Chapter 2: Software Processes - Exam-Oriented Summary

Main Topics

- Software Process Models
- Process Activities
- Coping with Change
- Rational Unified Process (RUP)

Core Process Activities

1. Specification: Define system services and constraints.
2. Design & Implementation: Turn specs into executable software.
3. Validation: Ensure software meets requirements.
4. Evolution: Update system to reflect new needs.

Software Process Models

- Waterfall: Plan-driven, sequential, inflexible to change.
- Incremental: Develop in parts, allows feedback, faster delivery.
- Reuse-oriented: Builds from existing components (COTS, libraries).

Waterfall Model

- Phases: Requirements > Design > Implementation > Testing > Maintenance
- Drawbacks: Difficult to accommodate change, poor for evolving needs.

Incremental Development

- Benefits: Early delivery, easier feedback, lower change cost.
- Problems: Less visibility, architecture degradation without refactoring.

Reuse-Oriented Development

- Based on using existing components: COTS, web services, libraries.
- Phases: Component analysis > Req modification > Design > Integration.

Chapter 2: Software Processes - Exam-Oriented Summary

Requirements Engineering Process

- Feasibility Study
- Elicitation & Analysis
- Specification
- Validation

Design Activities

- Architectural Design
- Interface Design
- Component Design
- Database Design

Validation & Testing

- Development Testing: Individual components
- System Testing: Whole system
- Acceptance Testing: Customer validation with real data

Coping with Change

- Change Avoidance: Use prototyping to anticipate needs
- Change Tolerance: Incremental development to adapt easily

Prototyping

- Helps in requirements validation and UI design
- Improves usability, reduces cost
- Often discarded and not used in production

Incremental Delivery

- System delivered in prioritized parts

- Early functionality, customer feedback, lower risk

Spiral Model (Boehm)

- Phases: Objective setting, Risk reduction, Development, Planning
- Risk-driven, iterative, rarely used as-is

Rational Unified Process (RUP)

- Phases: Inception, Elaboration, Construction, Transition
- Iterative within and across phases
- Good Practices:
  - Iterative development
  - Manage requirements
  - Component-based architectures
  - UML modeling
  - Quality assurance
  - Change control

Key Exam Points

- Know the 3 main models: Waterfall, Incremental, Reuse-oriented
- Understand Spiral and RUP as modern flexible models
- Be able to compare plan-driven vs agile processes
- Understand prototyping and when to use it
- Know the phases of testing and requirements engineering