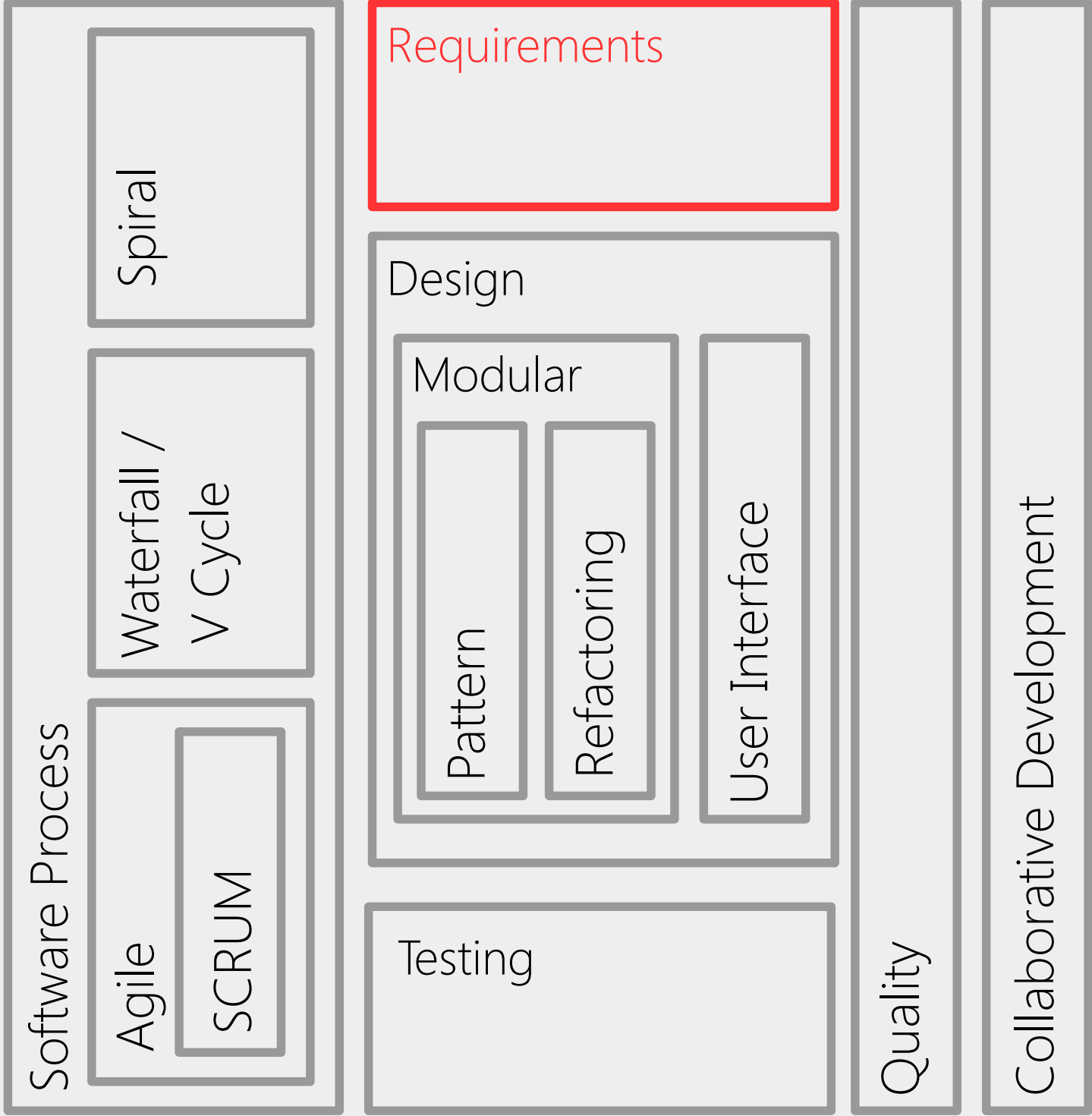
Quality

Collaborative Development

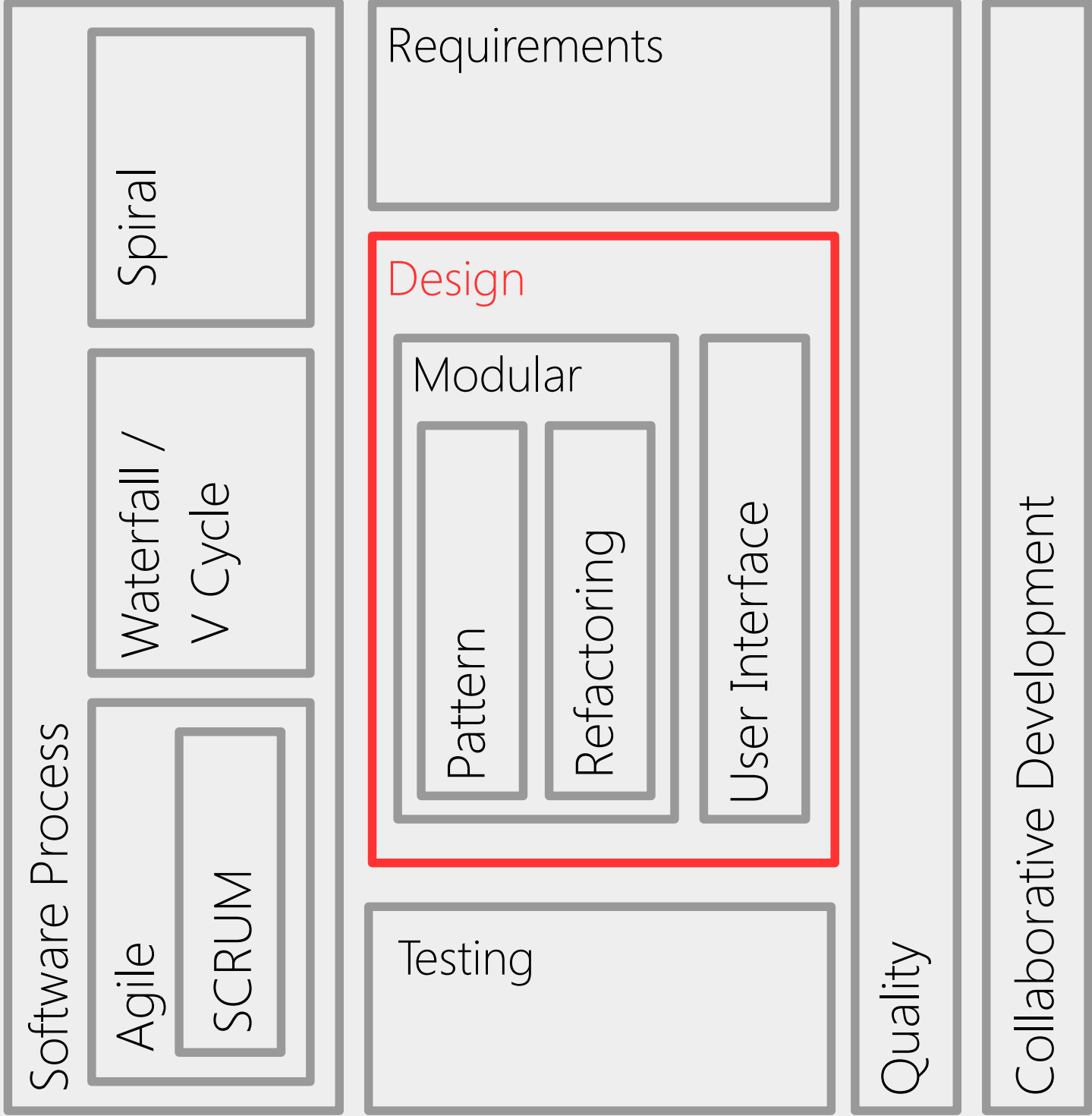
- describe benefits of using a software process
- describe waterfall and spiral model including drawbacks
- describe the importance of agile methods



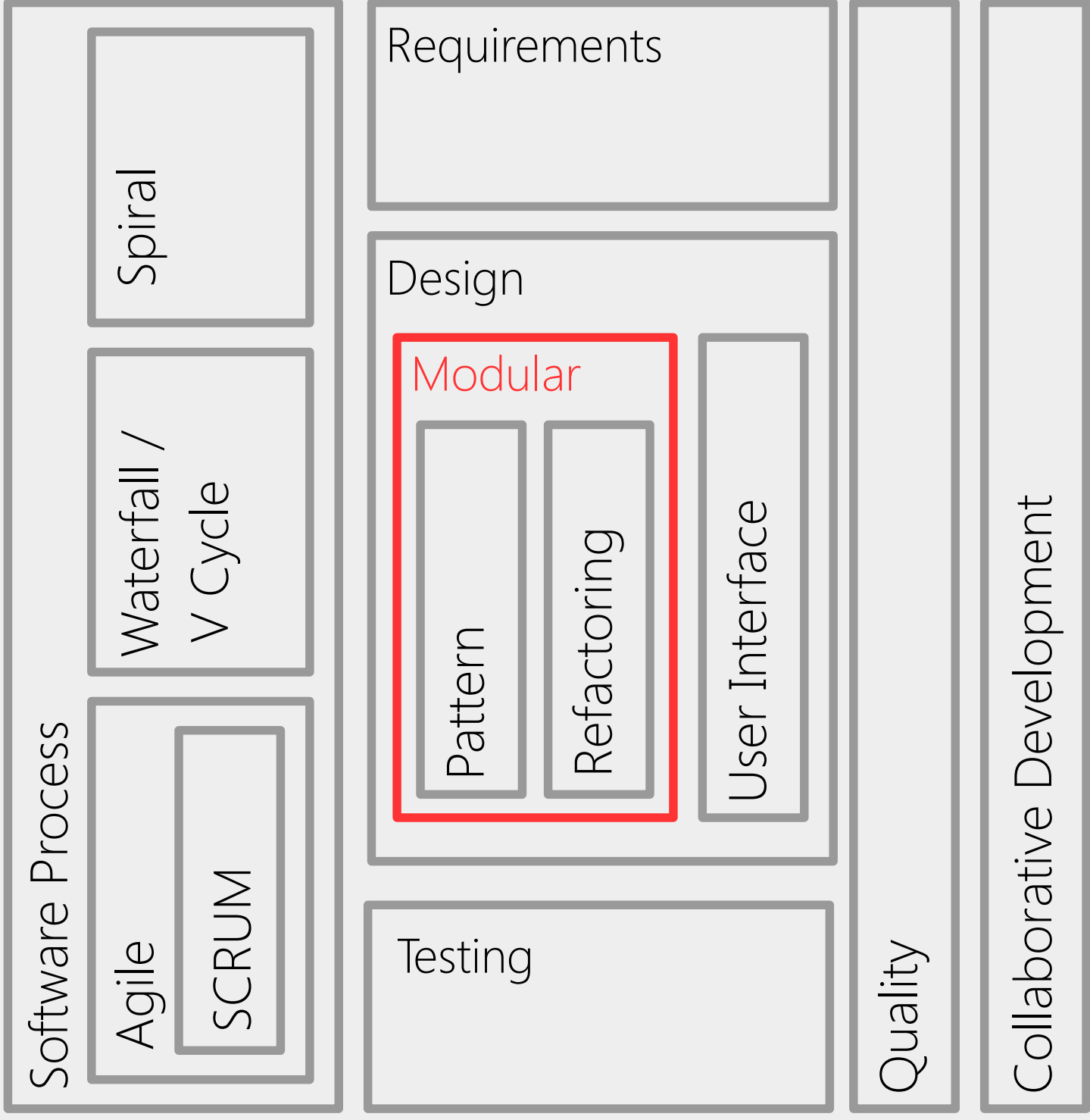
- describe the general principles of agile
- understand that agile encompass different approaches and that it's not a silver bullet
- describe SCRUM
  - Roles
  - Ceremonies
  - Artifacts



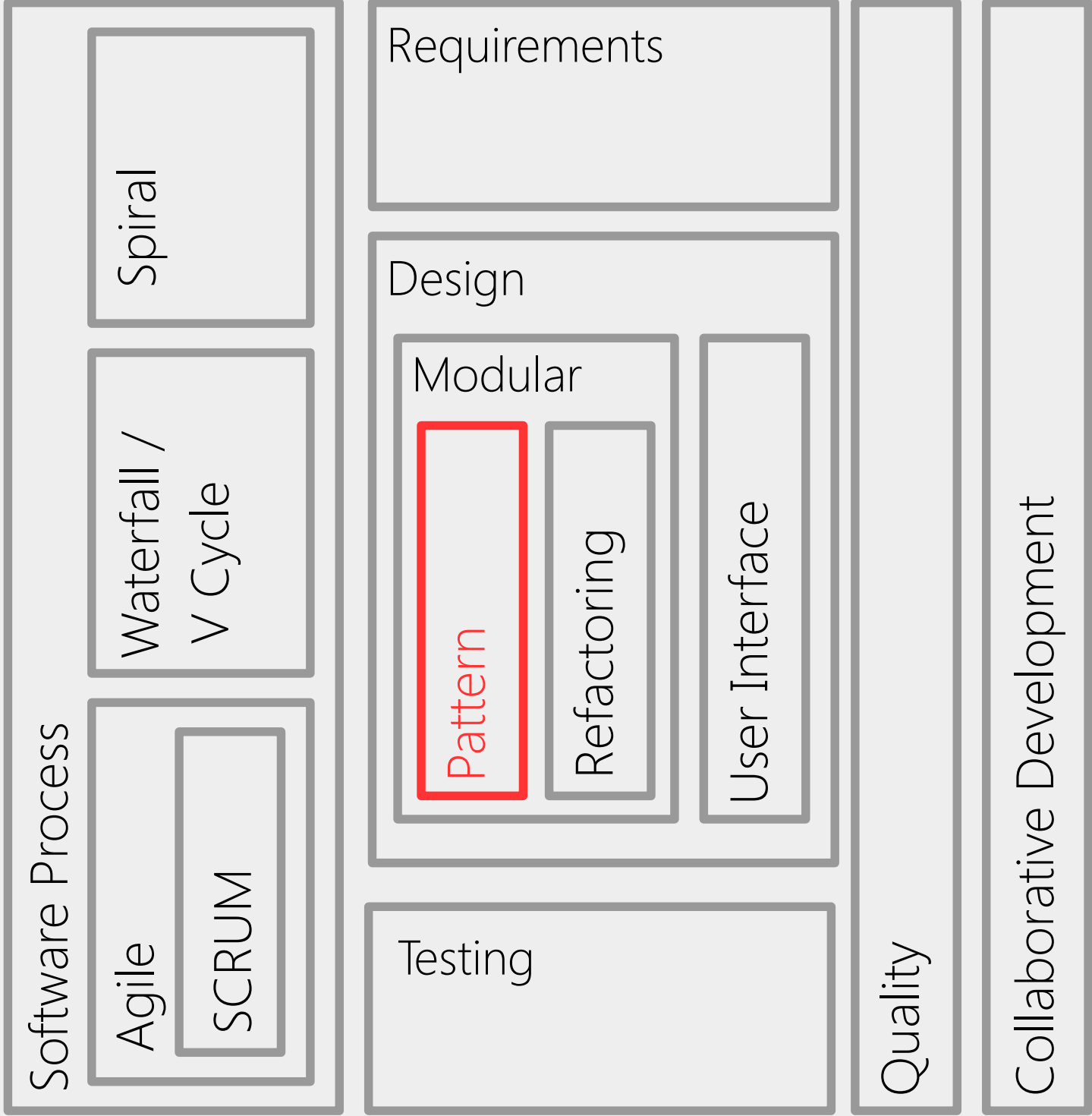
- Why requirements are needed
- How to elicit requirements
- User stories, and how to write good ones (INVEST guidelines)



- Describe benefits of design
- How to approach design
- UML Class Diagram and Sequence Diagram (210 review)

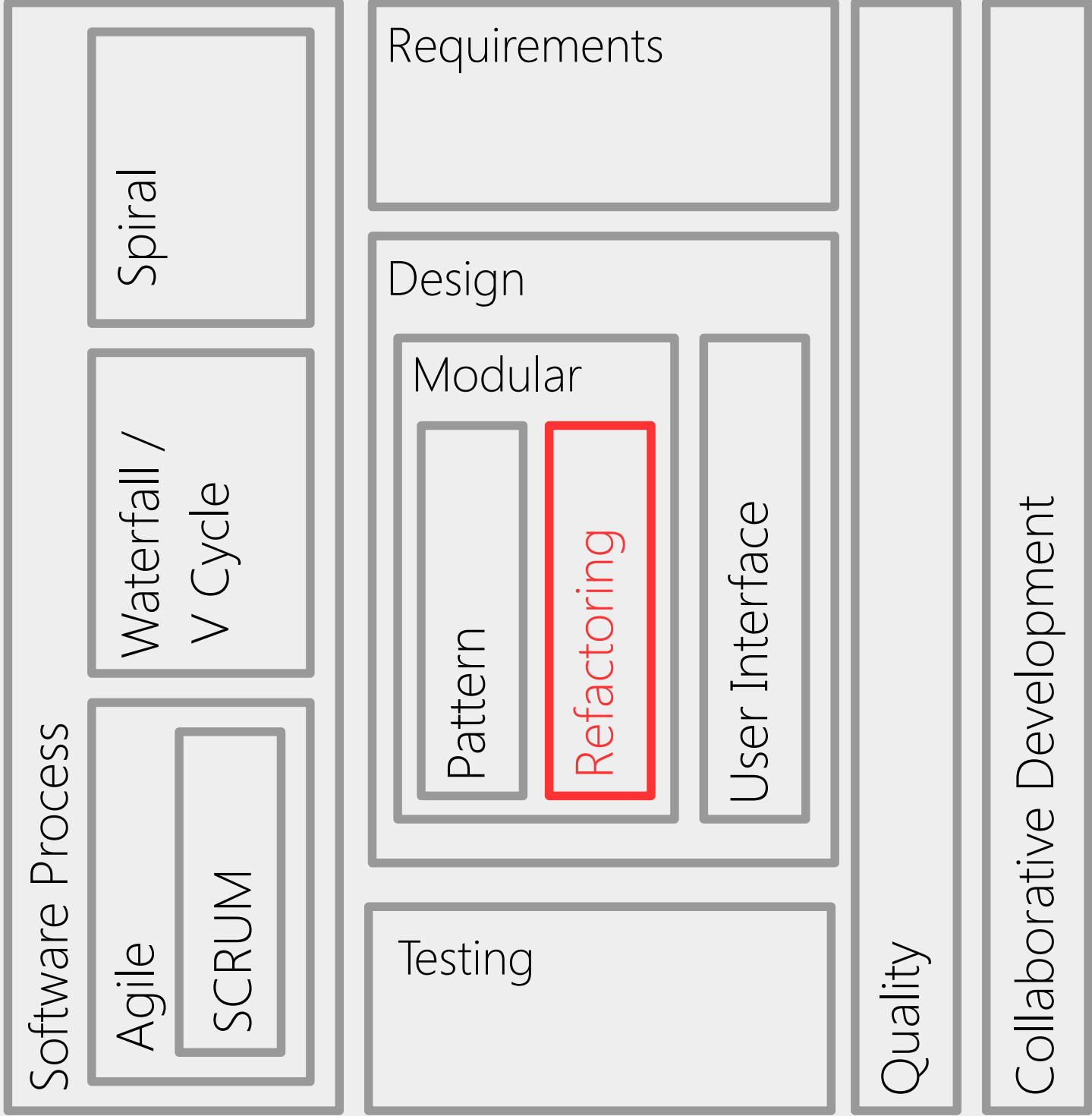


- Understand/apply
  - Strong cohesion
  - Low coupling
- Information hiding
- Liskov Substitution Principle
- Open/Closed Principle
- Law of Demeter

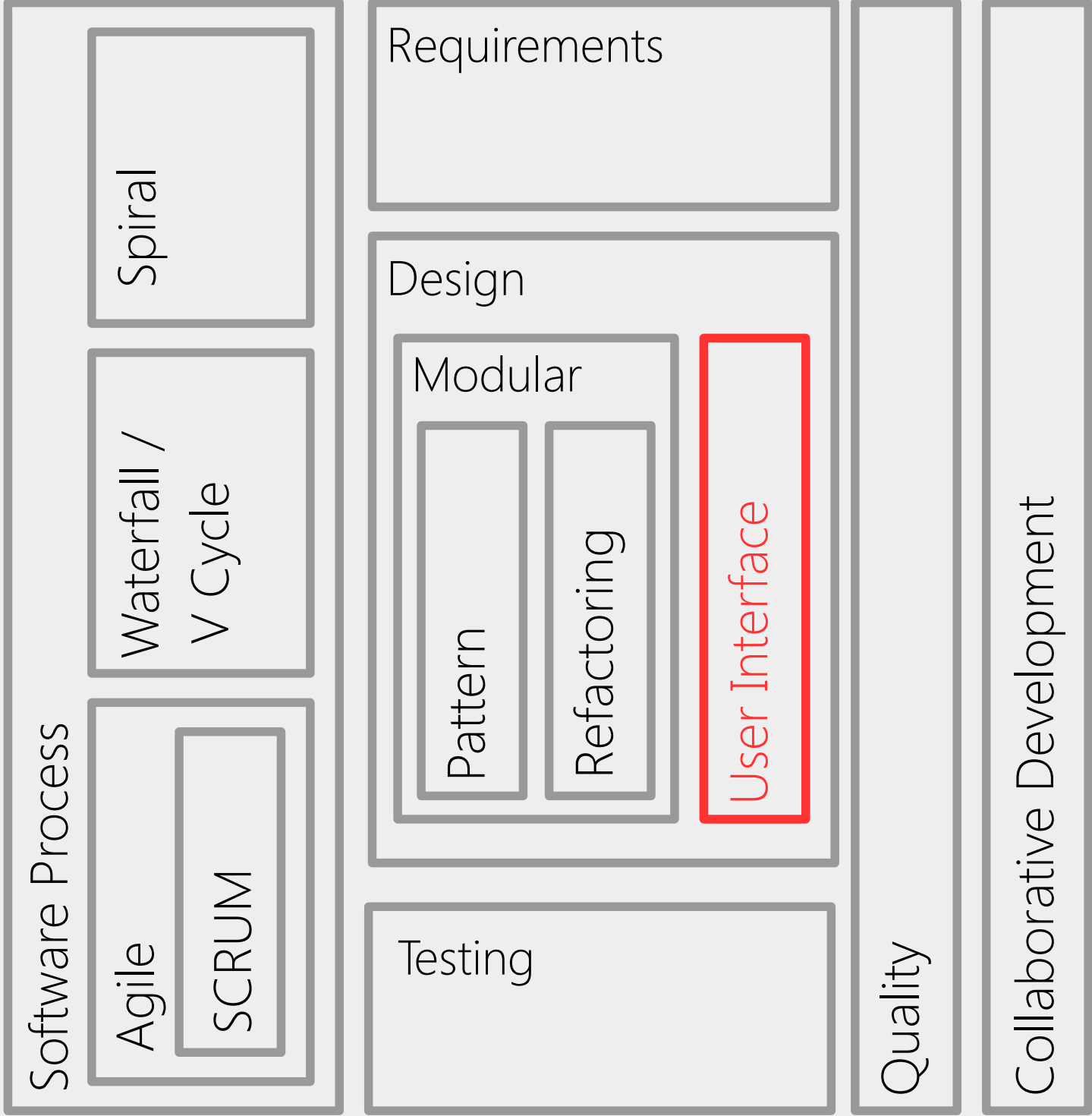


- Describe benefits and drawbacks of design patterns
- Use the following patterns:
  - Factory
  - Singleton
  - Decorator
  - Proxy
  - Template
  - Composite
  - Adapter
  - Observer
- Know what anti-pattern are

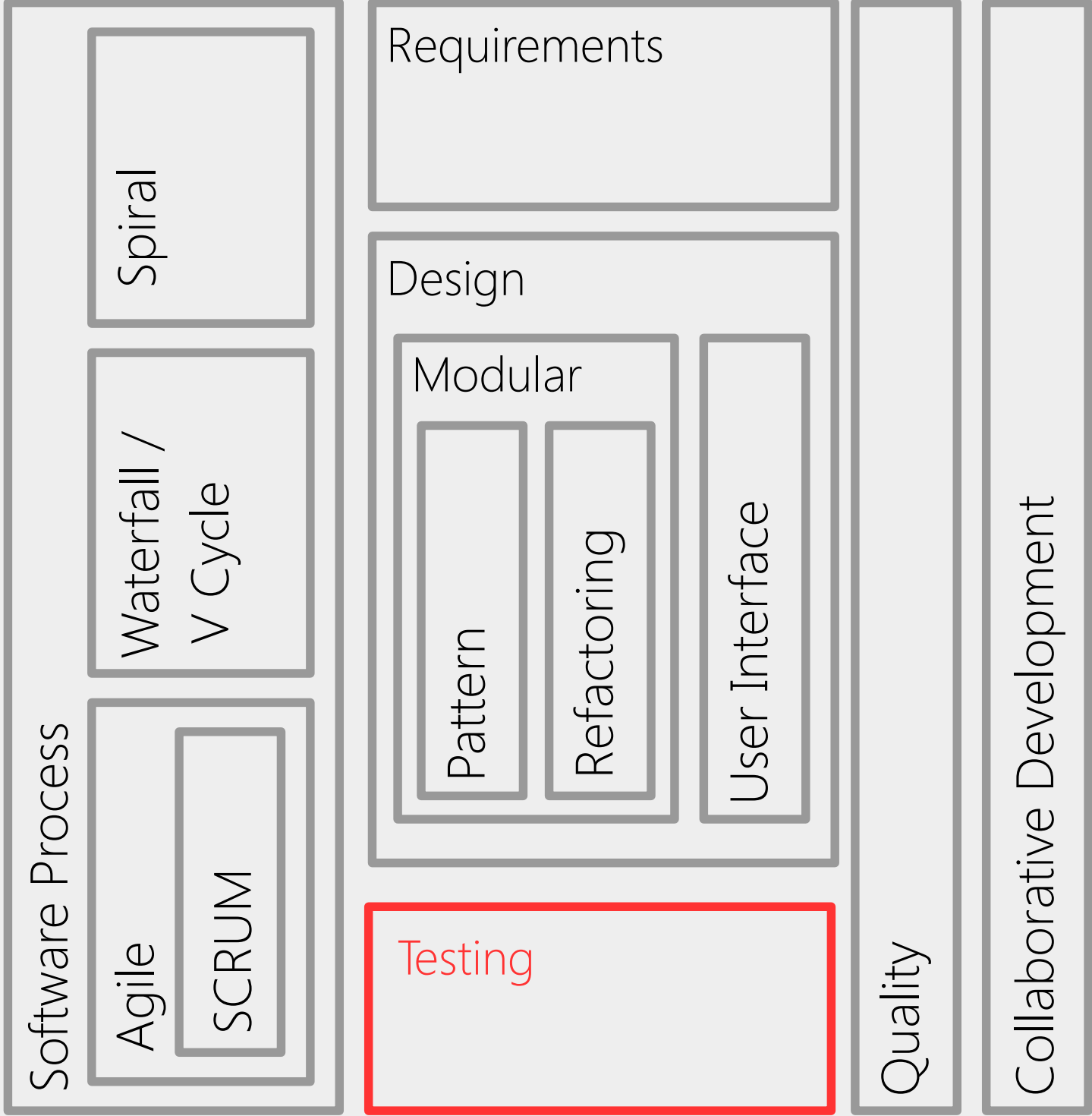




- Why refactor
- When refactor
- How to refactor
  - Identify code smells
  - Apply transformation rules
  - Ensure that behavior is constant



- be able to assess the usability of an interface
- Nielsen Principles of Design
- Components for usage
- Usability Testing



- Types of Testing (Unit, Regression, Integration, Acceptance)
- Black Box Testing
  - Equivalence classes
  - Boundary Tests
- White Box Testing
  - Coverage

## Software Process

Agile

SCRUM

Waterfall /  
V Cycle

Spiral

Testing

Design

Modular

Pattern

Refactoring

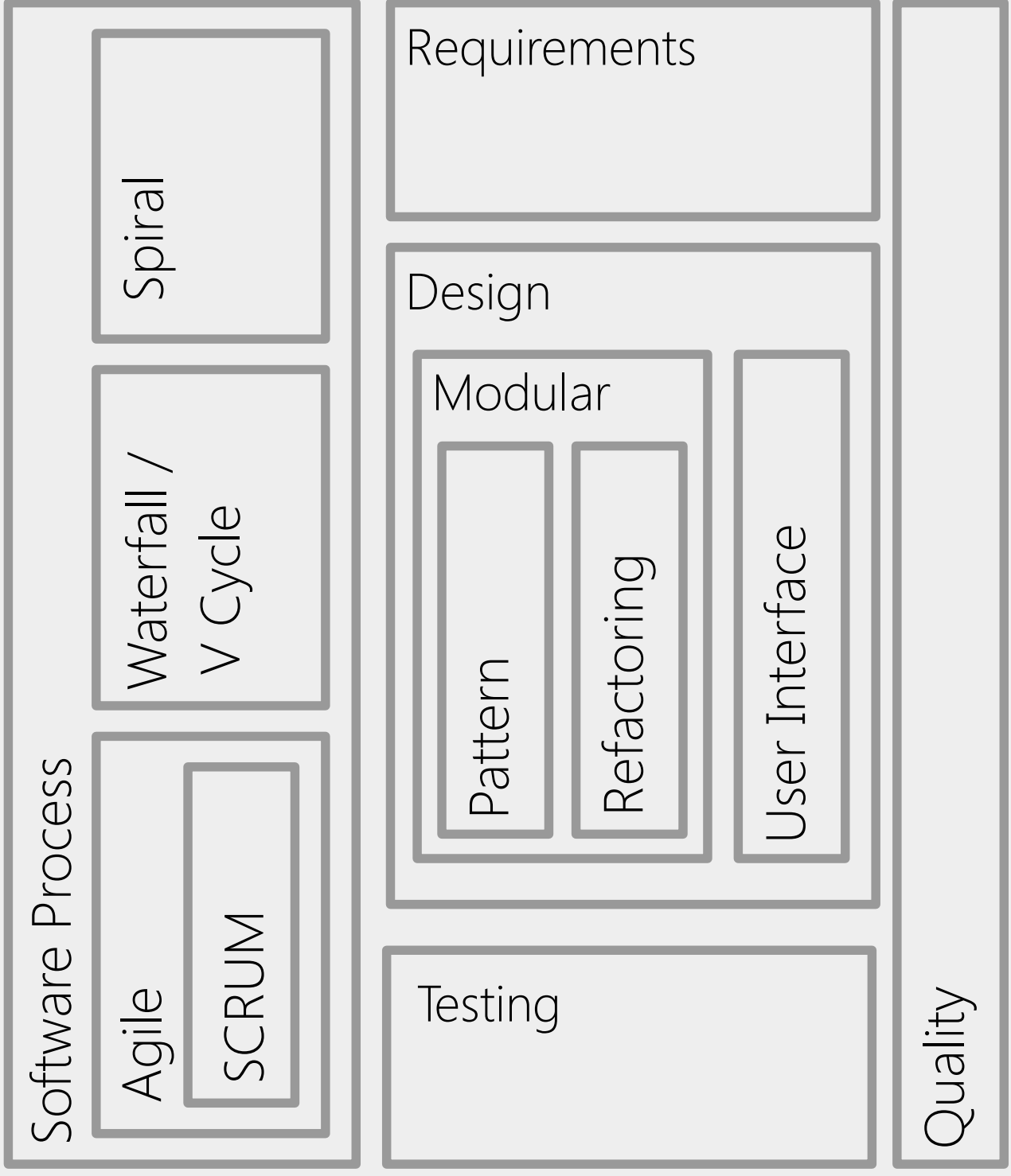
User Interface

Requirements

Quality (mainly code)

Collaborative Development

- Know that quality is affected by the whole process
- describe techniques for checking/improving code quality



Collaborative Development

- Know why VCS are great (yes they are)
- Describe the difference between centralized and distributed VCS

# How to study

- Re-read the slides and follow the links for clarification and more context.
- Master in-class exercise
- Ask questions on Piazza after looking if the answer is not already there
- I will be holding office hours next week (see course calendar)

