

# Introduction to Rational Unified Process

---



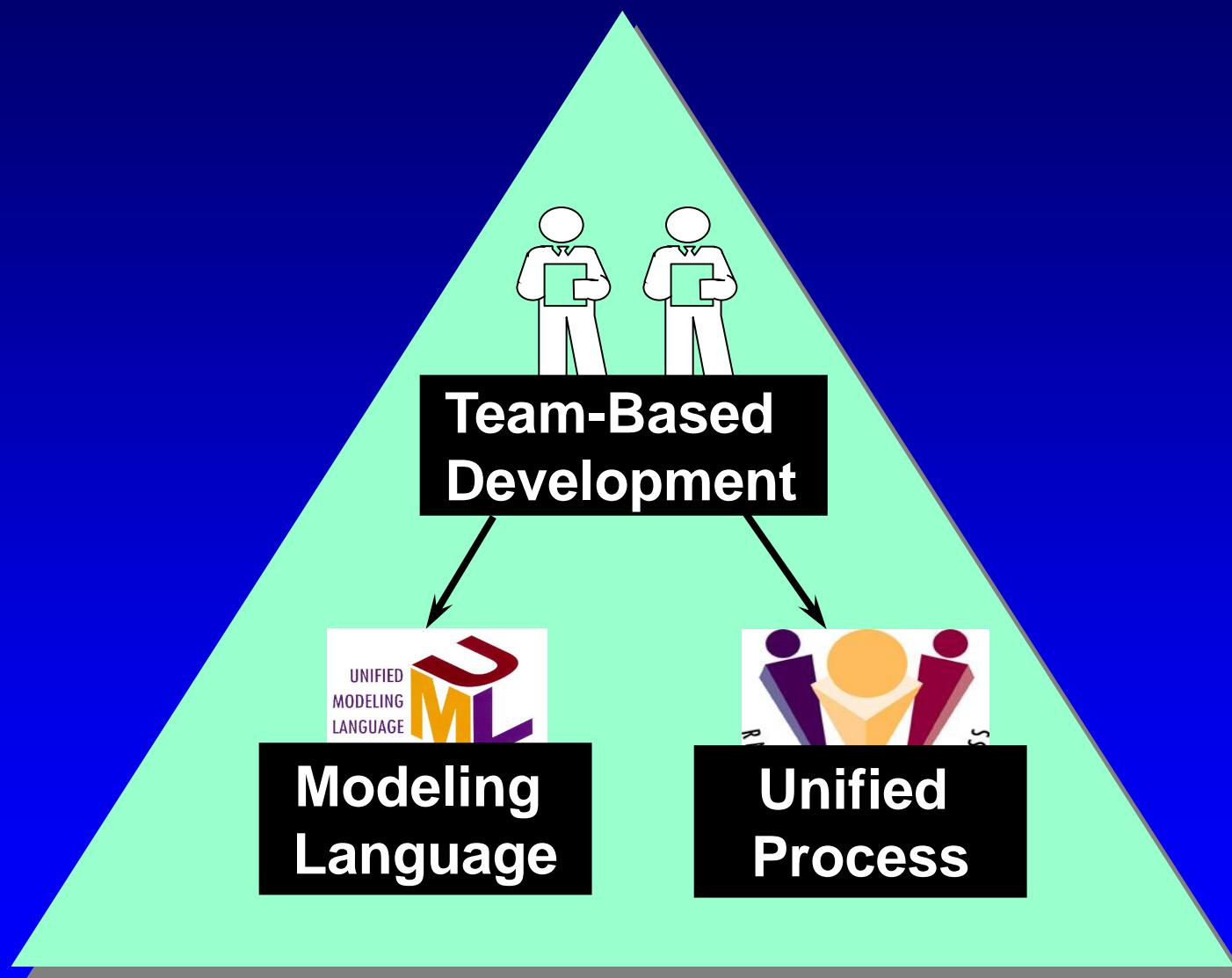
# Objectives: Rational Unified Process

---



- ◆ Describe the **Unified Modeling Language (UML)**
- ◆ Define what a **software development process** is
- ◆ Describe the **Rational Unified Process**
- ◆ Explain the four **phases** of the Rational Unified Process and their associated milestones
- ◆ Define **iterations** and their relation to phases
- ◆ Explain the relations between:
  - Models and workflows
  - Phases, iterations, and workflows
- ◆ Define **artifact**, **worker**, and **activity**
- ◆ State the importance of automated tool support

# In Building a System, a Language Is Not Enough



# What Is the UML?

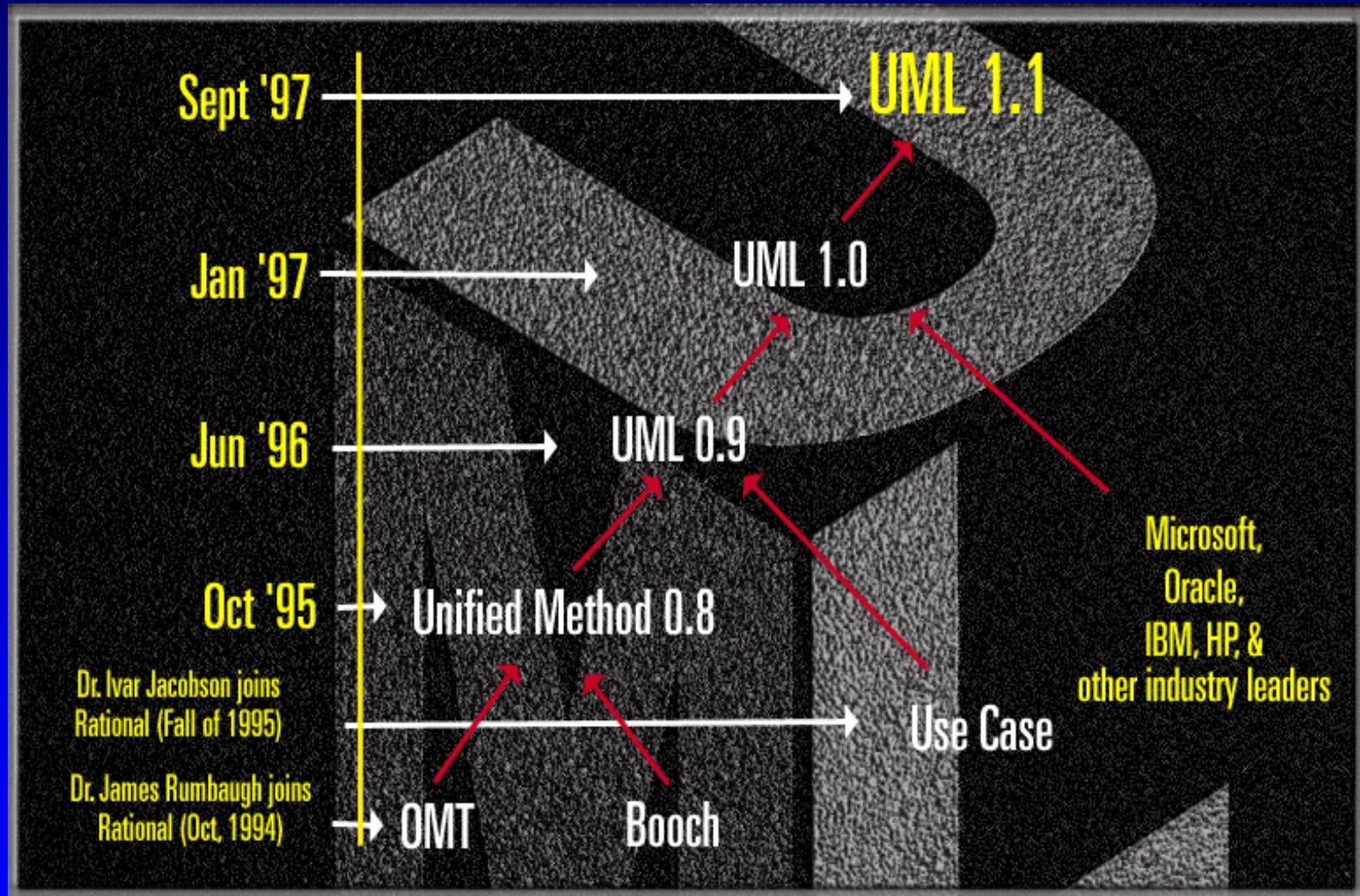
---



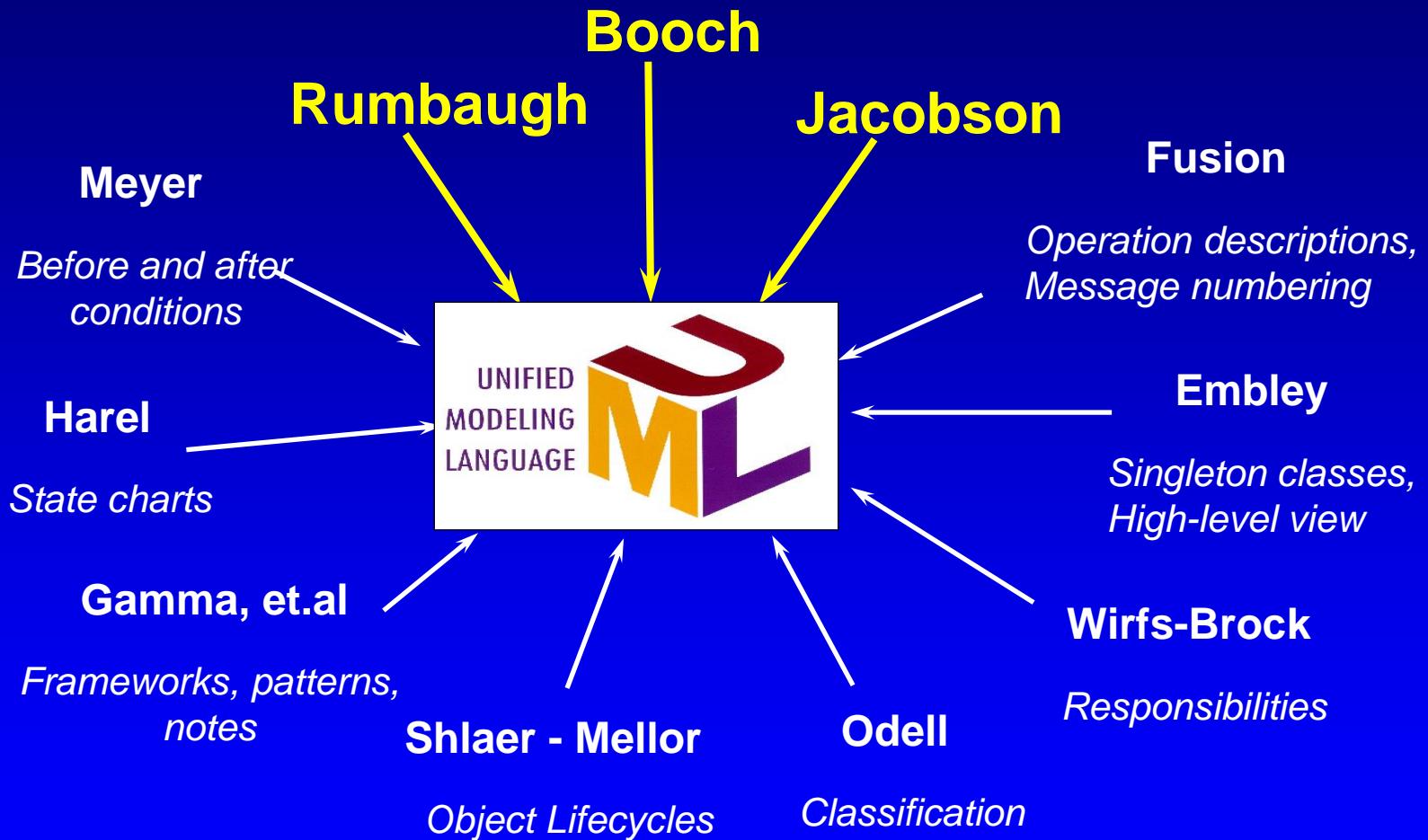
- ◆ The Unified Modeling Language (UML) is a language for
    - Specifying
    - Visualizing
    - Constructing
    - Documenting
- the artifacts of a software-intensive system



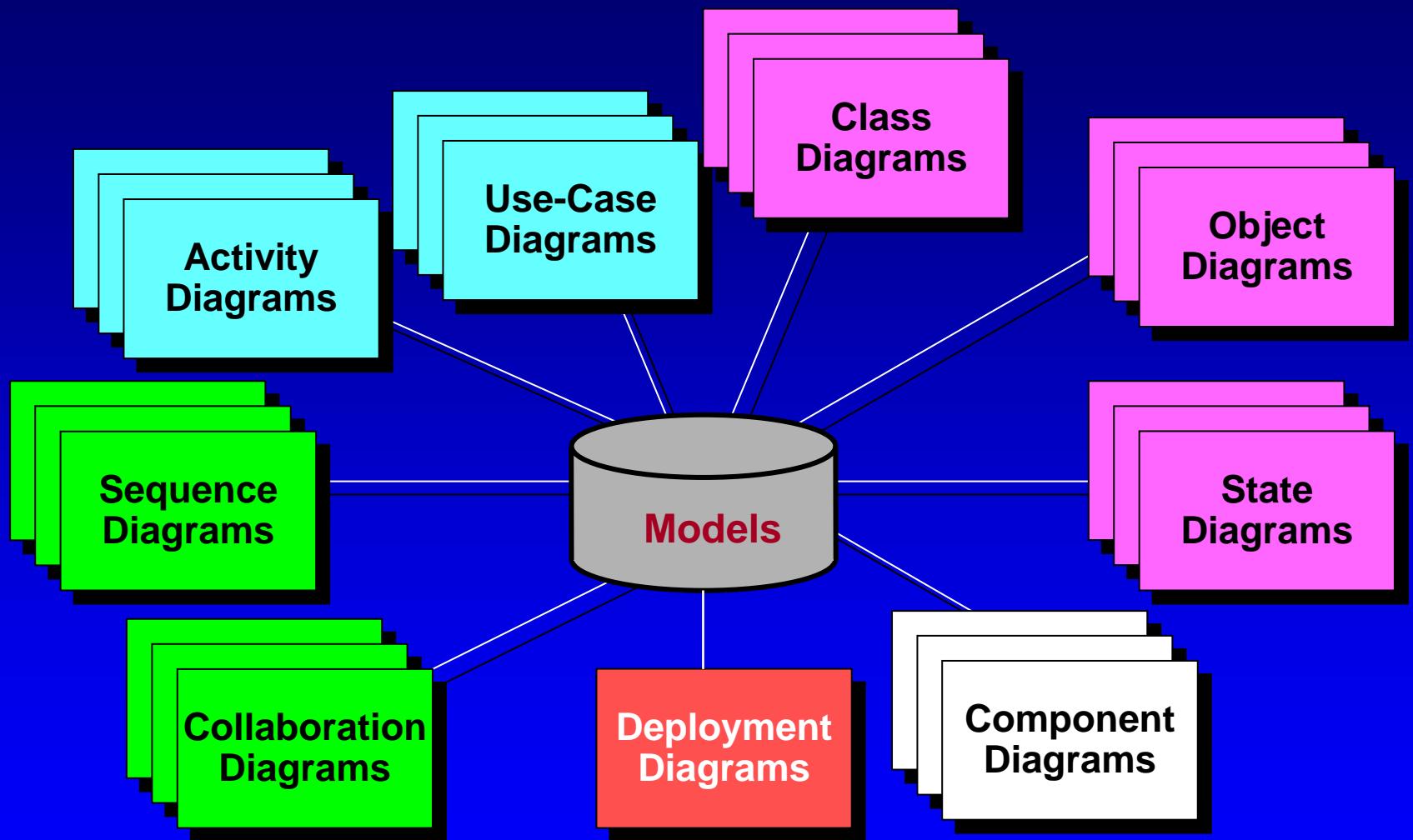
# UML History



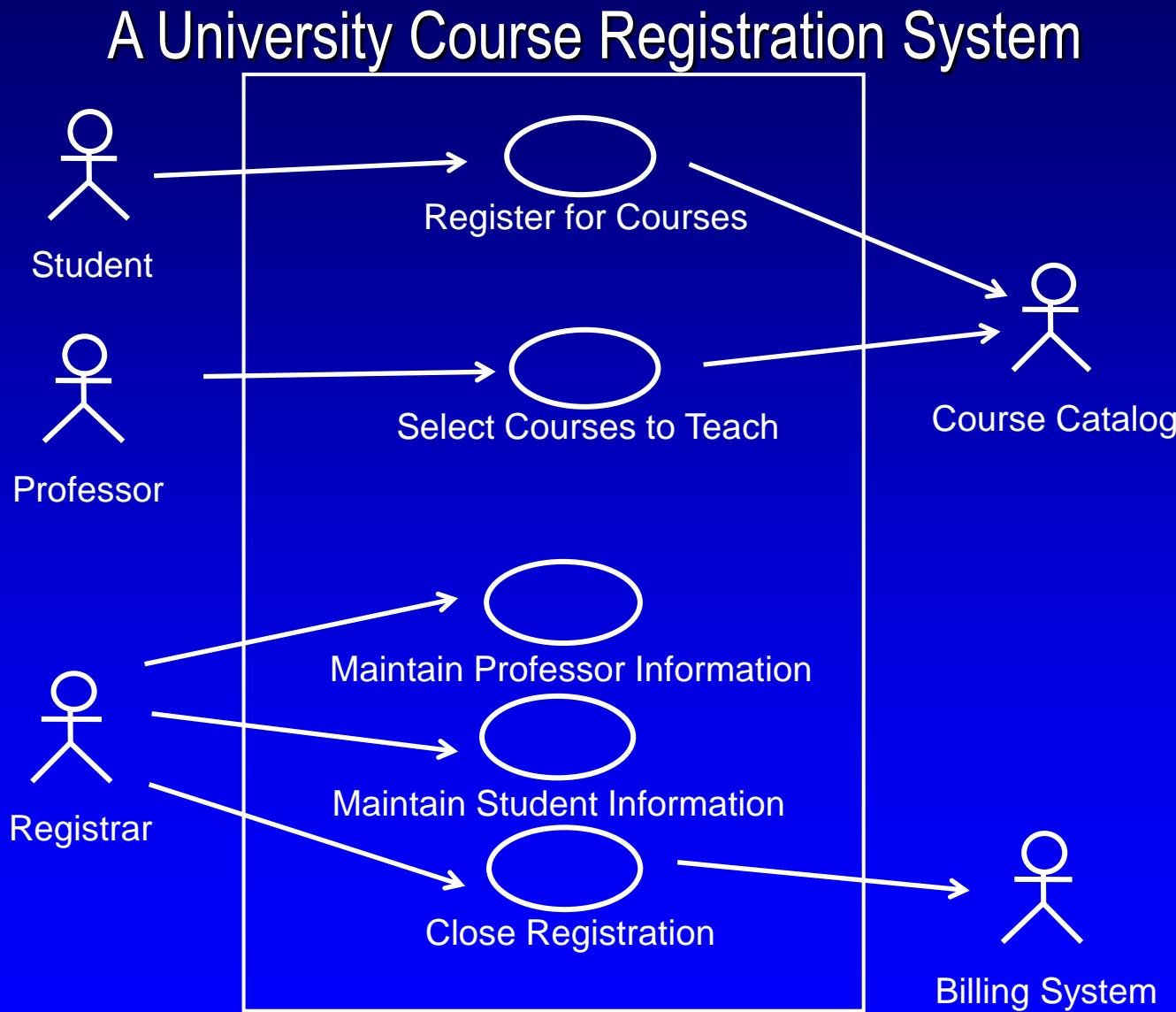
# Inputs to UML



# The UML Provides Standardized Diagrams

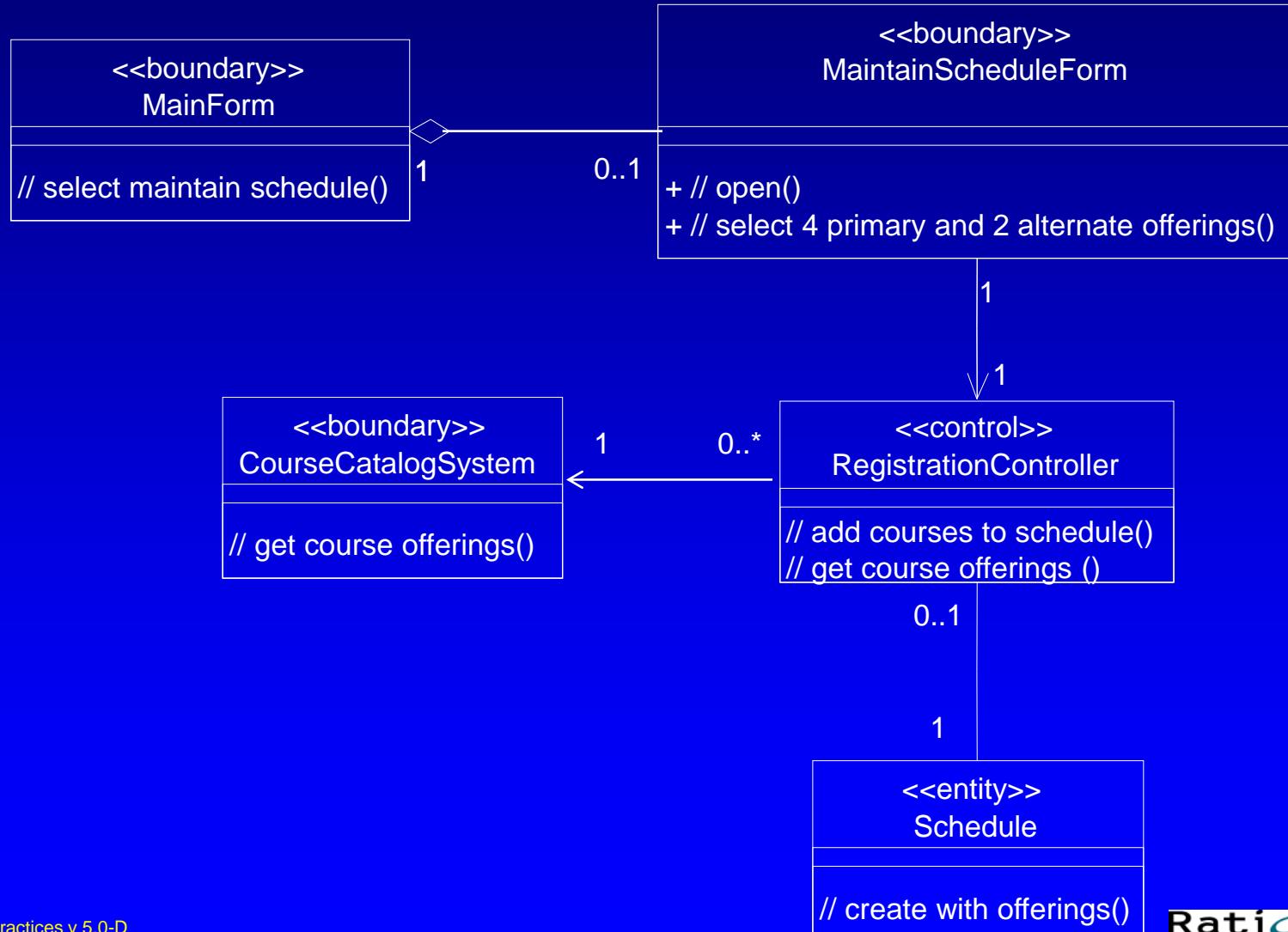


# A Sample UML Diagram: Use-Cases

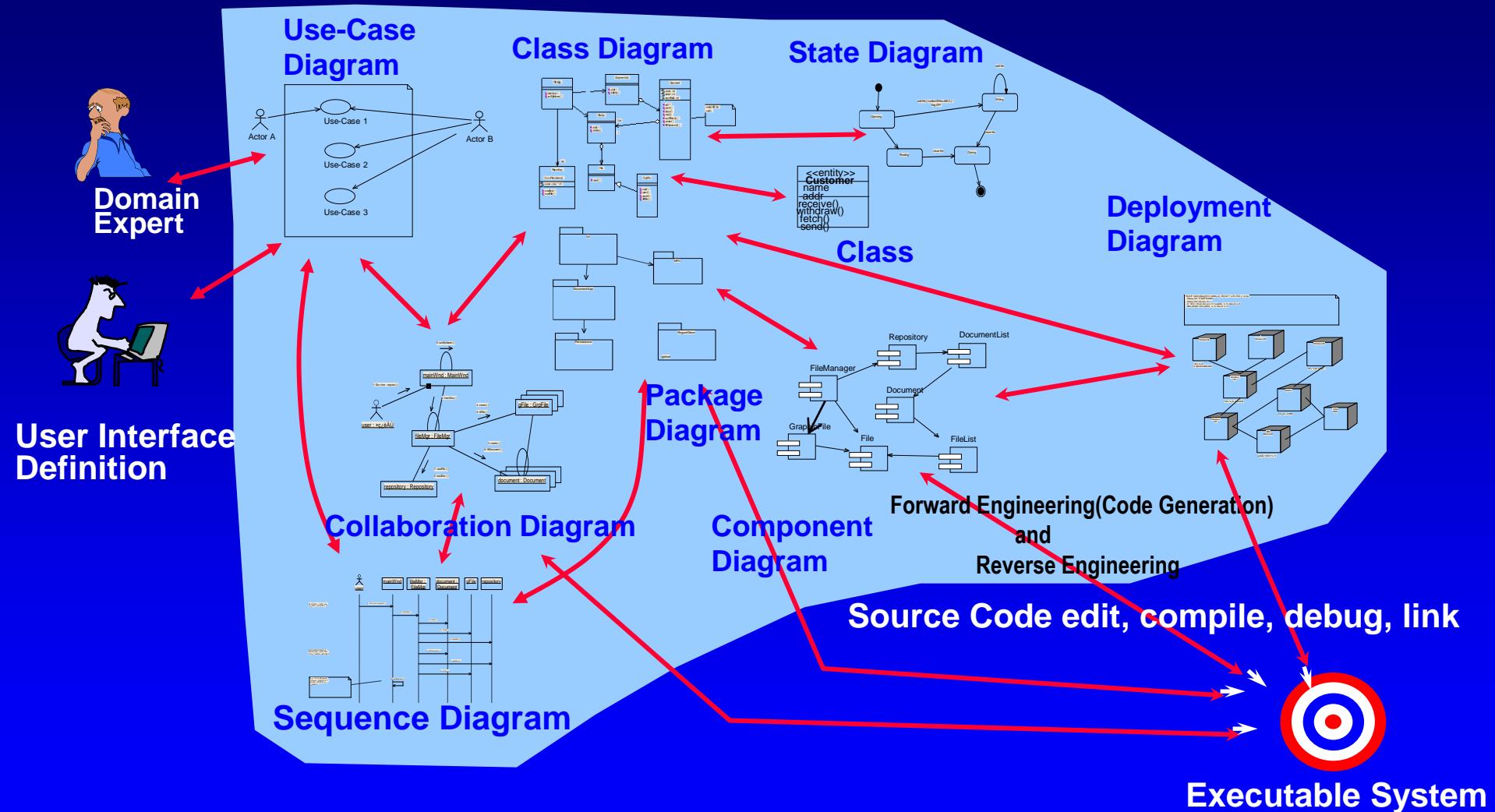


# A Sample UML Diagram: Classes

## A University Course Registration System



# UML Diagrams Are Key Artifacts



# What Is a Process?

---



A process defines **Who** is doing **What**, **When** and **How** to reach a certain goal. In software engineering the goal is to build a software product or to enhance an existing one



# An Effective Process ...

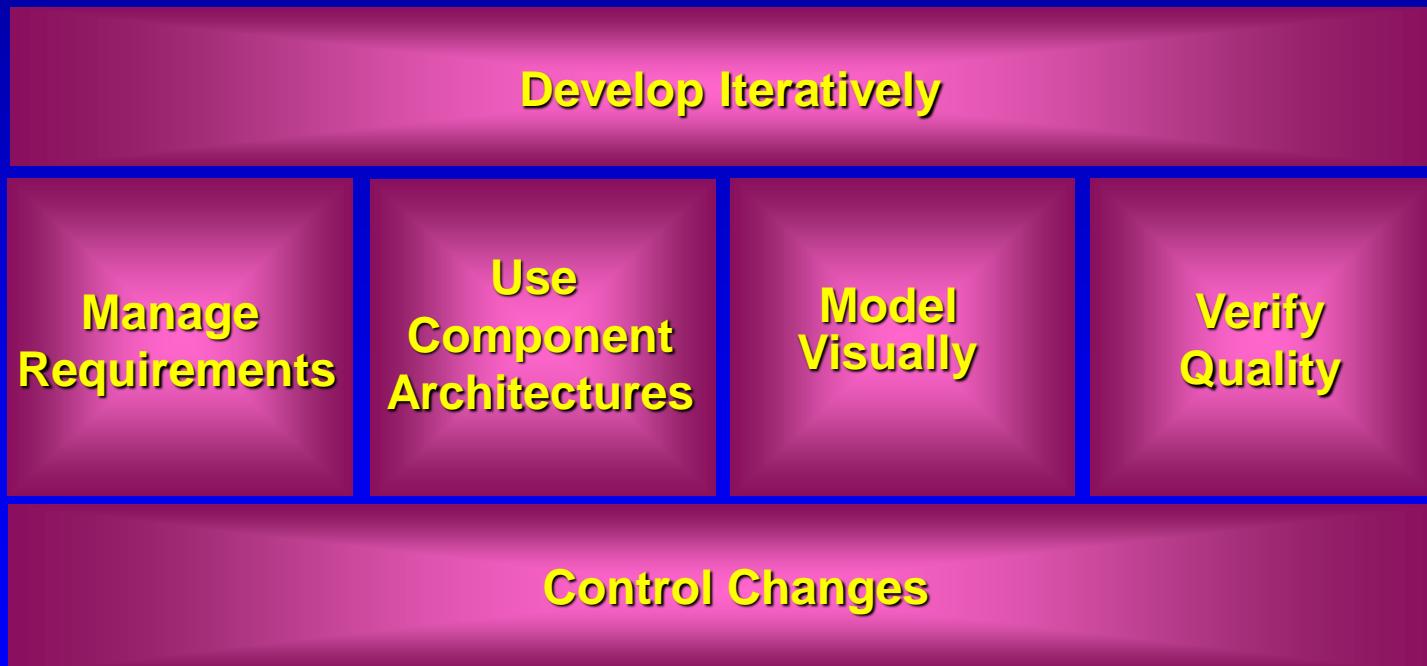
---



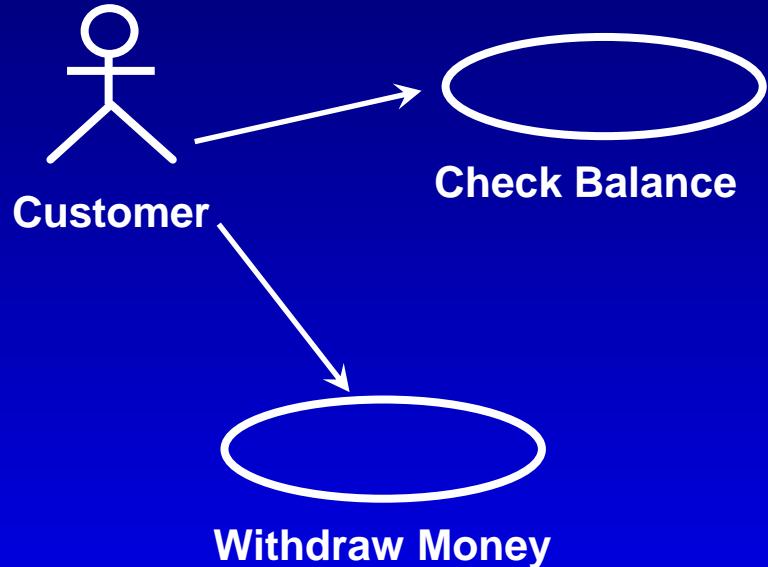
- ◆ Provides guidelines for efficient development of quality software
- ◆ Reduces risk and increases predictability
- ◆ Captures and presents best practices
  - Learn from other's experiences
  - Mentor on your desktop
  - Extension of training material
- ◆ Promotes common vision and culture
- ◆ Provides roadmap for applying tools
- ◆ Delivers information on-line, at your finger tips

# Rational Unified Process Delivers Best Practices

Rational Unified Process describes how to effectively implement the six best practices for software development



# Rational Unified Process Is Use-Case Driven



## Use-Cases for a Cash Machine

❶ An **actor** is someone or something outside the system that interacts with the system

❷ A **Use-Case** is a sequence of actions a system performs that yields an observable result of value to a particular actor

# Use-Cases Include a Flow of Events

---



## Flow of events for the Withdraw Money Use-Case

1. The Use-Case begins when the customer inserts a cash card. The system reads and validates information on the card.
2. The system prompts for the PIN. The system validates the PIN.
3. The system asks which operation the customer wishes to perform. The customer selects “Cash withdrawal.”
4. The system requests the amount. The customer enters the amount.
5. The system requests the account type. The customer selects checking or savings.
6. The system communicates with the ATM network . . .

# Benefits of a Use-Case Driven Process

---



- ◆ Use-Cases are concise, simple, and understandable by a wide range of stakeholders
  - End users, developers and acquirers understand functional requirements of the system
- ◆ Use-Cases drive numerous activities in the process:
  - Creation and validation of the design model
  - Definition of test cases and procedures of the test model
  - Planning of iterations
  - Creation of user documentation
  - System deployment
- ◆ Use-Cases help synchronize the content of different models

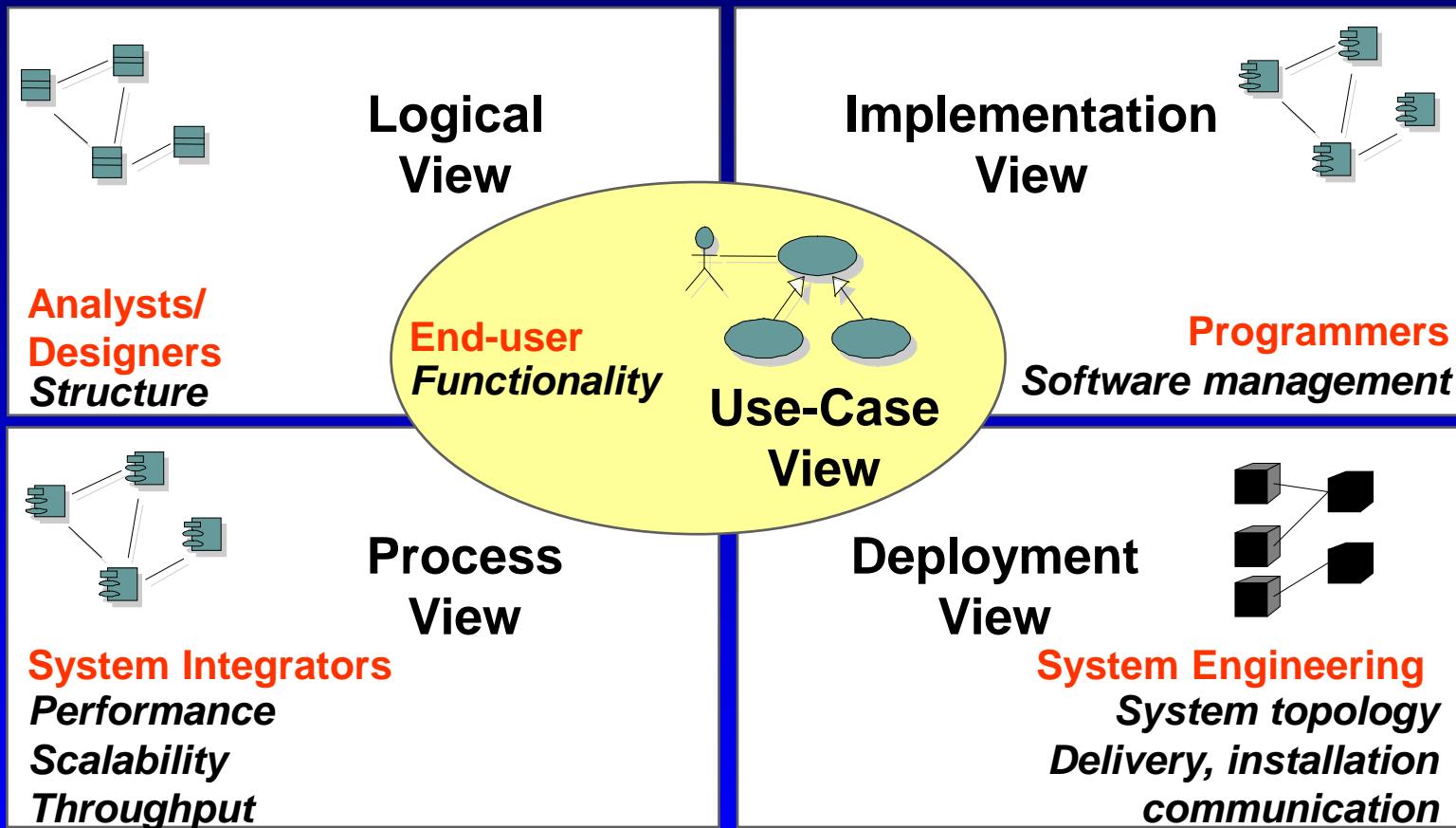
# Rational Unified Process Is Architecture-Centric

---



- ◆ Architecture is the focus of the early iterations
  - Building, validating, and baselining the architecture constitute the primary objective of elaboration
- ◆ The Architectural Prototype validates the architecture and serves as the baseline for the rest of development
- ◆ The Software Architecture Document is the primary artifact that describes the architecture chosen
- ◆ Other artifacts derive from architecture:
  - Design guidelines including use of patterns and idioms
  - Product structure
  - Team structure

# Representing Architecture: The 4+1 View Model



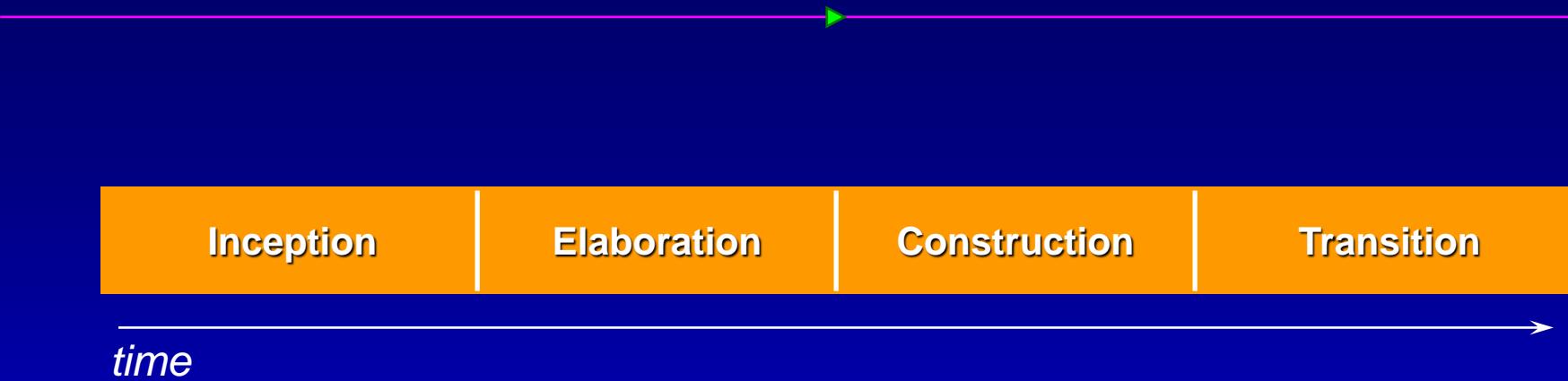
# Benefits of an Architecture-Centric Process

---



- ◆ Lets you gain and retain intellectual control over a project, to manage its complexity, and to maintain system integrity
- ◆ Provides an effective basis for large-scale reuse
- ◆ Provides a basis for project management
- ◆ Facilitates component-based development
  - A component fulfills a clear function in the context of a well-defined architecture
  - A component conforms to and provides the physical realization of a set of interfaces
  - Components exist relative to a given architecture

# Process Architecture - Lifecycle Phases



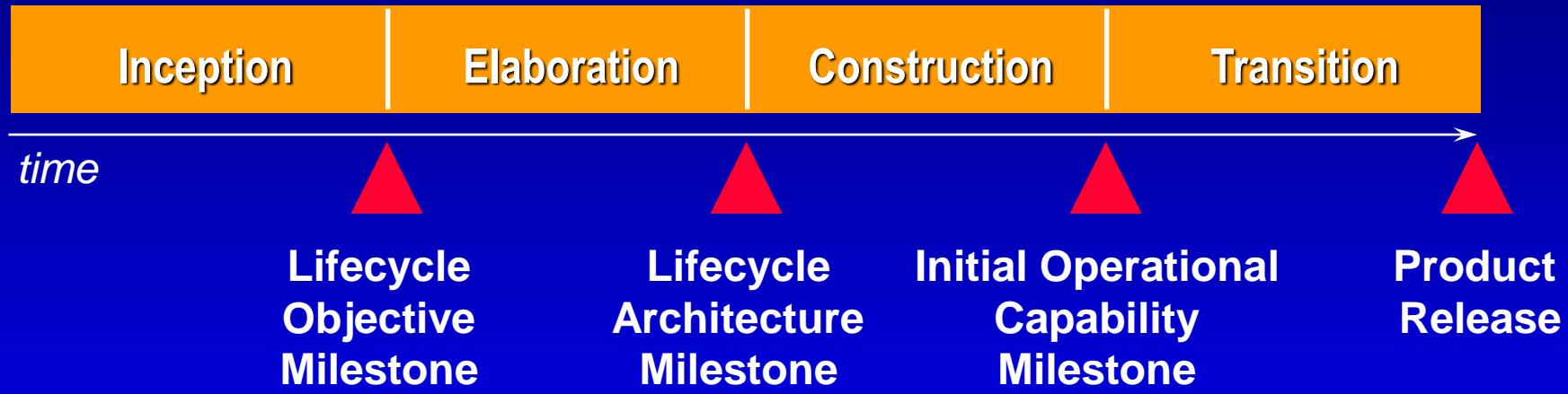
```
graph LR; Inception[Inception] --- Elaboration[Elaboration]; Elaboration --- Construction[Construction]; Construction --- Transition[Transition];
```

The Rational Unified Process has four phases:

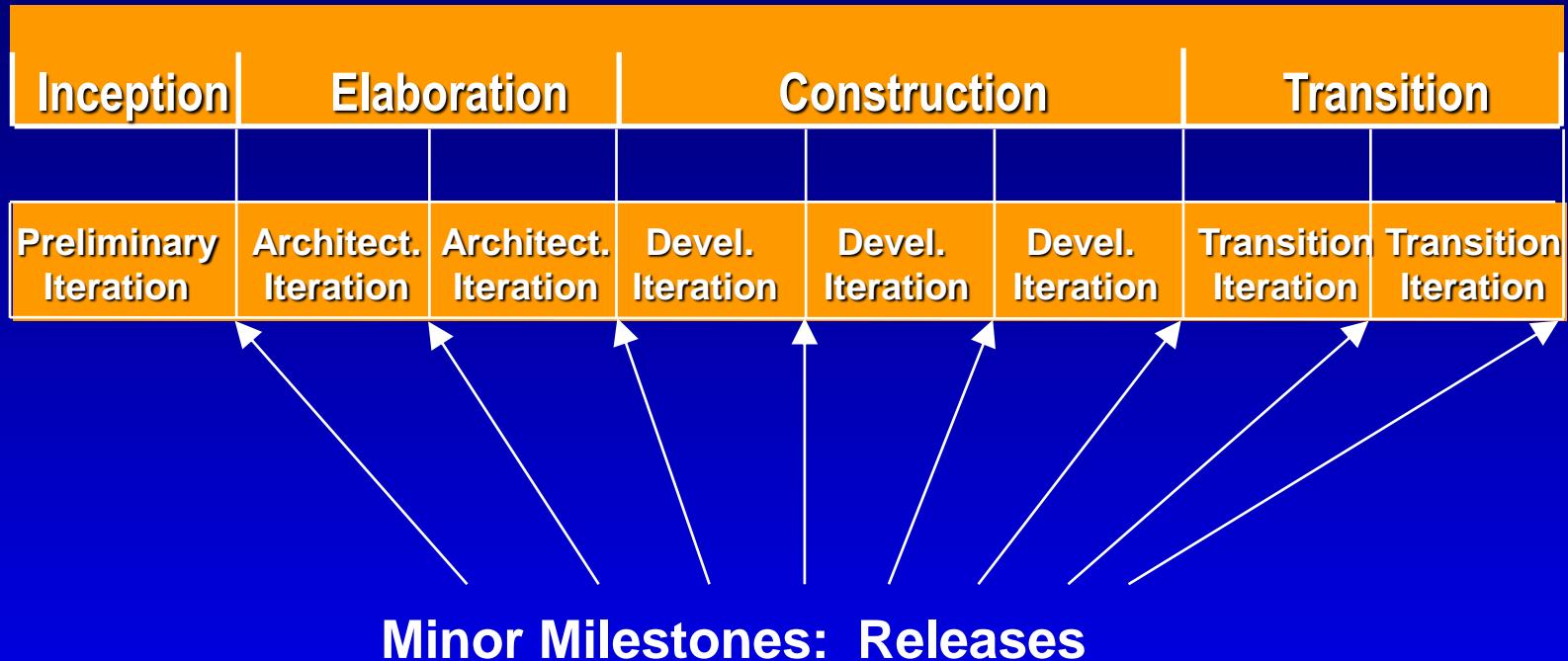
- **Inception** - Define the scope of project
- **Elaboration** - Plan project, specify features, baseline architecture
- **Construction** - Build the product
- **Transition** - Transition the product into end user community

# Phase Boundaries Mark Major Milestones

---



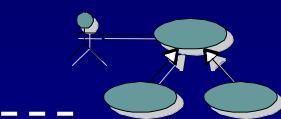
# Iterations and Phases



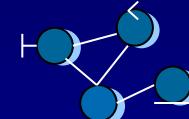
An **iteration** is a distinct sequence of activities with an established plan and evaluation criteria, resulting in an executable release (internal or external)

# Major Workflows Produce Models

Business Modeling



Business Use-Case Model



Business Object Model

*automated by*

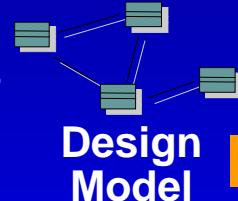
Requirements



Use-Case Model

*realized by*

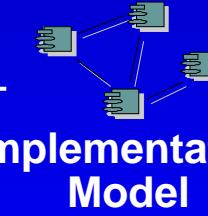
Analysis & Design



Design Model

*implemented by*

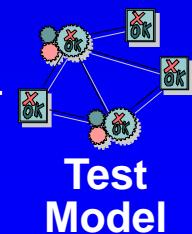
Implementation



Implementation Model

*verified by*

Test



Test Model

# Bringing It All Together: The Iterative Model

## Process Workflows

Business Modeling

Requirements

Analysis & Design

Implementation

Test

Deployment

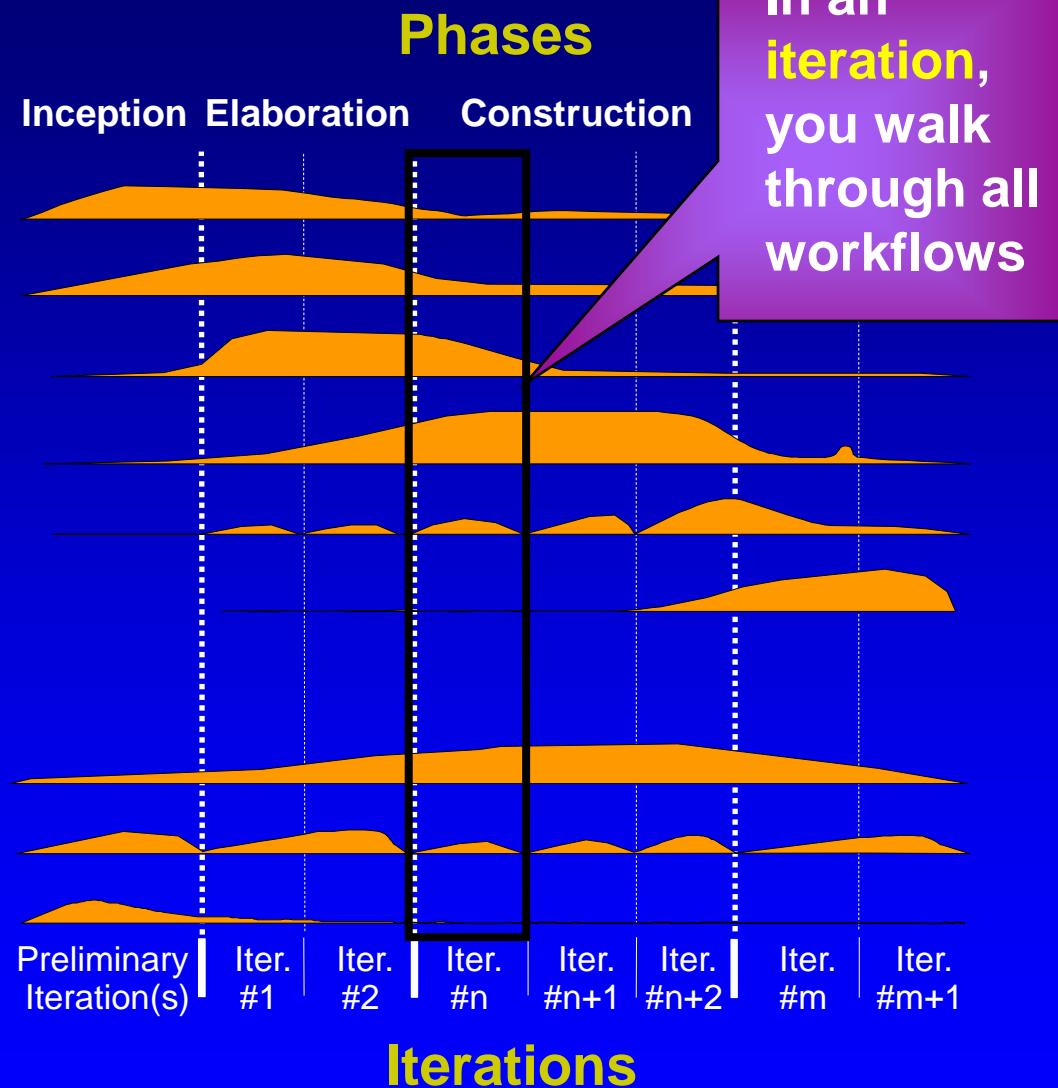
## Supporting Workflows

Configuration & Change Mgmt

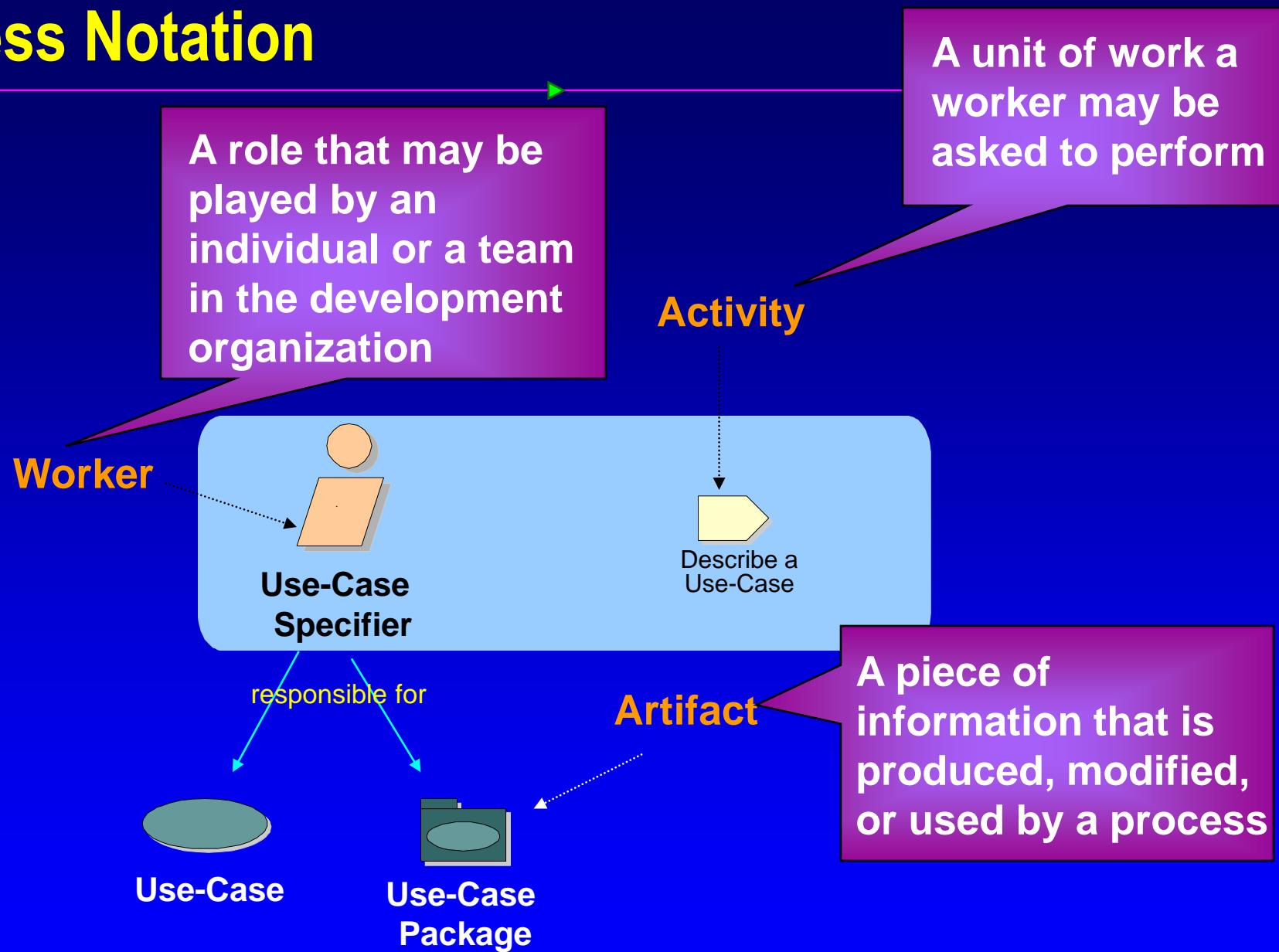
Project Management

Environment

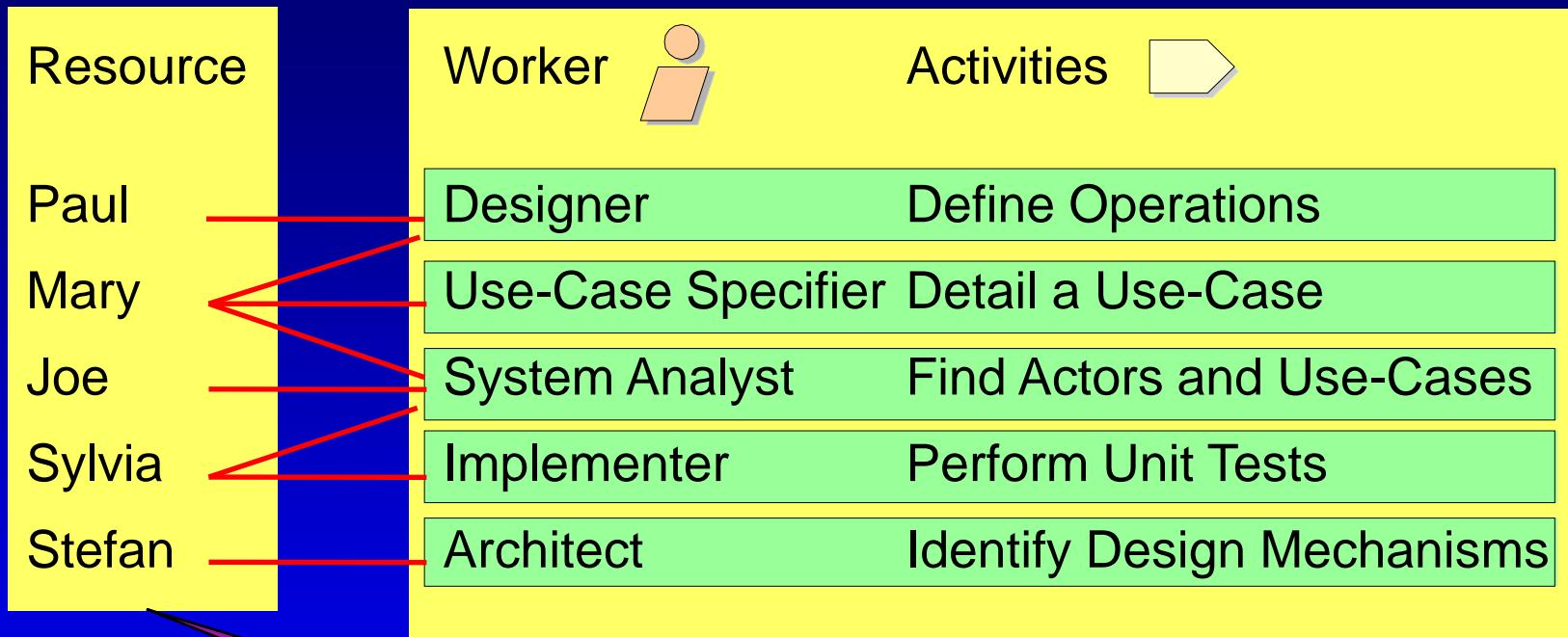
Workflows group activities logically



# Process Notation

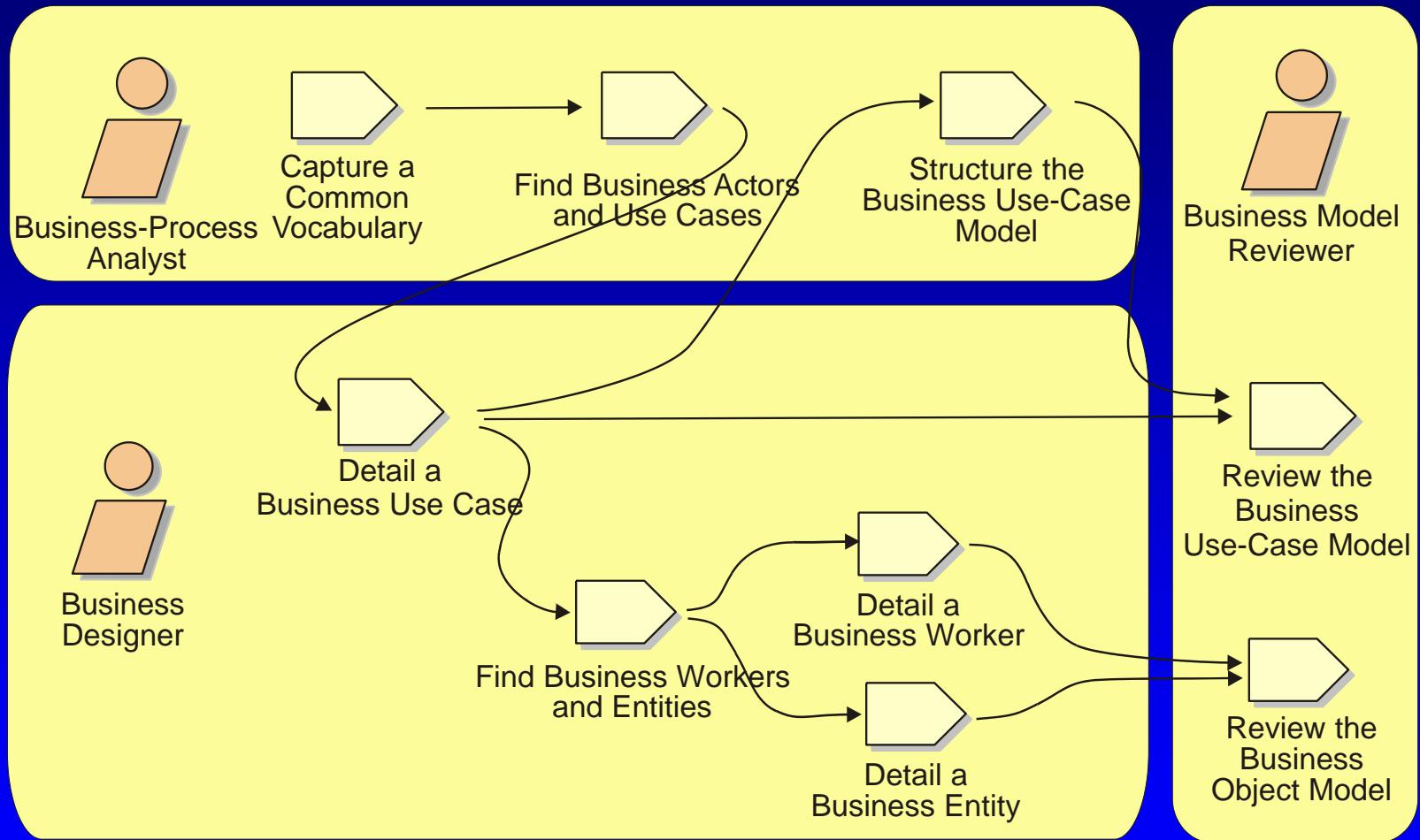


# Workers Are Used for Resource Planning

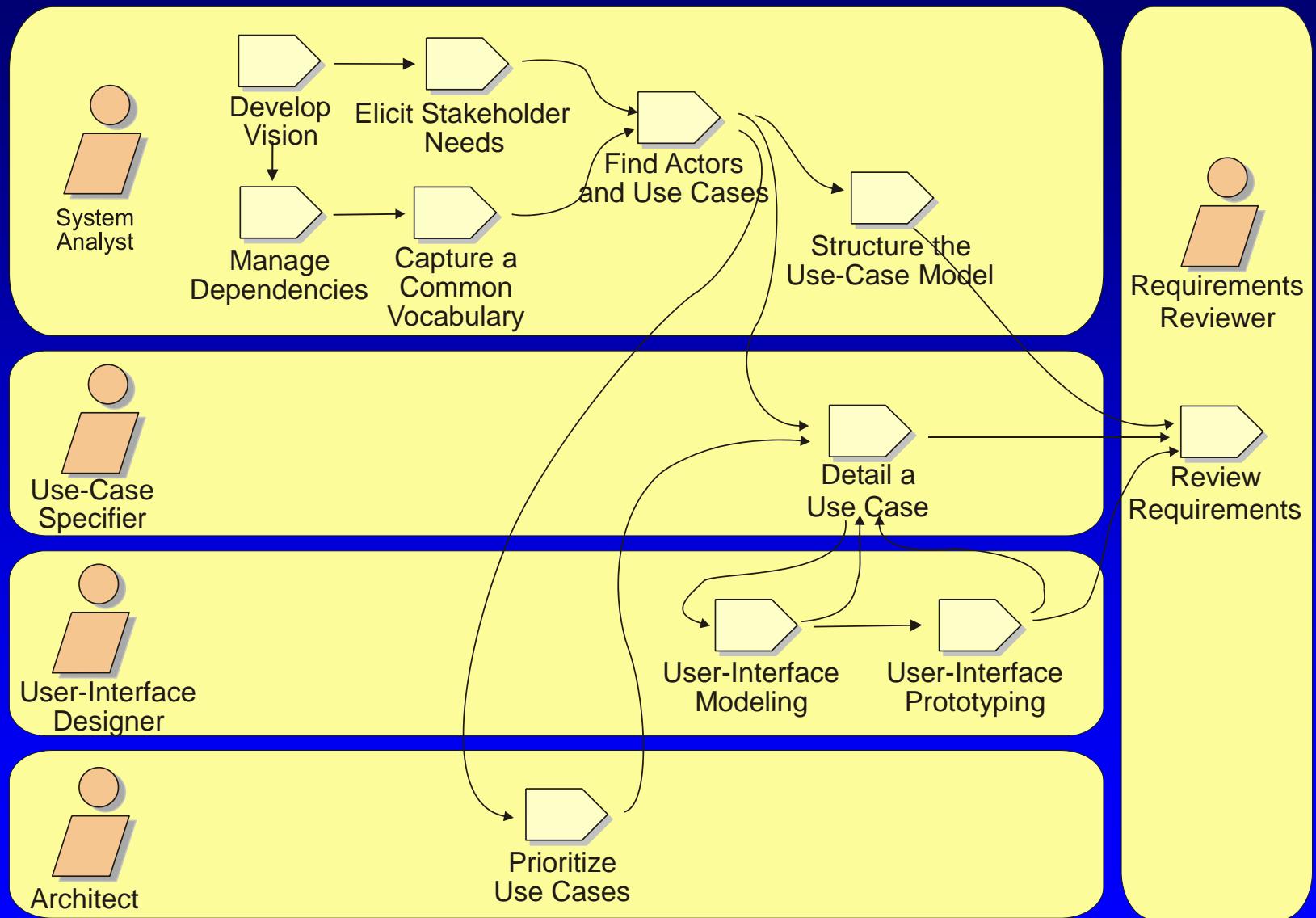


Each **individual** in  
the project is  
assigned to one or  
several **workers**

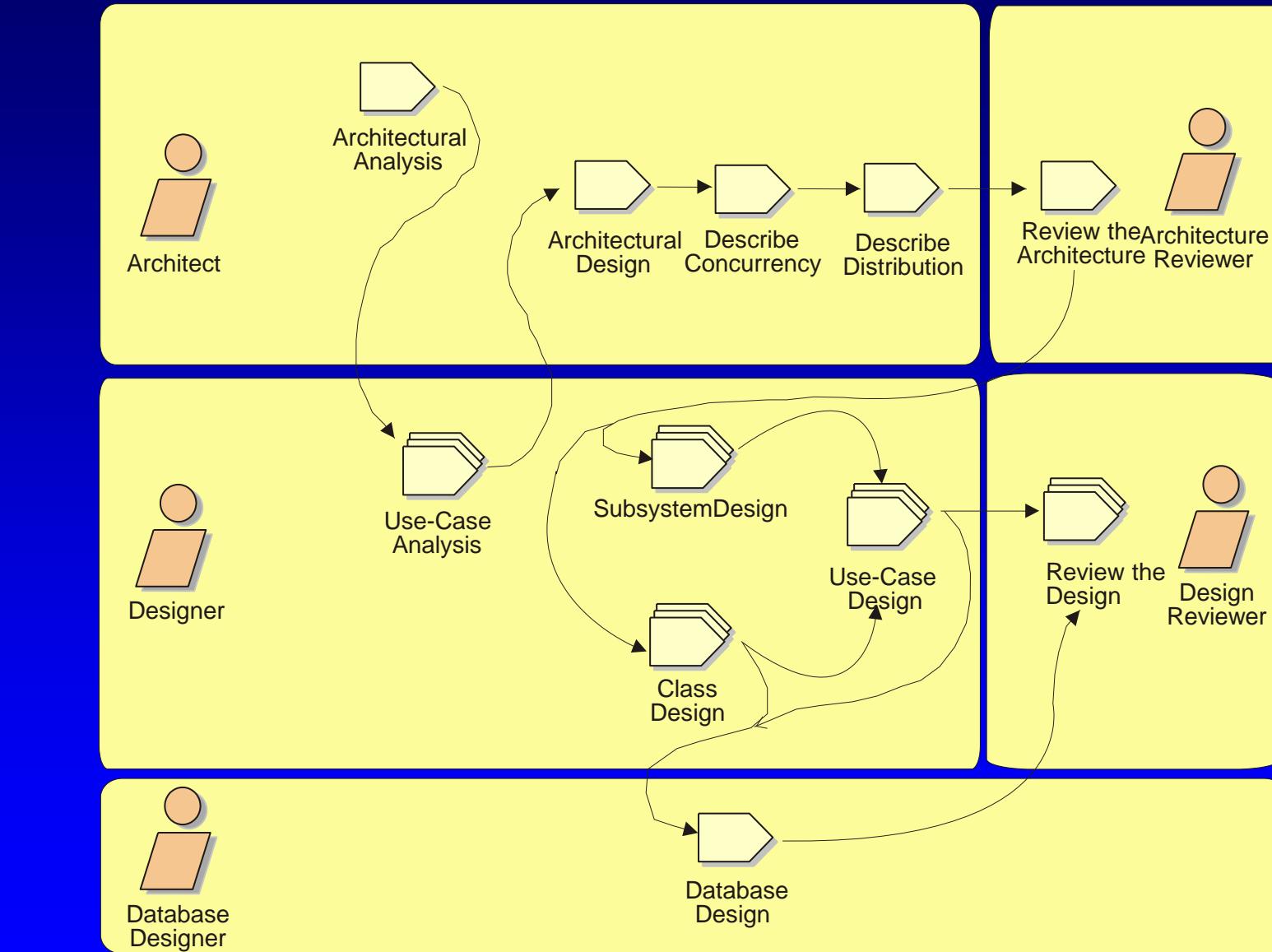
# Business Modeling Workflow



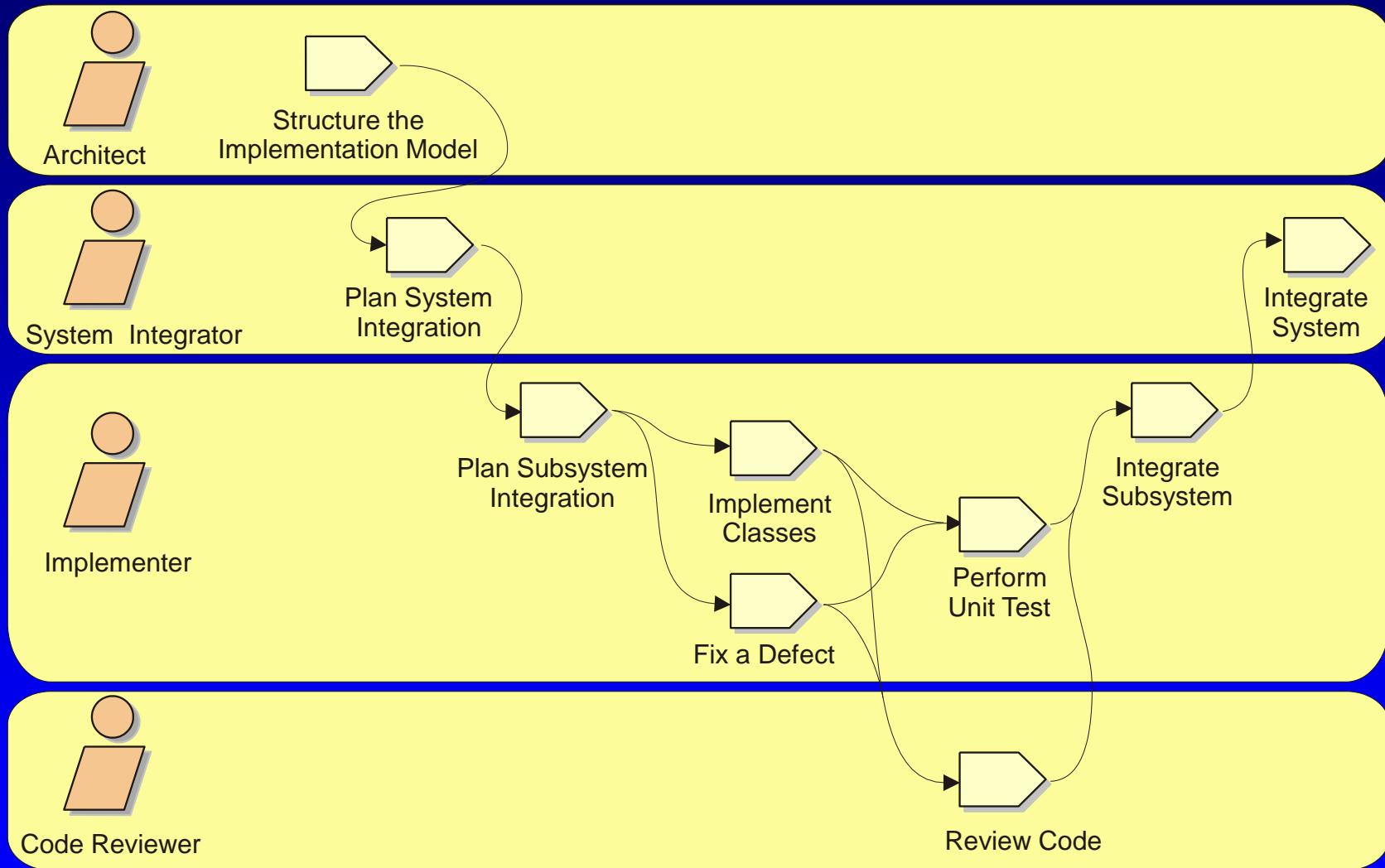
# Requirements Workflow



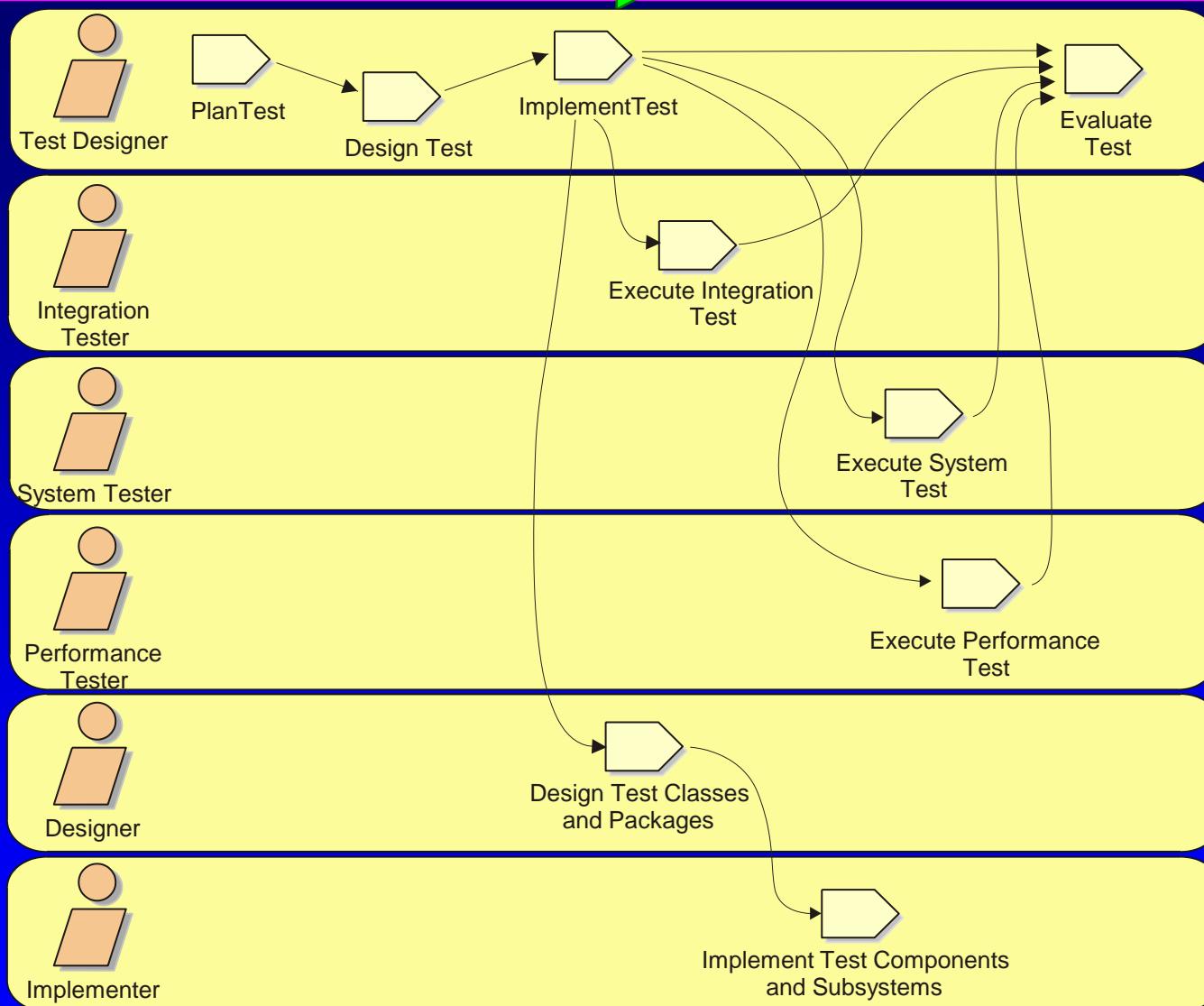
# Analysis & Design Workflow



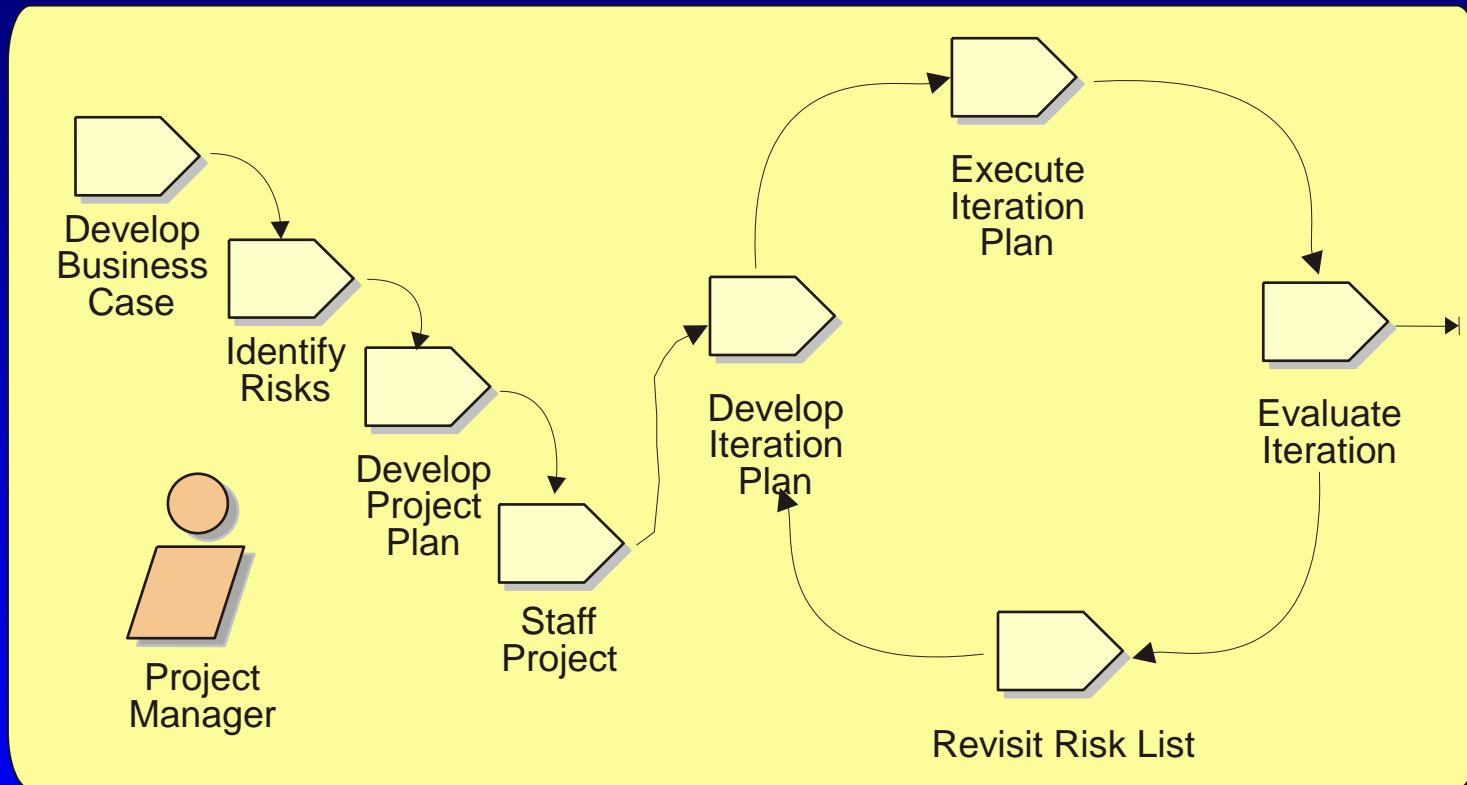
# Implementation Workflow



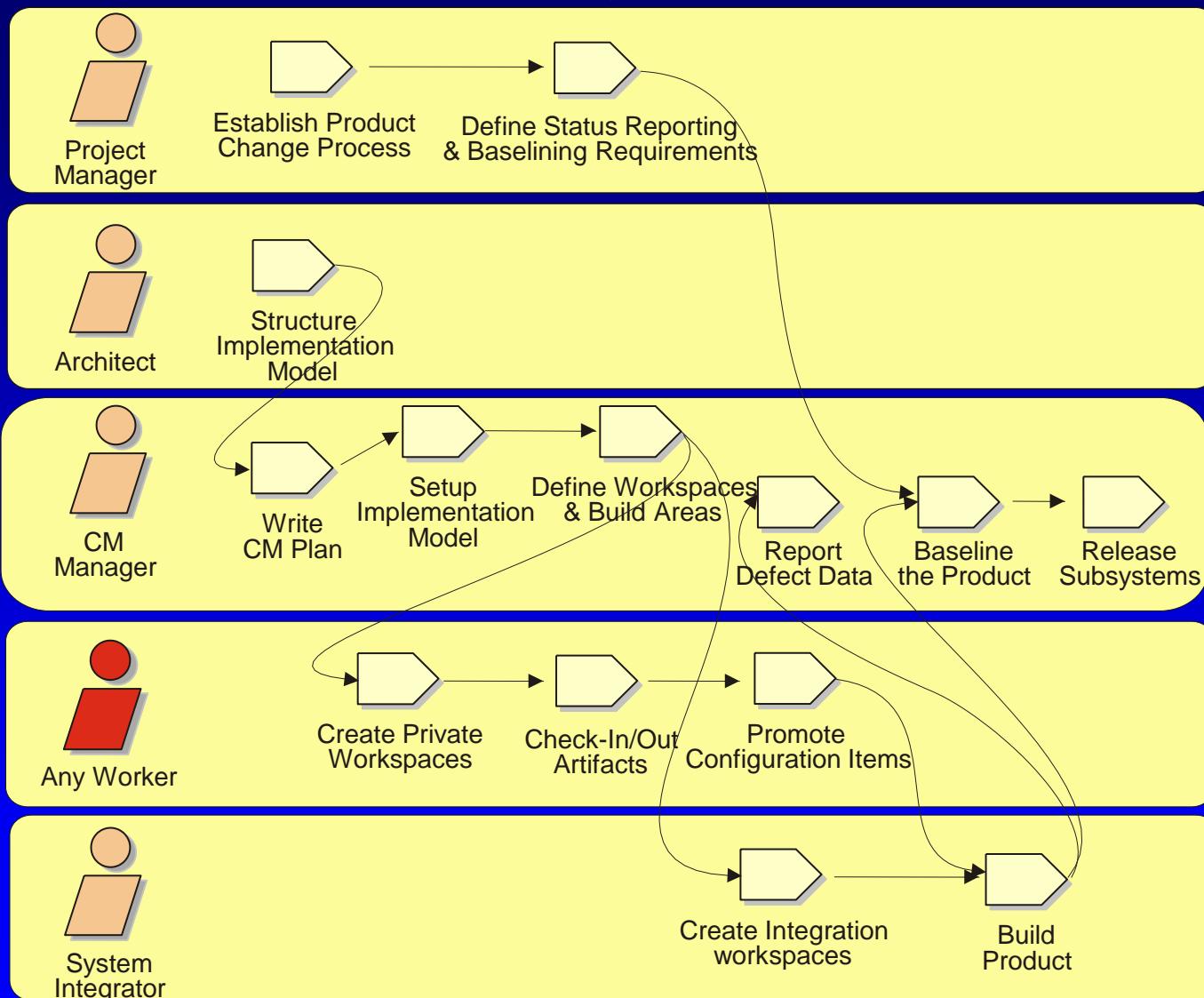
# Test Workflow



# Project Management Workflow



# Configuration and Change Management Workflow



# Environment Workflow

---



- ◆ Configuring the process
- ◆ Improving the process
- ◆ Selecting and acquiring tools
- ◆ Toolsmithing
- ◆ Supporting the development
- ◆ Training

# Guidelines, Mentors, and Templates

---



- ◆ Guidelines are the rules, recommendations, and heuristics that support activities
  - For example, modeling and programming guidelines
- ◆ Tool mentors explain how to use a specific tool to perform an activity or steps in an activity
  - For example, building a design model using Rational Rose
- ◆ Templates are predefined artifacts
  - For example, a Rational SoDA template for a Use-Case Report
- ◆ Guidelines, tool mentors and templates make it easier to apply the process correctly and consistently

# Tool Support for the Entire Project Lifecycle

## Process Workflows

Business Modeling	Requisite Pro, Rose, SoDA
Requirements	Requisite Pro, Rose, SoDA
Analysis and Design	Rose, SoDA, Apex
Implementation	Rose, Apex, SoDA, Purify, ...
Test	SQA TeamTest, Quantify, PerformanceStudio,...
Deployment	SoDA, ClearCase, ...

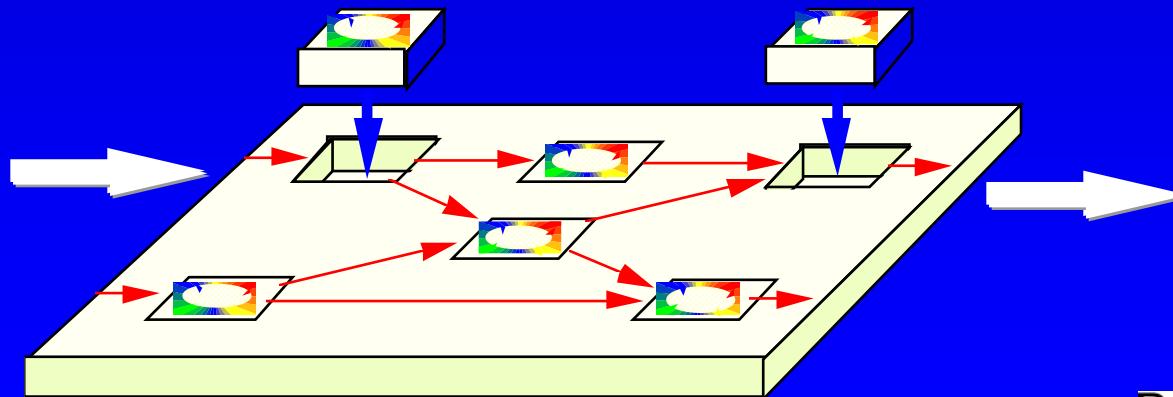
## Supporting Workflows

Config. & Change Mgmt.	ClearCase, ClearQuest
Project Management	Unified Process, Microsoft® Project, ...
Environment	Unified Process, Rational Tools

# Adopting a Process

---

- ◆ Process adoption includes configuring and implementing the process
- ◆ In **configuring** the process, the process framework is adapted to the needs and constraints of the adopting organization
  - The result is documented in a “Development Case”
- ◆ In **implementing** the process, the organization’s practice is changed to effectively use the process



# Summary: Rational Unified Process

---

- ◆ The Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of a software-intensive system
- ◆ A software development process defines **Who** is doing **What, When** and **How** in building a software product
- ◆ The Rational Unified Process has four phases: **Inception, Elaboration, Construction and Transition**
- ◆ Each phase ends at a major milestone and contains one or more iterations
- ◆ An **iteration** is a distinct sequence of activities with an established plan and evaluation criteria, resulting in an executable release

# Summary (cont.): Rational Unified Process

---

- ◆ A **workflow** groups related activities together
- ◆ Each workflow is exercised during an **iteration** and results in a model that is incrementally produced
- ◆ An **artifact** is a piece of information that is produced, modified, or used by a process
- ◆ A **worker** is a role that may be played by an individual or a team in the development organization
- ◆ An **activity** is a unit of work a worker may be asked to perform