

# Unified Software Development Process (UDSP or USP)

- Complement to UML
- Rational Unified Process (RUP)
  - IBM owns Rational now
- No OMG involvement
- Organizational vocabulary plus use of UML

# Key Elements

- "Use-case driven
- Architecture centric
- Iterative and incremental"

# Use-Case Driven

- A *use case* is a sequence of actors and actions
  - Related to user stories and scenarios
- The set of use cases form the *use case model*
  - Represented in UML with a Use-Case diagram
- They serve as a user-centered functional specification
  - Similar to a data flow diagram
- They drive the development process
- They are developed in tandem with the system architecture

# Architecture-Centric

- Highest level of system design
- Comprises different views, including static and dynamic
- Can include the platform, framework (reusable resources), deployment and non-functional requirements
- May describe use-case independent components
  - *E.g.* those mandated by the platform
- Key use cases suggest structural units
  - Components and classes
- The architecture is refined as use cases are refined

# Iterative and Incremental

- *Iteration* is a unit of process; e.g. a subproject
- *Increment* is a unit of product; such as a partial system
- Each iteration is based on a set of use cases selected for risk mitigation
- May refine (improve) or add to (provide additional functionality to) existing increments

# Vocabulary

(from Jacobsen et al.)

- *Project: A development effort taking a system through a software life cycle*
  - Comprises a set of cycles
- *Software Life Cycle: A cycle over four [the four] phases in the following order: inception | elaboration | construction | transition*
  - Each cycle releases a production-ready version of the project

# Vocabulary – 2

- *Phase: The span of time between two major milestones of a development process*
  - Comprises a series of iterations
  - Each phase addresses each of the five workflow, albeit not equally
  - Terminates in a milestone
- *Iteration: A distinct set of activities conducted according a devoted (iteration) plan and evaluation criteria that results in a release, either internal or external*

# Vocabulary - 3

- *Workflow: A realization of (a part of) a business use case. Can be described in terms of activity diagrams that include participating workers, the activities they perform, and the artifacts they produce*
  - Comprises a set of Activities
  - UDSP has five kinds of workflows: requirements | analysis | design | implementation | testing

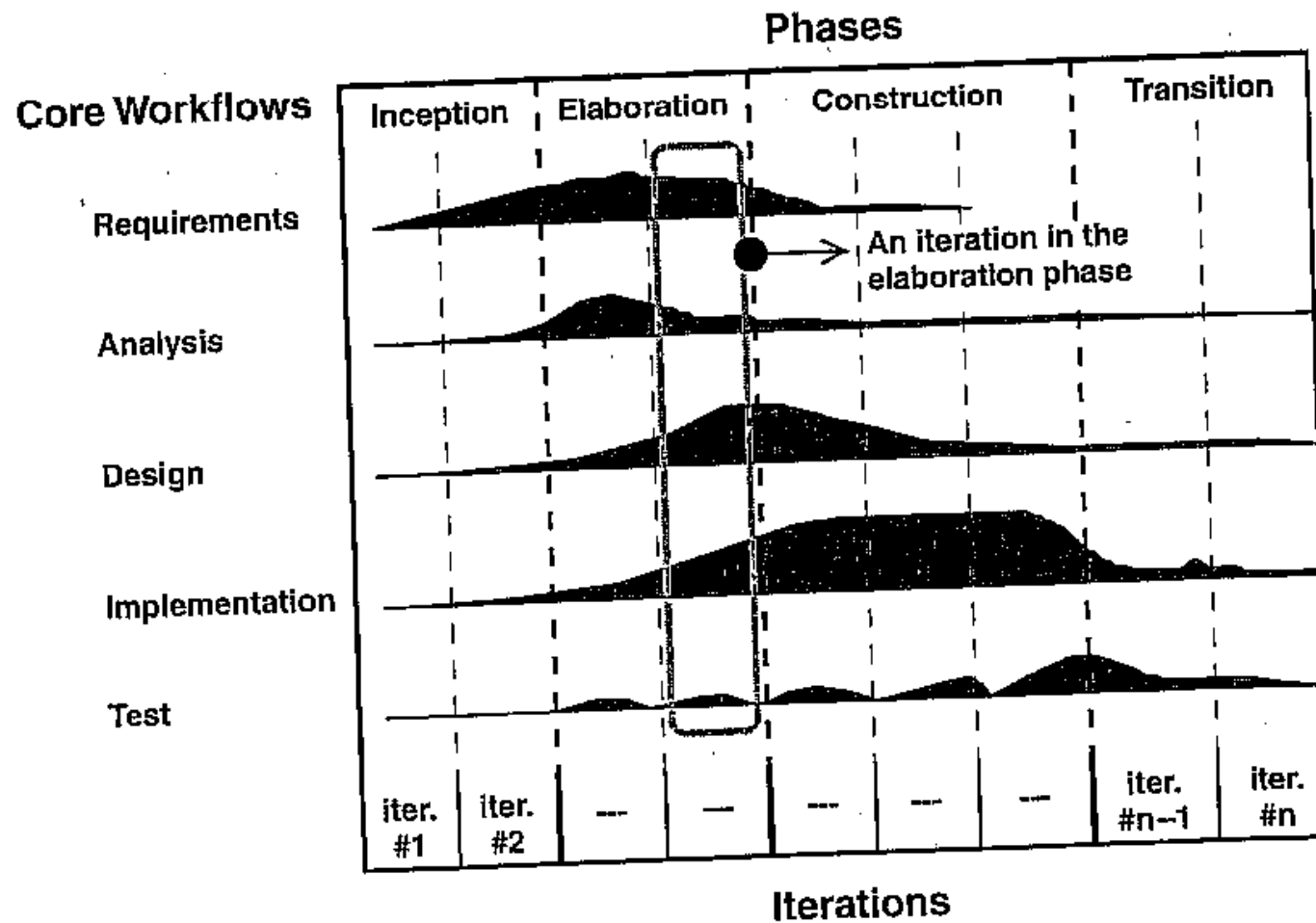
# Vocabulary - 4

- *Activity: A tangible unit of work performed by a worker in a workflow that (1) implies a well-defined responsibility for the worker, (2) yields a well-defined result (a set of artifacts) based on a well-defined input (another set of artifacts), and (3) represents a unit of work with crisply defined boundaries that is likely to be referred to in a project plan when tasks are assigned to individuals*
  - Finer-grained than what we have used so far

# Vocabulary - 5

- *(Software Development) Process: A concept that works as a template that can be reused by creating instances of it. The definition of a complete set of activities needed to transform users' requirements into a consistent set of artifacts that represent a software product and, later, to transform changes in those requirements into a new, consistent set of artifacts*
  - Should be specialized to an organization and project's needs

# Workflows, Phases & Iterations



# Requirements Workflow

(from *Jacobsen et al.*)

Work to be done	Resulting Artifacts
List candidate requirements	Feature list
Understand system context	Business or domain model
Capture functional requirements	Use-case model
Capture non-functional requirements	Supplementary requirements or individual use cases (for use-case specific requirements)

# Requirements Workflow - 2

- Artifacts: Use-case model, actor, architecture description, glossary, user interface prototype
- Workers: system analyst, use-case specifier, user interface designer, architect
- Activities: find actors and use cases, prioritize use cases, detail a use case, prototype user interface, structure the use-case model

# Analysis Workflow

- Conceptual structural model (UML)
  - Boundary, control and entity (persistent information) classes
- Refinement and structuring of requirements
  - Added precision
- First cut at design (1:5 ratio of classes)
- Artifacts: analysis model, use-case realization, analysis package, architectural description
- Workers: Architect, use-case engineer, component engineer
- Activities: architectural analysis, analyze a use case, analyze a class, analyze a package

# Design Workflow

- Understand non-functional requirements, specify implementation, decompose implementation work, specify interfaces, create diagram, create *seamless abstraction* (metaphor?)

# Design Workflow - 2

- Artifacts: design model, design class, use-case realization, design subsystem, interface, architecture description (design), deployment model, architecture description (deployment)
- Workers: architect, use-case engineer, component engineer
- Activities: architecture design, design a use case, design a class, design a subsystem

# Implementation Workflow

- Plan integration of increment, determine physical deployment, implement classes and subsystems, unit test
- Artifacts: implementation model, subsystem, interface, architecture description, build plan
- Workers: architect, component engineer, system integrator
- Activities: architecture implementation, system integration, subsystem implementation, class implementation, unit test

# Test Workflow

- Iteration test plan, test implementation, test performance
- Artifacts: test model, test case, test procedure, test component (automation), test plan, defect, test evaluation
- Workers: test designer, component engineer, integration tester, system tester
- Activities: plan test, design test, implement test, perform integration test, perform system test, evaluate test

# Overall Work Products

- Requirements
- Use cases
- Nonfunctional specification
- Test cases
- Source code
- Manuals

- Architecture (UML)
- Visual models (UML)
  - Analysis
  - Design (static and dynamic)
  - Implementation (packages)
  - Deployment (machines and processes)

# Phases

- *Inception*
  - Conception, vision, business case, risk analysis, initial project planning, vague use cases and architecture
- *Elaboration*
  - Use cases and architecture refined; modeling, selection of key use cases; refined plan
- *Construction*
  - Development
- *Transition*
  - Beta release, minor revisions, manufacturing, user training, etc.

# Summarizing Themes

- Get the requirements right
- Get the architecture right
- Use components
- Think and communicate in UML
- Iterate
- Manage risks