UNIT-II

- ❑ Rational Unified Process
- ❑ 4+1 View Architecture
- ❑ Use Case Overview.

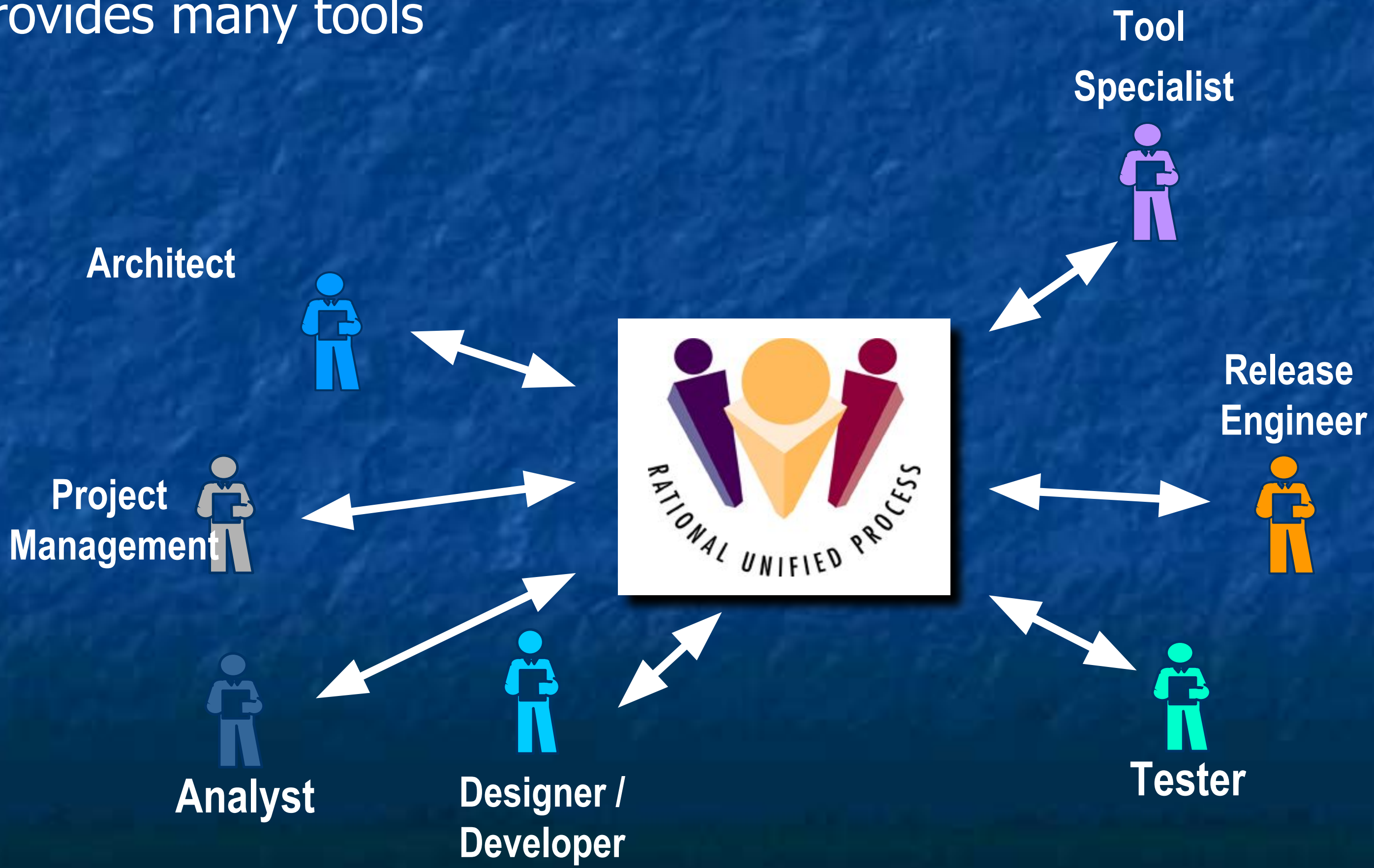
Rational Unified Process (RUP)

- Introduction
- Phases
- Core Workflows
- Best Practices
- Tools

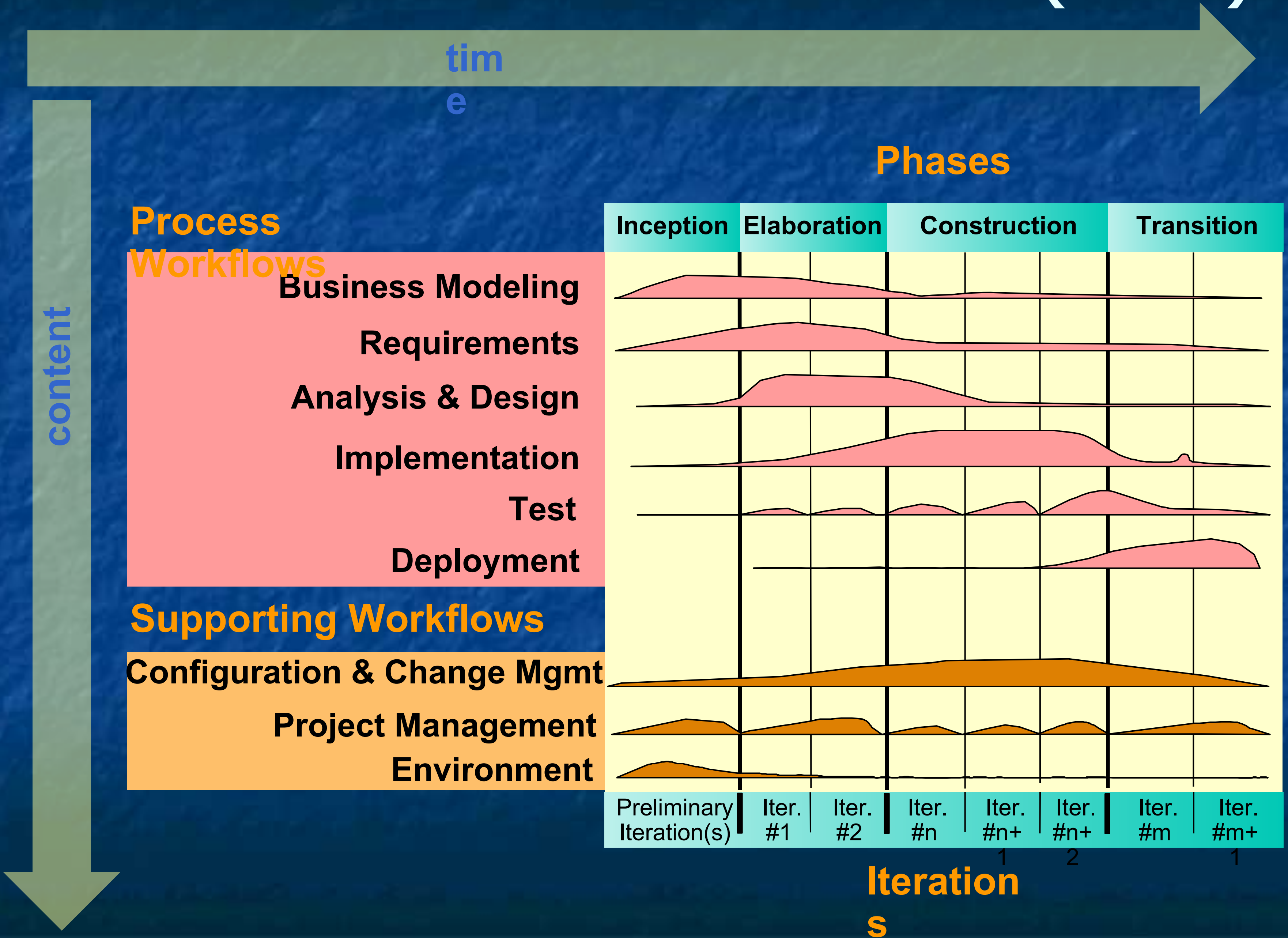
- **Team-Unifying Approach**

  The RUP unifies a software team by providing a common view of the development process and a shared vision of a common goal

- **Increased Team Productivity**
  - knowledge base of all processes
  - view of how to develop software
  - modeling language
  - Rational provides many tools



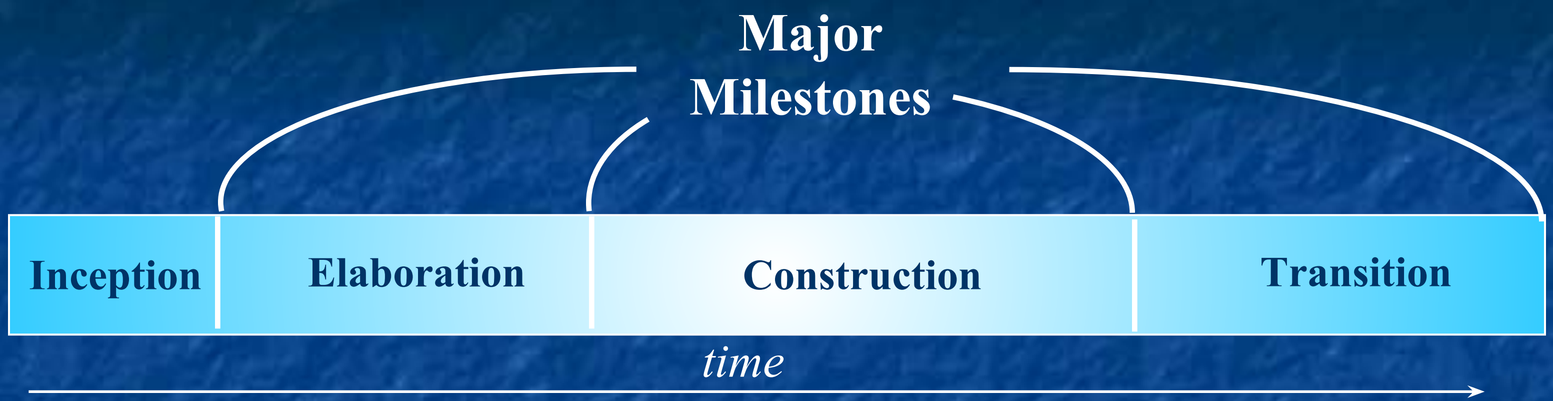
Rational Unified Process (RUP)



Rational Unified Process (RUP)

- Introduction
- Phases
- Core Workflows
- Best Practices
- Tools

Phases in the Process



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

Inception phase

- Establishing the project's software scope and boundary conditions, including an operational vision, acceptance criteria.
- Discriminating the critical use cases of the system.
- Estimating the overall cost and schedule for the entire project (and more detailed estimates for the elaboration phase that will immediately follow).
- Estimating potential risks (the sources of unpredictability)
- Preparing the supporting environment for the project.

Elaboration phase

- **Defining, validating .**
- **Refining the Vision, based on new information obtained during the phase.**
- **Refining the development case and putting in place the development environment, including the process & tools.**
- **Refining the architecture and selecting components.**
- **The selected architectural components are integrated and assessed against the primary scenarios.**

Construction phase

- Resource management, control and process optimization.
- Complete component development and testing against the defined evaluation criteria.
- Assessment of product releases against acceptance criteria for the vision.

Transition phase

- Executing deployment plans.
- Finalizing end-user support material.
- Testing the deliverable product at the development site.
- Creating a product release.
- Getting user feedback.
- Fine-tuning the product based on feedback.
- Making the product available to end users.

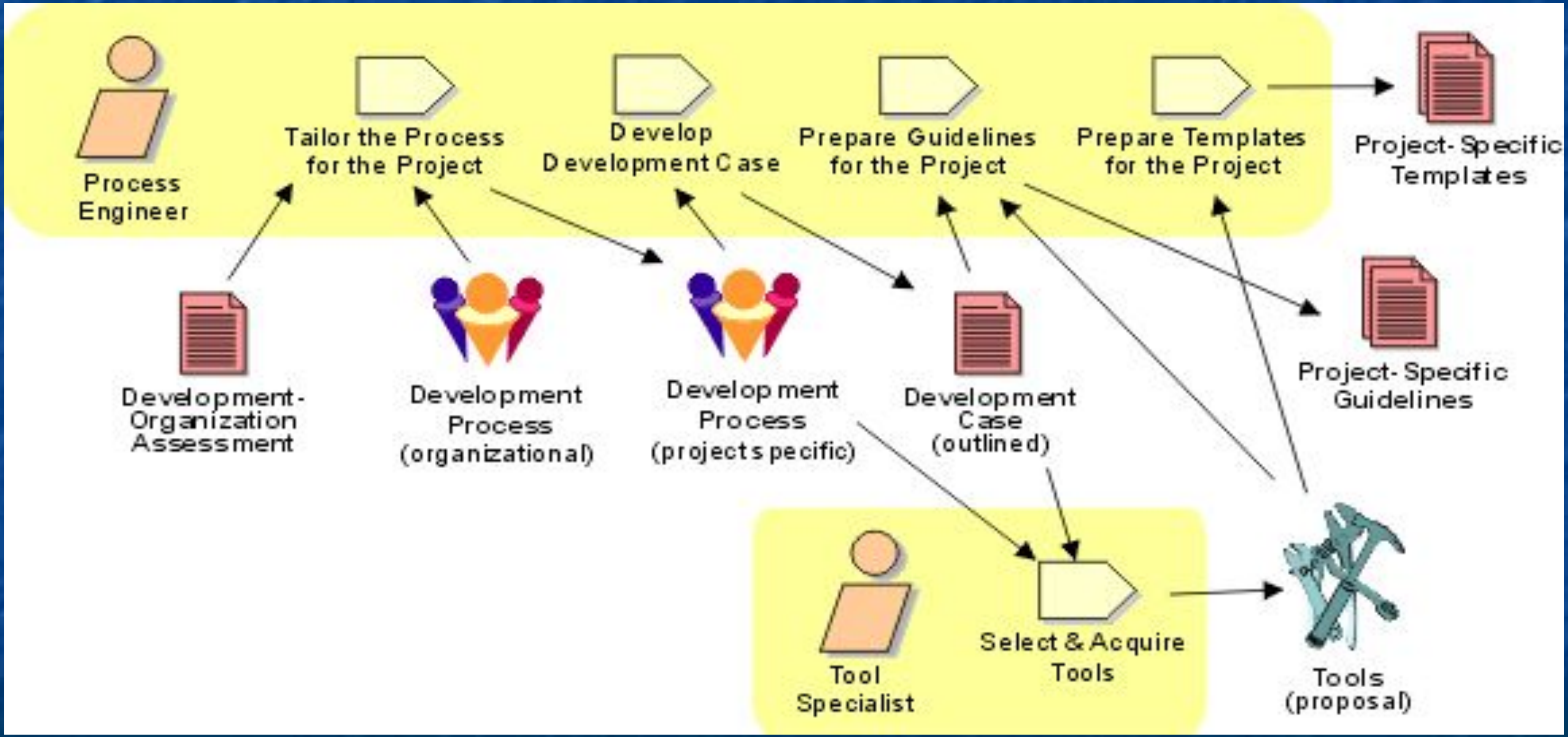
Rational Unified Process (RUP)

- Introduction
- Phases
- Core Workflows
- Best Practices
- Tools

What is a workflow?

- A set of activities that is performed by the various roles in a project
- Describes a meaningful sequence of activities that produce a useful result (an artifact)
- Shows interaction between roles

Workflow Detail: Prepare Environment for Project


Process Engineer
Tailor the Process for the Project
Develop Development Case
Prepare Guidelines for the Project
Prepare Templates for the Project
Project-Specific Templates
Development-Organization Assessment
Development Process (organizational)
Development Process (project specific)
Development Case (outlined)
Project-Specific Guidelines
Tool Specialist
Select & Acquire Tools
Tools (proposal)

Workflows - 3 key elements

- Three key elements of each workflows:
  - Artifacts
  - Roles
  - Activities

Artifacts

A piece of information that:

- Is produced, modified, or used by a process
- Defines an area of responsibility
- Is subject to version control.

An artifact can be a *model*, a *model element*, or a *document*. A document can enclose other documents.

Roles

- Represent a role that an individual may play on the project
- Responsible for producing artifacts
- Distinct from actors

Activities

- Tasks performed by people representing particular roles in order to produce artifacts

Brief summary of process workflows

- Business Modelling
- Requirements
- Analysis & Design
- Implementation
- Test
- Deployment

Business Modelling

- Understand structure & dynamics of organization in which system is to be deployed
- Understand current problems in the target organization & identify improvement potential
- Ensure customers, end users & developers have common understanding of target organisation
- Derive system requirements to support target organisation

Analysis & Design

- Transform requirements into a design of the system
- Evolve a robust architecture for the system
- Adapt design to match the implementation environment, designing it for performance

Implementation

- Define organization of the code, in terms of implementation subsystems organized in layers
- Implement classes & objects in terms of components
- Test developed components as units
- Integrate results into an executable system

Test

- Verify interaction between objects
- Verify proper integration of all components of the software
- Verify that all requirements have been correctly implemented
- Identify & ensure defects are addressed prior to deployment

Deployment

- Provide custom installation
- Provide shrink wrap product offering
- Provide software over internet

Brief summary of supporting workflows

- Configuration & Change Management
- Project Management
- Environment

Configuration & Change Management

- Supports development methods
- Maintains product integrity
- Ensures completeness & correctness of configured product
- Provides stable environment within which to develop product
- Restricts changes to artifacts based on project policies
- Provides an audit trail on why, when & by whom any artifact was changed

Project Management

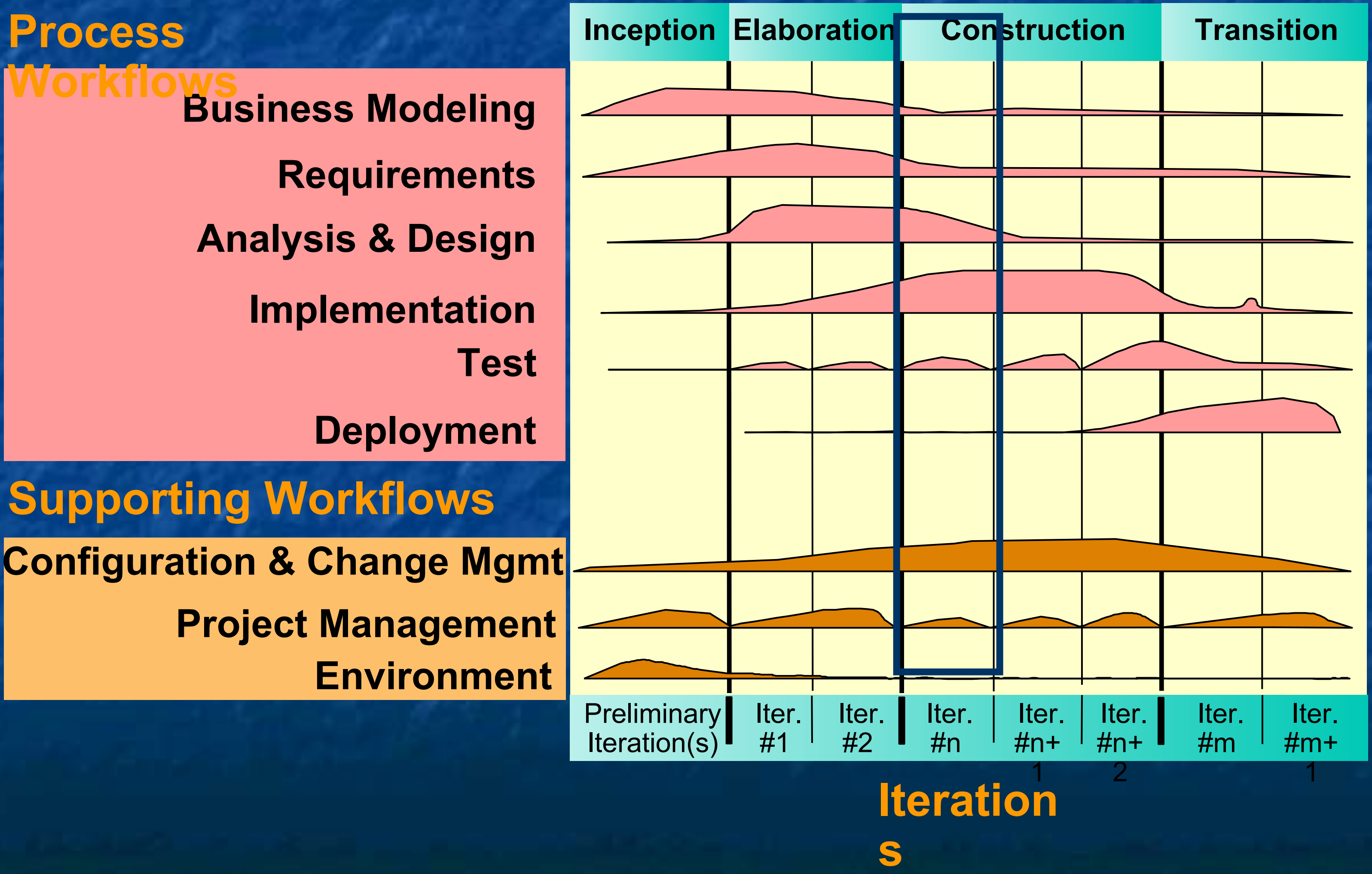
- A framework for managing software-intensive projects
- Practical guidelines for planning, staffing, executing & monitoring projects
- A framework for managing risk

Environment

- Design, implement and manage the project’s required technical environments
- Define the technical architectures for the development, system validation, testing & staging/release management environments
- When possible, standard architectural models for given types of platforms should be utilized when defining the production environment

Bringing It All Together...

In an iteration, you walk through all workflows

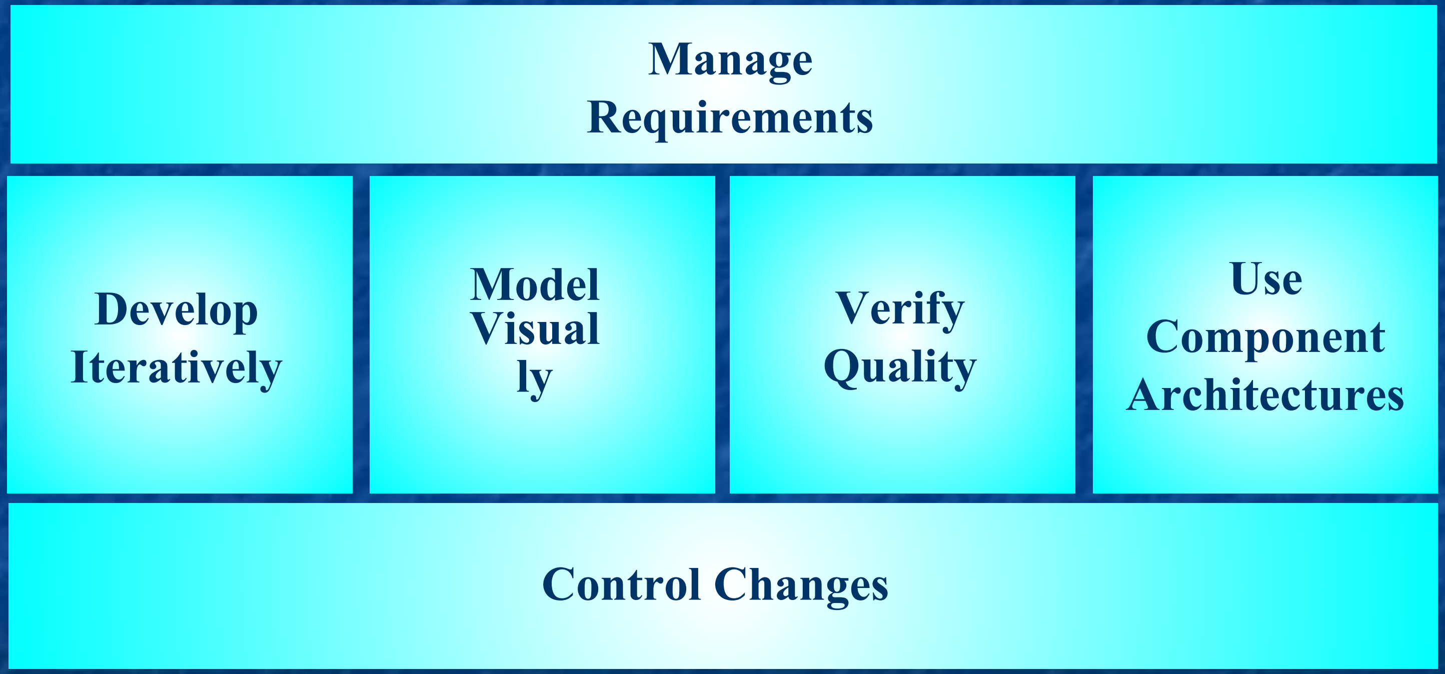


Rational Unified Process (RUP)

- Introduction
- Phases
- Core Workflows
- Best Practices
- Tools

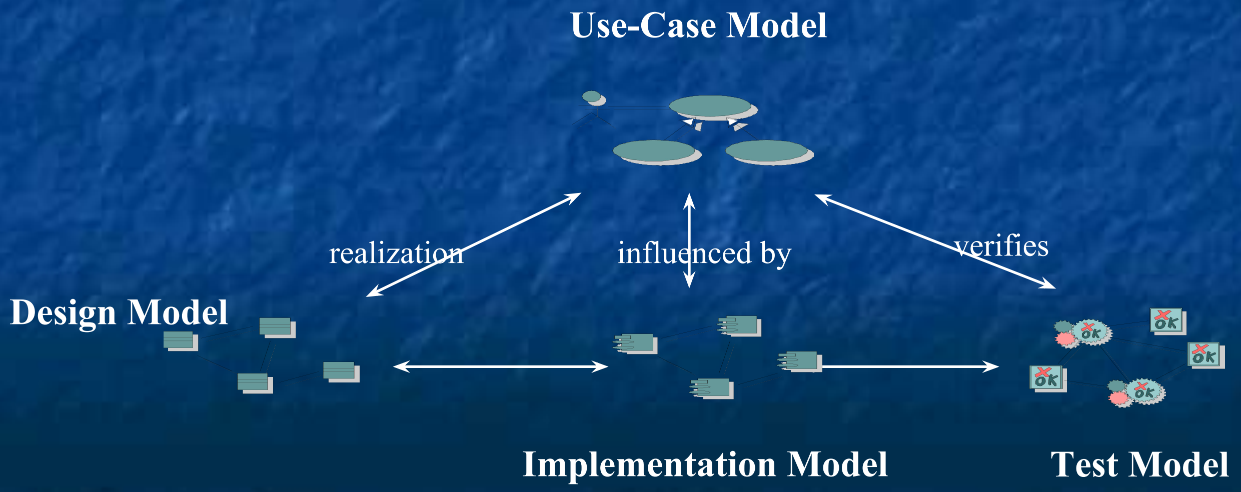
Rational Unified Process

Describes the effective implementation of key "Best Practices"



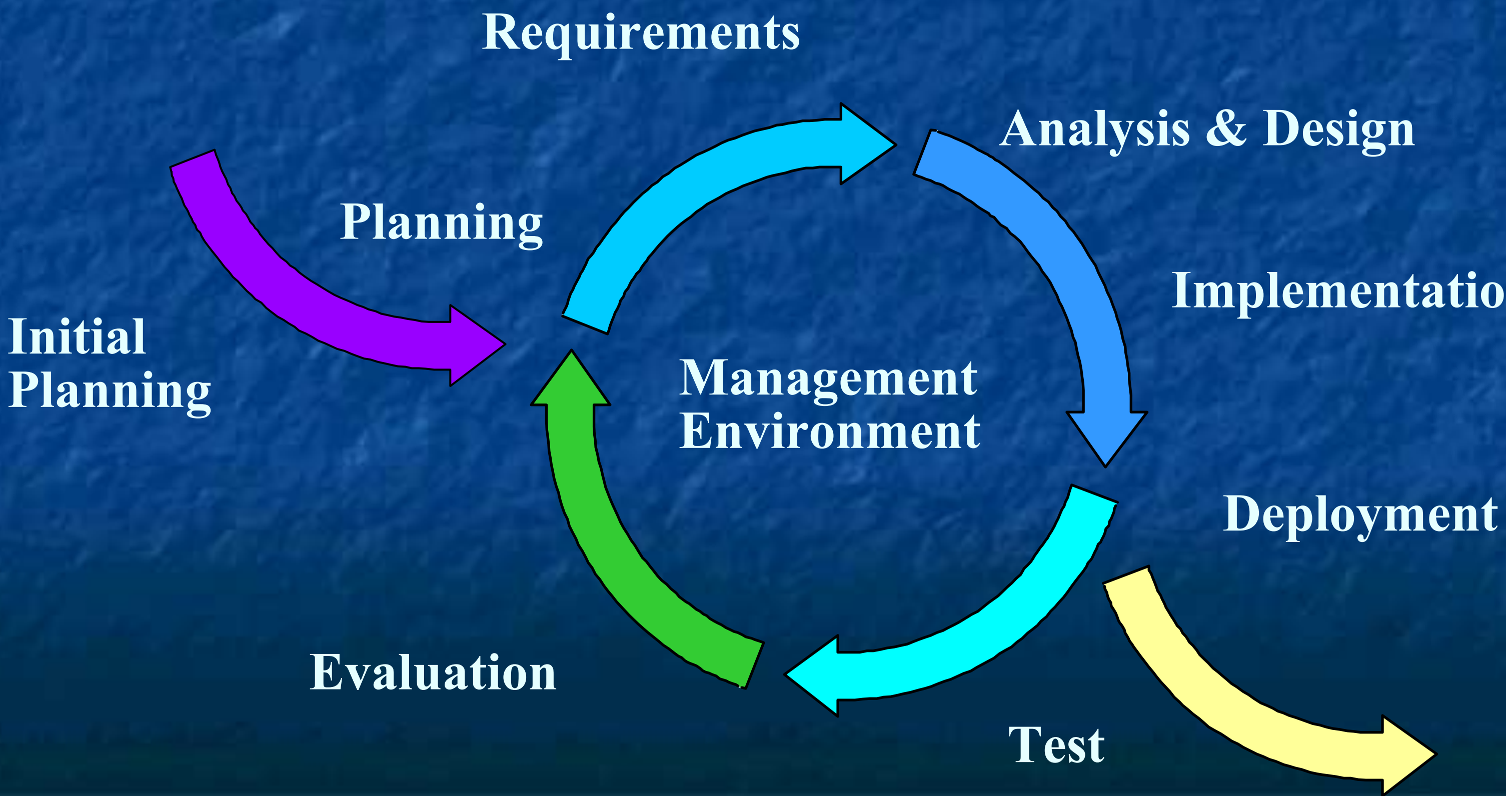
1. Manage Your Requirements

- Elicit, organize, and document required functionality and constraints
- Track and document tradeoffs and decisions
- Business requirements are easily captured and communicated through use cases
- Use cases are important planning instruments

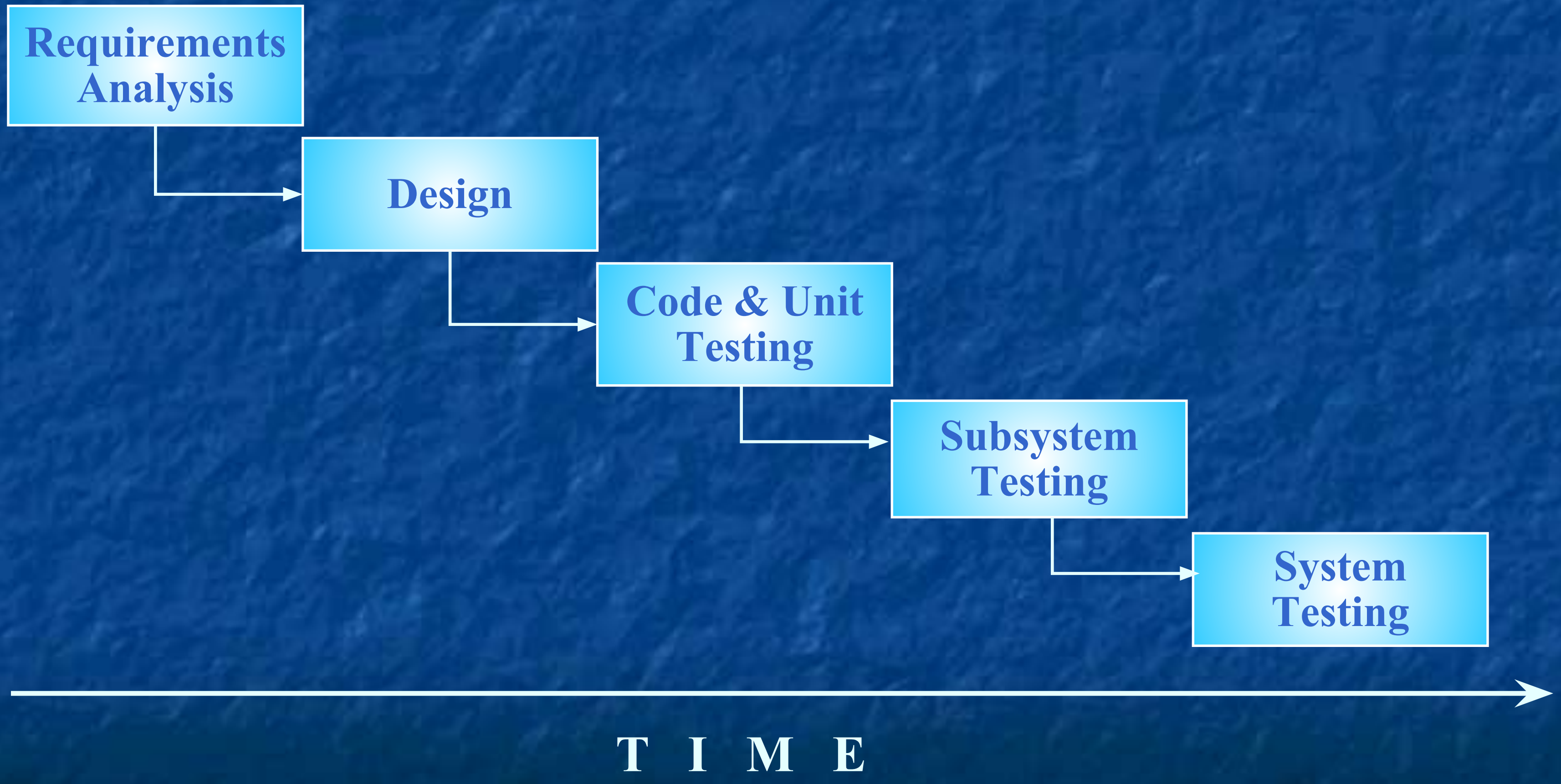


2. Develop Software Iteratively

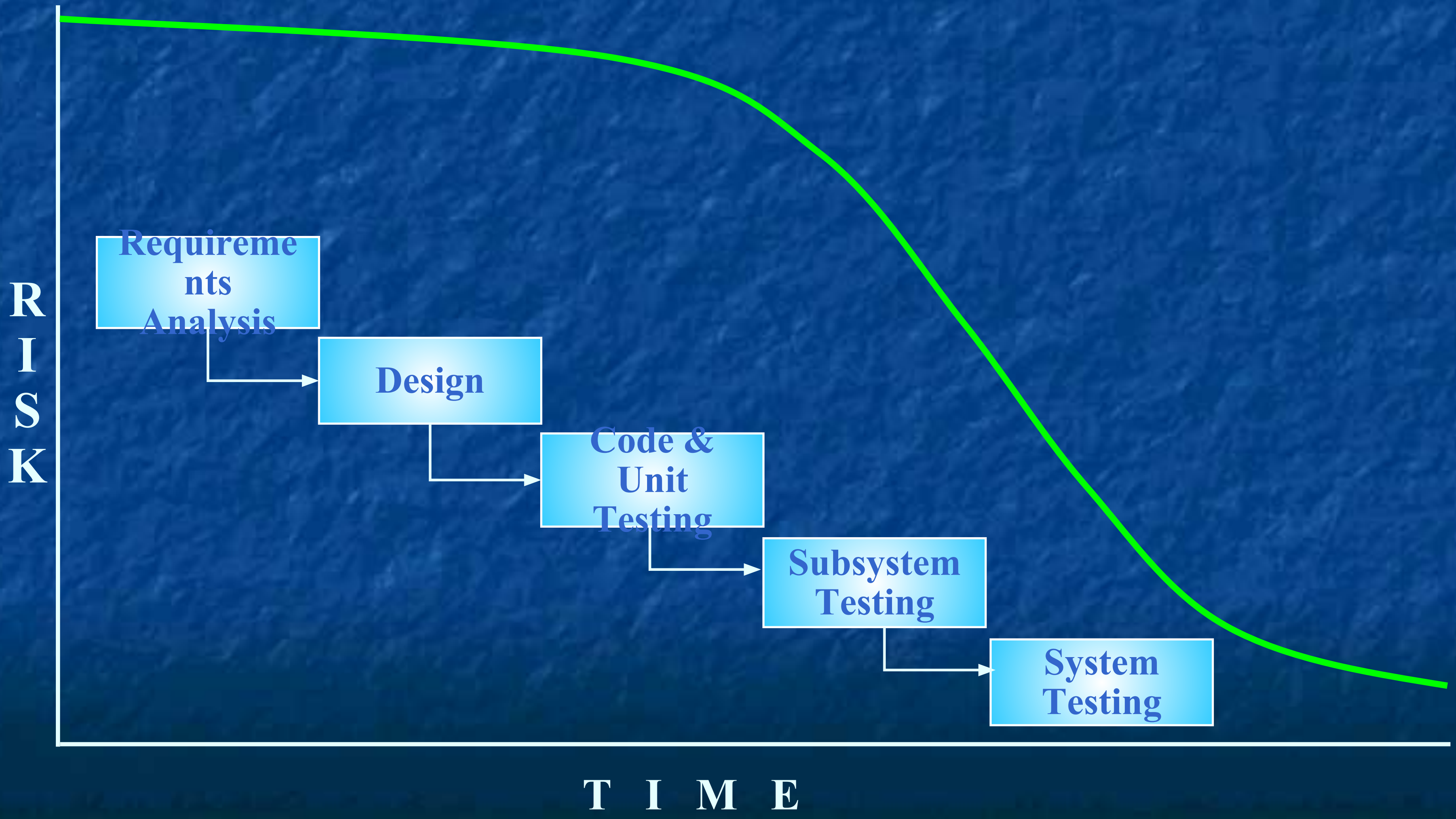
- An initial design will likely be flawed with respect to its key requirements
- Late-phase discovery of design defects results in costly over-runs and/or project cancellation



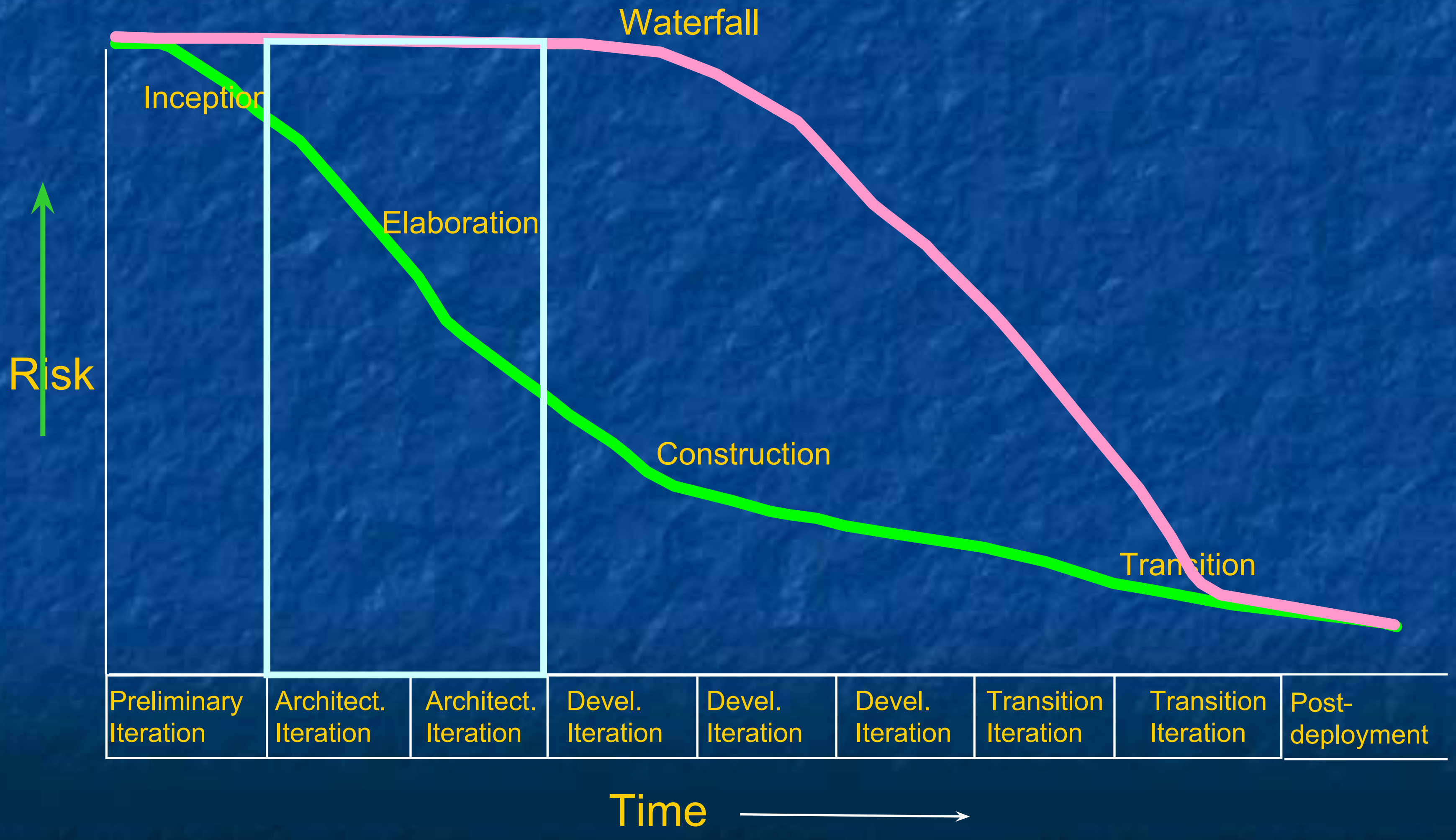
Waterfall Development



Waterfall Development: Risk vs. Time



Risk Profile of an Iterative Development



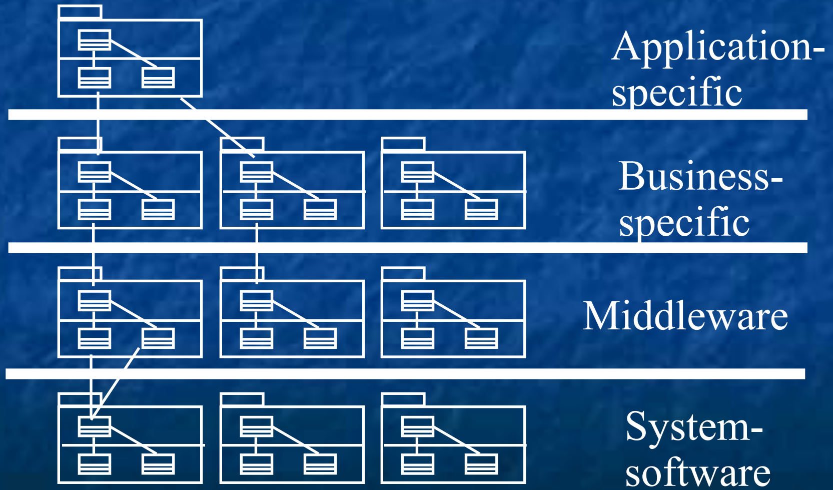
Iterative Development Characteristics

- Critical risks are resolved before making large investments
- Initial iterations enable early user feedback
- Testing and integration are continuous
- Objective milestones provide short-term focus
- Progress is measured by assessing implementations
- Partial implementations can be deployed

3. Employ Component-based Architecture

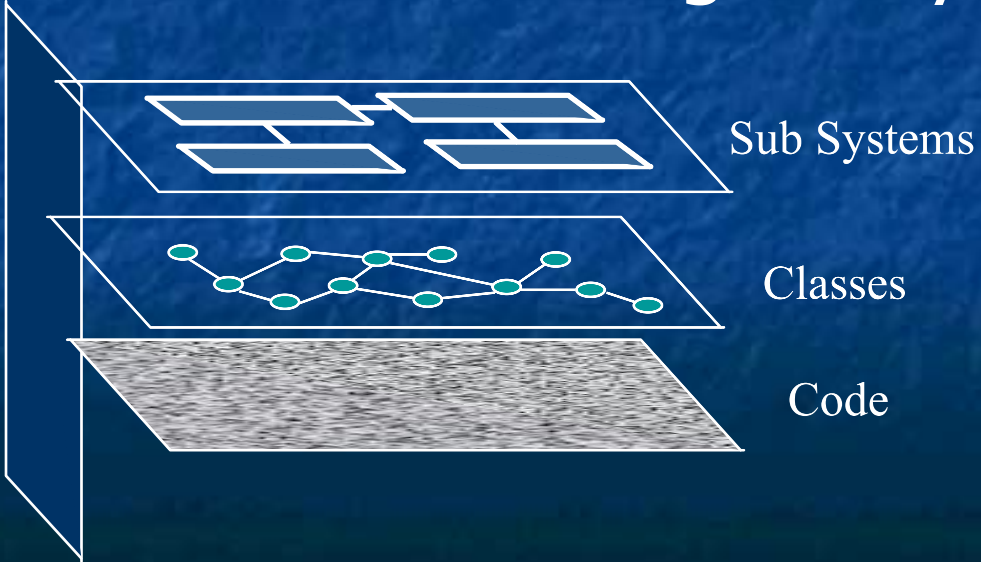
- Design, implement and test your architecture up-front!
- A systematic approach to define a "good" architecture
  - **Resilient to change by using well-defined interfaces**
  - **By using and reverse engineering components**
  - **Derived from top rank use cases**

Component-based Architecture with layers



4. Model Software Visually

- Aiding understanding of complex systems
- Exploring and comparing design alternatives at a low cost
- Forming a foundation for implementation
- Capturing requirements precisely
- Communicating decisions unambiguously

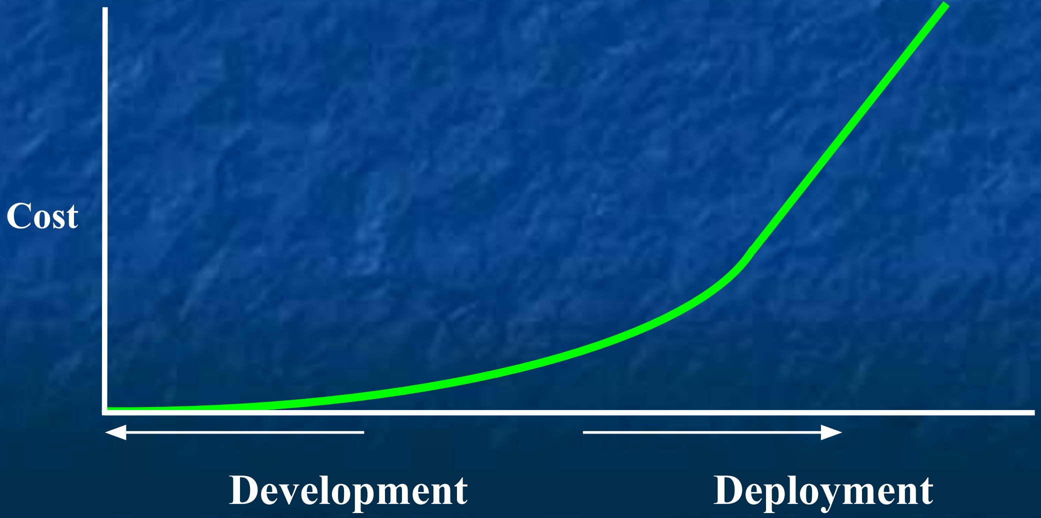


Visual Modeling raises the level of abstraction

5. Verify Software Quality

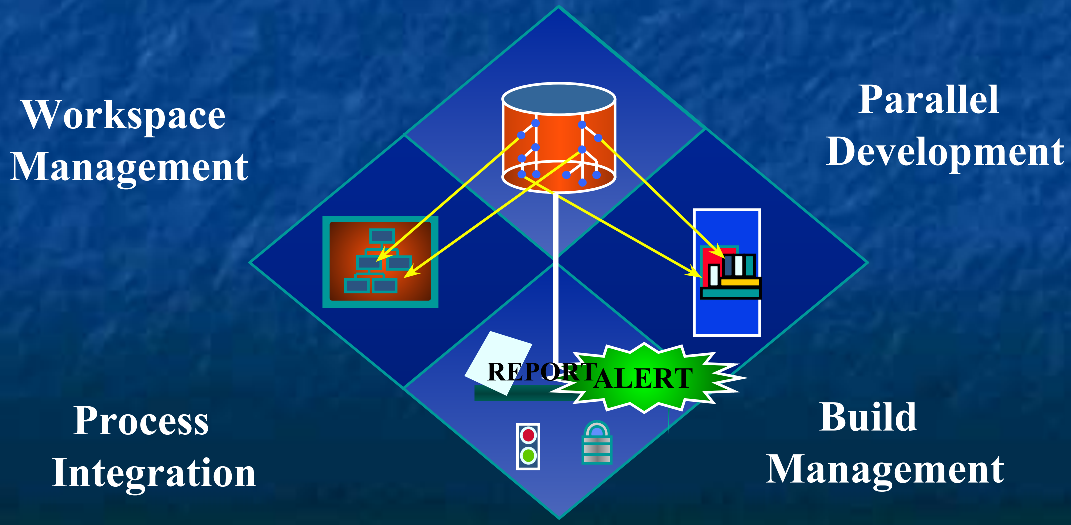
- Create tests for each key scenario to ensure that all requirements are properly implemented
- Unacceptable application performance hurts as much as unacceptable reliability
- Verify software reliability - memory leaks, bottle necks
- Test every iteration - automate test!

Software problems are 100 to 1000 times more costly to find and repair after deployment



6. Control Changes to Software

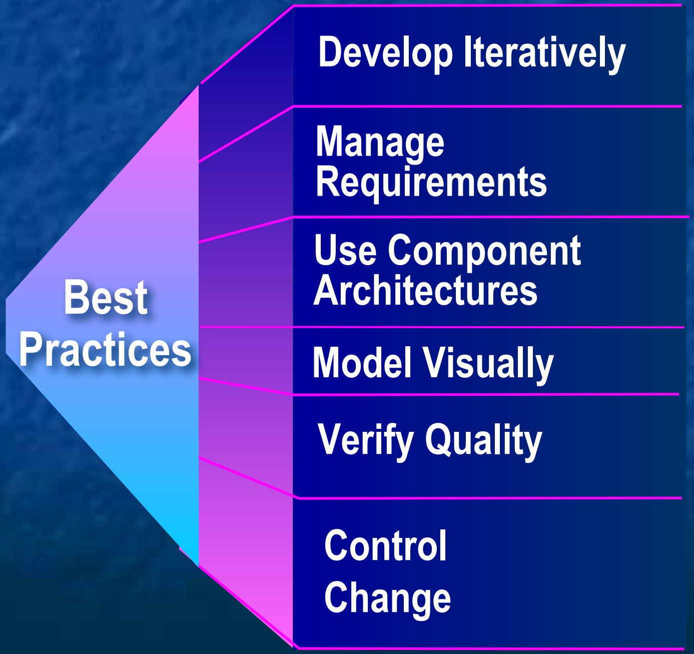
- Control, track and monitor changes to enable iterative development
- Establish secure workspaces for each developer
  - Provide isolation from changes made in other workspaces
  - Control all software artifacts - models, code, docs, etc.
- Automate integration and build management

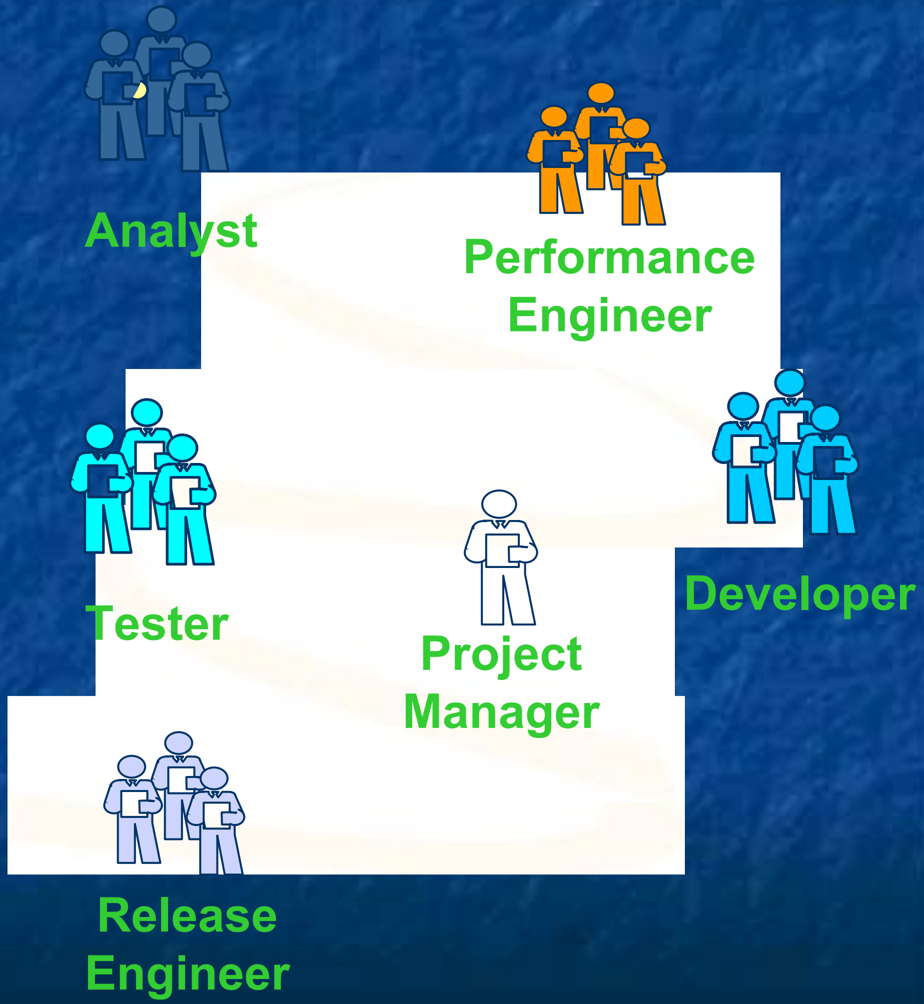


CM is more than just check-in and check-out

Summary: Best Practices of Software Engineering

- The result is software that is
  - On Time
  - On Budget
  - Meets Users Needs





Rational Unified Process (RUP)

- Introduction
- Phases
- Core Workflows
- Best Practices
- Tools

Tools

- The success of process adoption is significantly improved by the use of appropriate supporting tools.
- **Tool Mentors** provide detailed descriptions of how to perform specific process activities or steps, or produce a particular artifact or report, using one or more tools.

Tools

- Rational Unified Process
- RUP Builder
- Rational Process Workbench
- Rational Administrator
- Rational Suite AnalystStudio
- Rational ClearCase
- Rational ClearQuest
- Rational ProjectConsole
- Rational PurifyPlus
- Rational QualityArchitect

Tools

- Rational RequisitePro
- Rational Robot
- Rational Rose
- Rational Rose RealTime
- Rational SoDA
- Rational TestManager
- Rational Test RealTime
- Rational TestFactory
- Rational XDE Developer - Java Platform Edition
- Rational XDE Developer - .NET Edition