

Section 7.2

Software Development Processes

1. Overview
2. Classic development processes
3. Agile development

7.2.1 Overview

- What is a software life cycle process?
 - approach to conducting system development activities
 - goals and activities stay the same
 - order and focus of activities may differ
- Two broad categories of life cycle processes
 - activity centered
 - focus is on development activities
 - entity centered
 - focus is on work products

Overview (cont.)

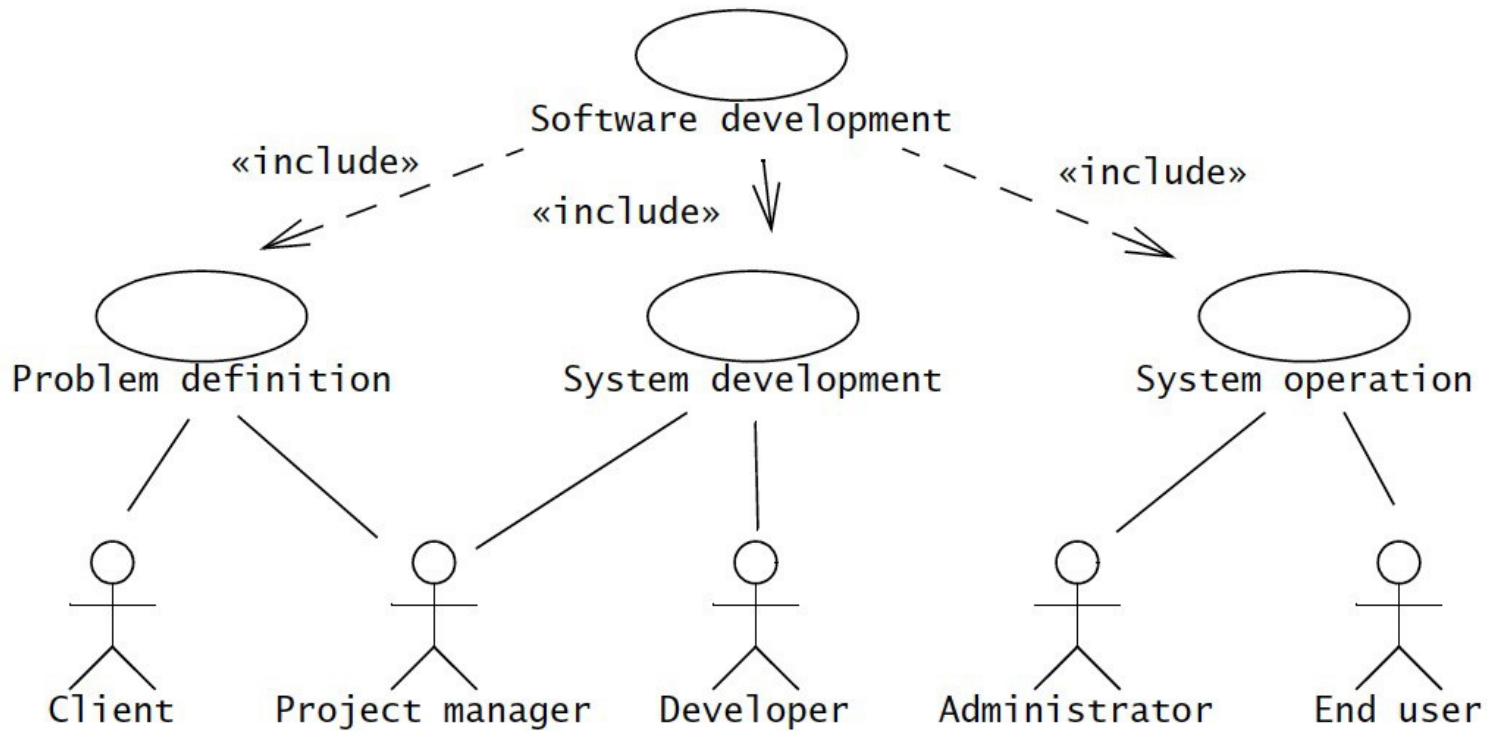


Figure 15-1 Simple life cycle for software development (UML use case diagram).

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

Overview (cont.)

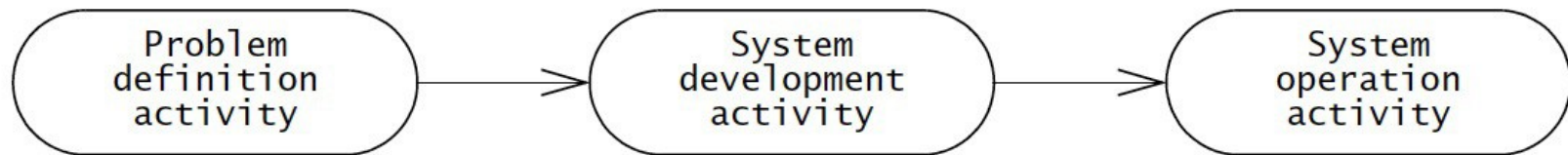


Figure 15-2 Simple life cycle for software development (UML activity diagram).

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

Overview (cont.)

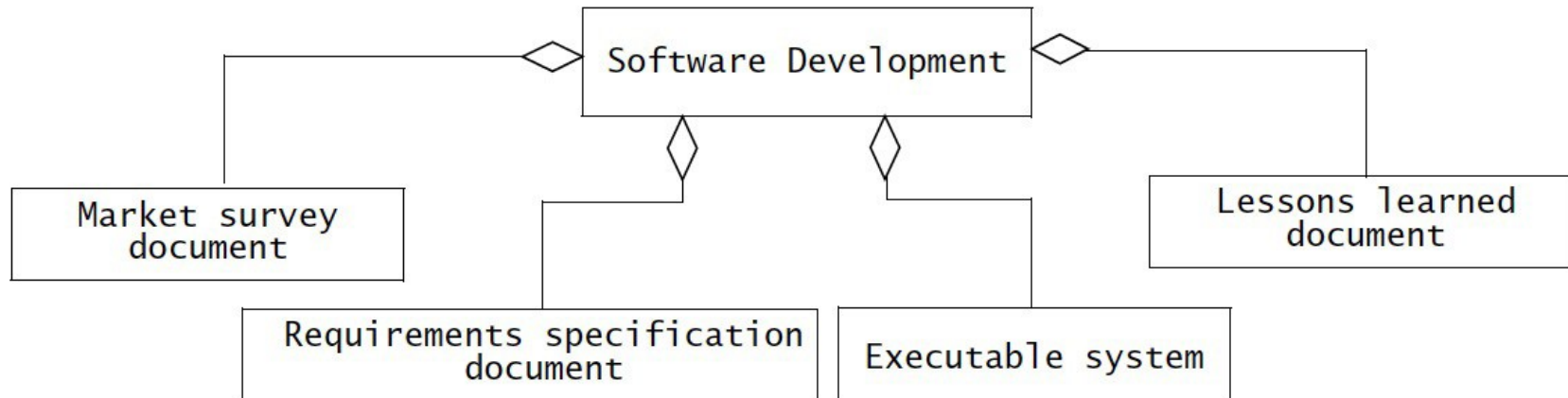




Figure 15-4 Entity-centered view of software development (UML class diagram).

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

7.2.2 Classic Development Processes

- Sequential activity centered models
 - waterfall model
 - V-model
- Iterative activity centered models
 - spiral model
 - Unified Software Development Process (USDP) 
- Entity centered models
 - issue-based development process 

Waterfall Model

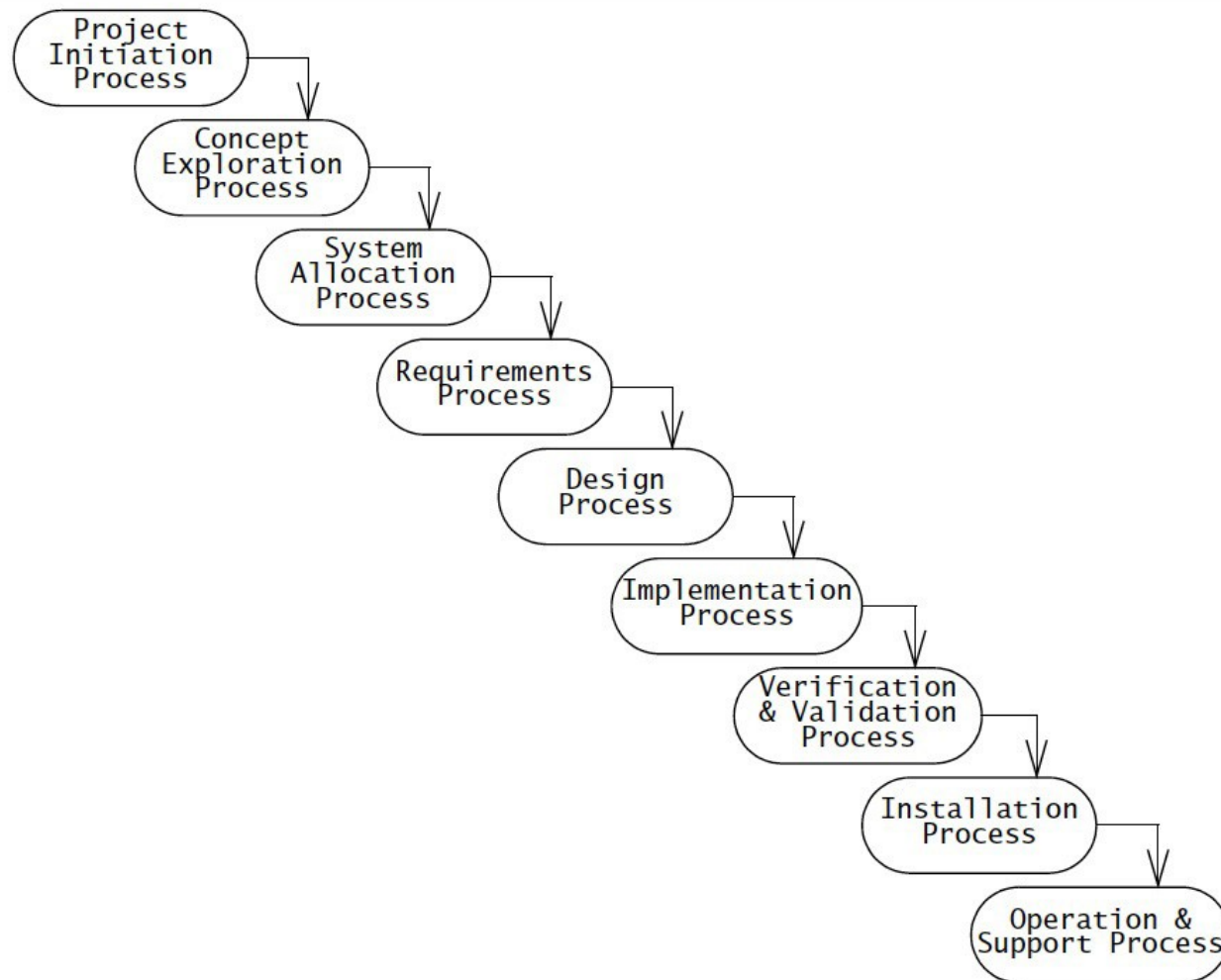


Figure 15-8 The waterfall model of software development is an activity-centered view of the software life cycle. Software development activities are performed in sequence (UML activity diagram adapted from [Royce, 1970] using IEEE 1074 names; project management and cross-development processes are omitted).

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

Waterfall Model (cont.)

- Characteristics of the waterfall model
 - activity centered
 - activities are executed sequentially
 - e.g. requirements activities completed before design starts
 - simple approach
- Goal
 - never revisit an activity once it's completed
- Strategy
 - constant verification at each step

V-Model

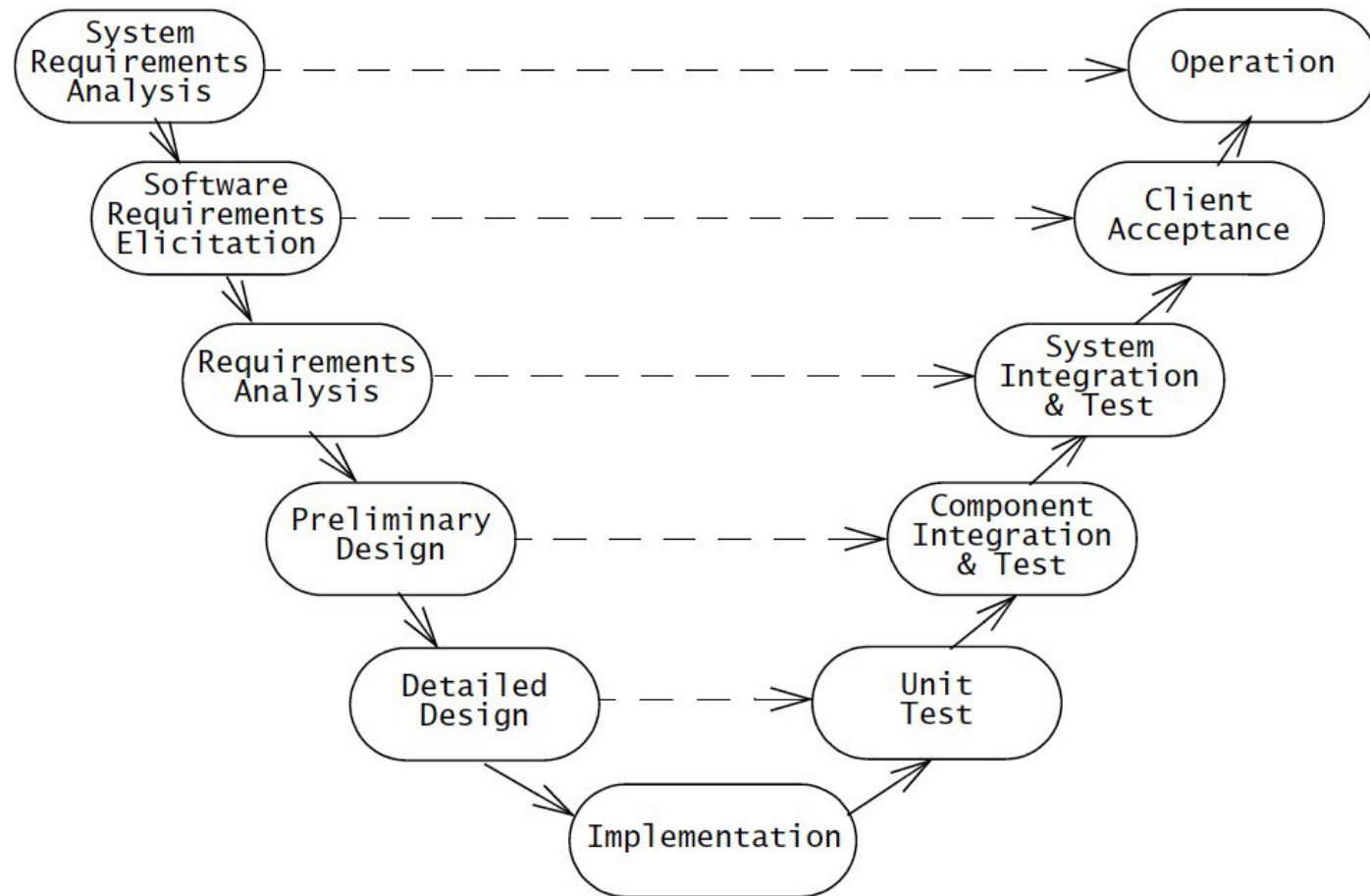


Figure 15-9 V-model of software development (UML activity diagram; adapted from [Jensen & Tonies, 1979]). The horizontal object flow denotes the information flow between activities at the same abstraction level. The V-shape layout of the activities was conserved to reflect the original drawing. However, the layout of the activities has no semantics in UML.

V-Model (cont.)

- Characteristics
 - activity centered
 - activities are executed sequentially
 - variation of the waterfall model
 - dependencies between development and verification are explicit
- Goal
 - depict the level of abstraction
 - higher levels deal with requirements
 - middle levels focus on architecture
 - lower levels concerned with details of software components

Spiral Model

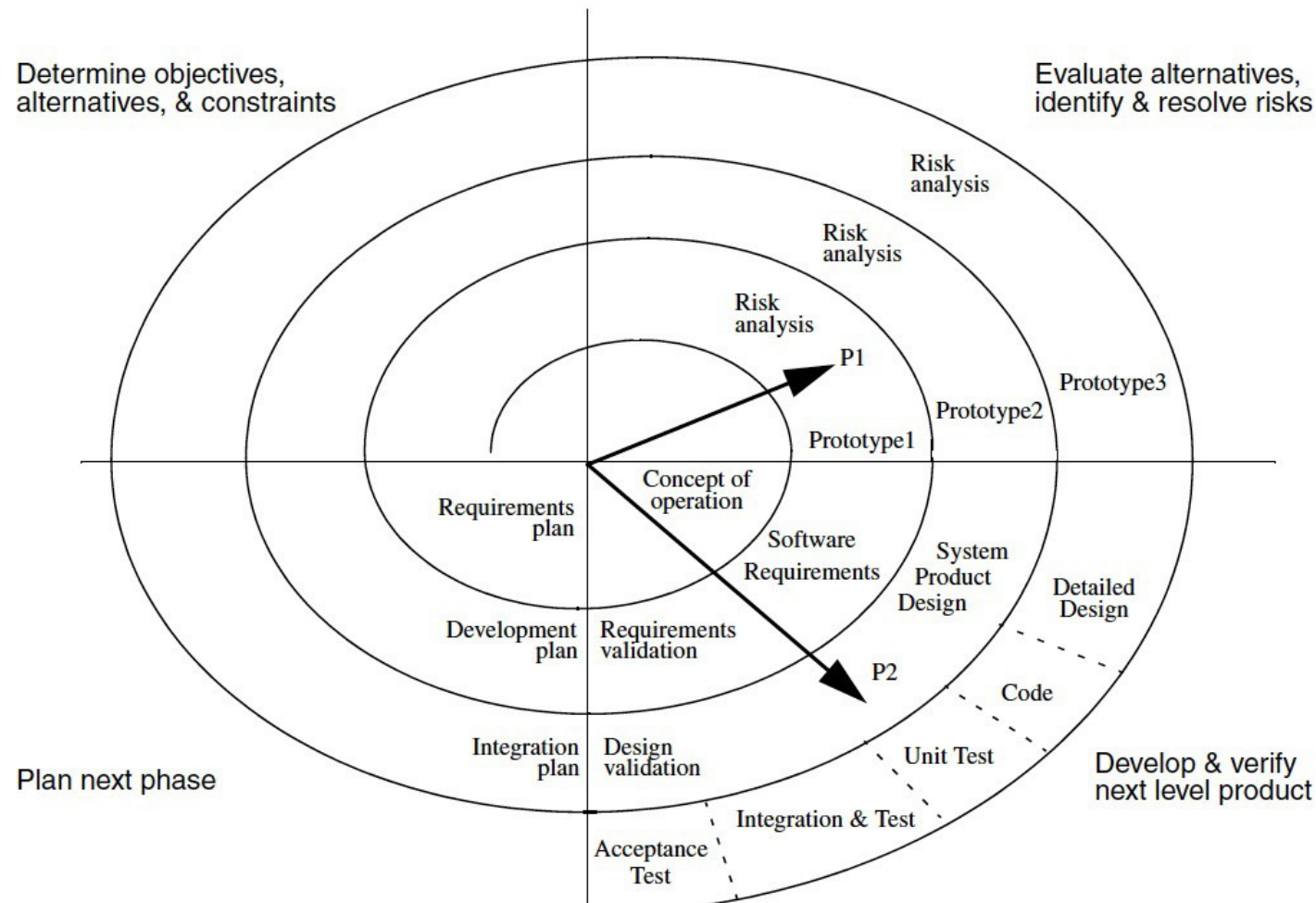


Figure 15-10 Boehm's spiral model (adapted from [Boehm, 1987]). The distance from the origin represents the cost accumulated by the project. The angle from the horizontal represents the type of activity. For example, the project P1 is currently in the risk analysis activity associated with software requirements. The project P2 is in the development of the system product design.

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

Spiral Model (cont.)

- Characteristics

- activity centered
- activities are executed iteratively
- focus on addressing risks incrementally, in order of priority
- adds risk management and prototyping to each iterative cycle
- each round has four phases:
 - determine objectives, constraints, alternatives
 - identify and resolve risks
 - develop and verify prototype
 - plan next round

- Goal

- address frequent changes during development

Unified Software Development Process

- Characteristics
 - activity centered
 - activities are executed iteratively
 - cycles characterize a stage of system maturity over its lifetime
 - birth, childhood, adulthood, retirement, death
 - each cycle ends with a release to the client

USDP (cont.)

- Characteristics (cont.)
 - each cycle can be in one of four phases:
 - inception, elaboration, construction, transition
 - each phase has multiple iterations
 - each iteration:
 - addresses a set of related use cases or mitigates some risks
 - managed as a project
 - has activities (workflows) performed in parallel

USDP (cont.)

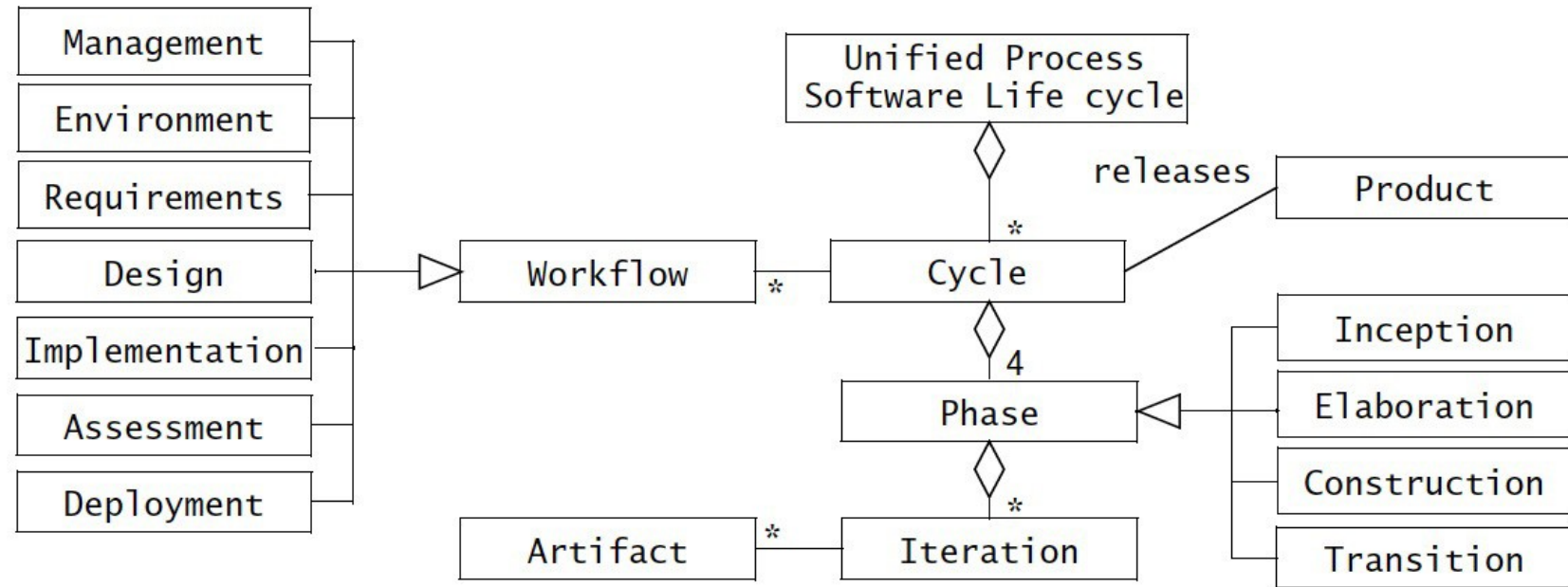


Figure 15-12 Workflows in the unified software life cycle used by Royce [Royce, 1998] (UML class diagram).

Copyright © 2011 Pearson Education, Inc. publishing as Prentice Hall

USDP (cont.)

- Inception phase
 - establish scope of system
 - identify use cases related to trade-off decisions (core use cases)
 - identify at least one software architecture
 - demonstrate core use cases against architecture
 - estimate and schedule resources
- Elaboration phase
 - planning
 - capture user's view of requirements
 - design software architecture
 - make build vs. buy decisions
 - derive cost, schedule, resource estimates

USDP (cont.)

- Construction phase
 - developing
 - buy or build components
 - compare release with acceptance criteria
- Transition phase
 - system is ready for deployment

USDP (cont.)

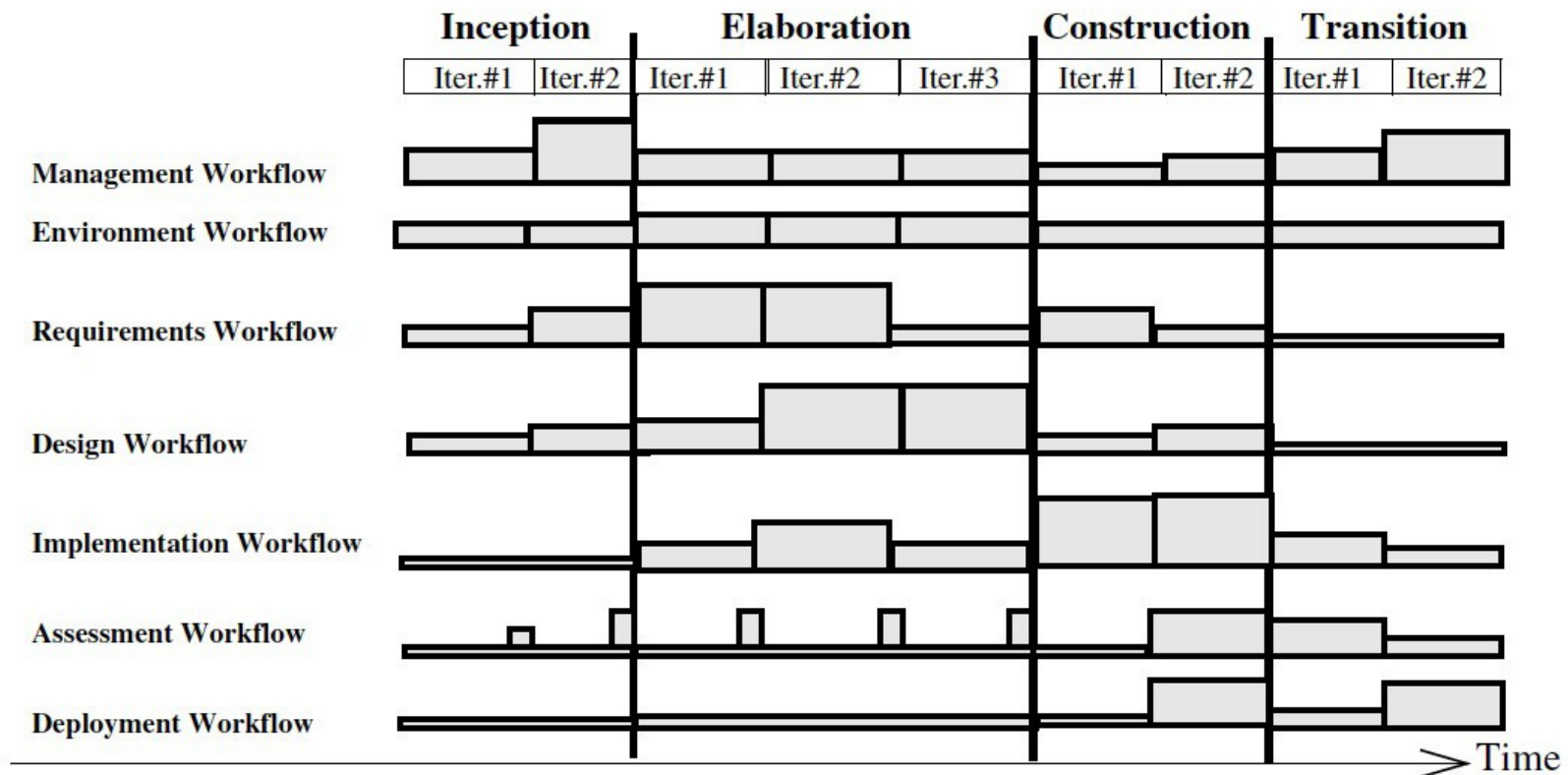


Figure 15-13 The seven workflows in the Unified Process—*Management*, *Environment*, *Requirements*, *Design*, *Implementation*, *Assessment*, and *Deployment*—are project functions performed during the entire lifetime of a system. Each of the workflows has different resource needs depending on the phase and iteration of the software system. The histograms for each workflow indicate the amount of work per iteration—the higher the block, the more resources are consumed by the workflow in that iteration (adapted from [Jacobson et al., 1999] and [Royce, 1998]).

7.2.3 Agile development

- Characteristics
 - emphasis on implementation phase
 - some requirement analysis
 - minimal design
 - testing concurrent with implementation
- Goal
 - shorter delivery times
 - faster response to change