Introduction of RUP - The Rational Unified Process

Jong-Hoon Lee

Dependable Software Laboratory

Konkuk University
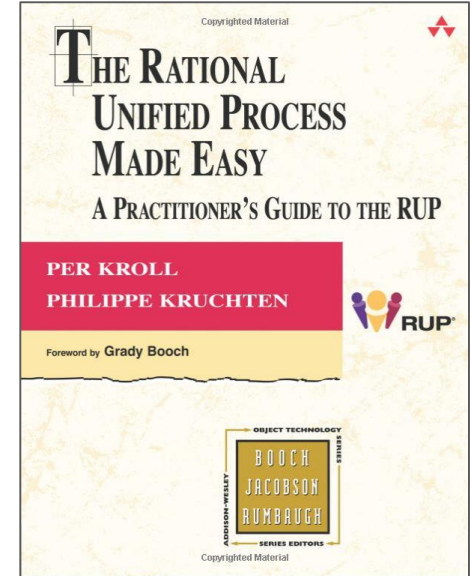
References

Textbook:

The Rational Unified Process Made Easy

A Practitioner's Guide to the RUP

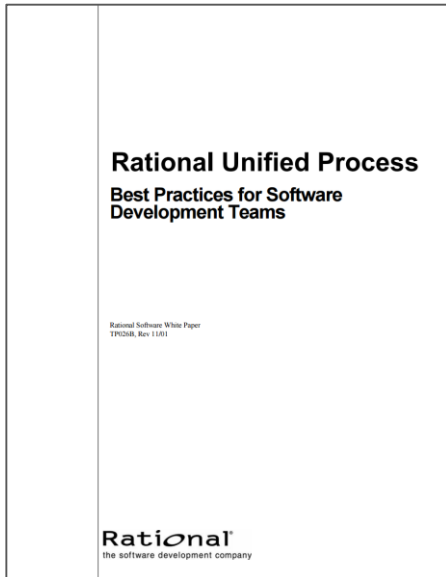- Per Kroll, Philippe Kruchten, Grady Booch

White Paper:

Rational Unified Process

Best Practices for Software Development Teams

- TP026B 11/01, IBM

Contents

What is the RUP (Rational Unified Process)?

The Inception Phase

The Elaboration Phase

The Construction Phase

The Transition Phase

What is the RUP?
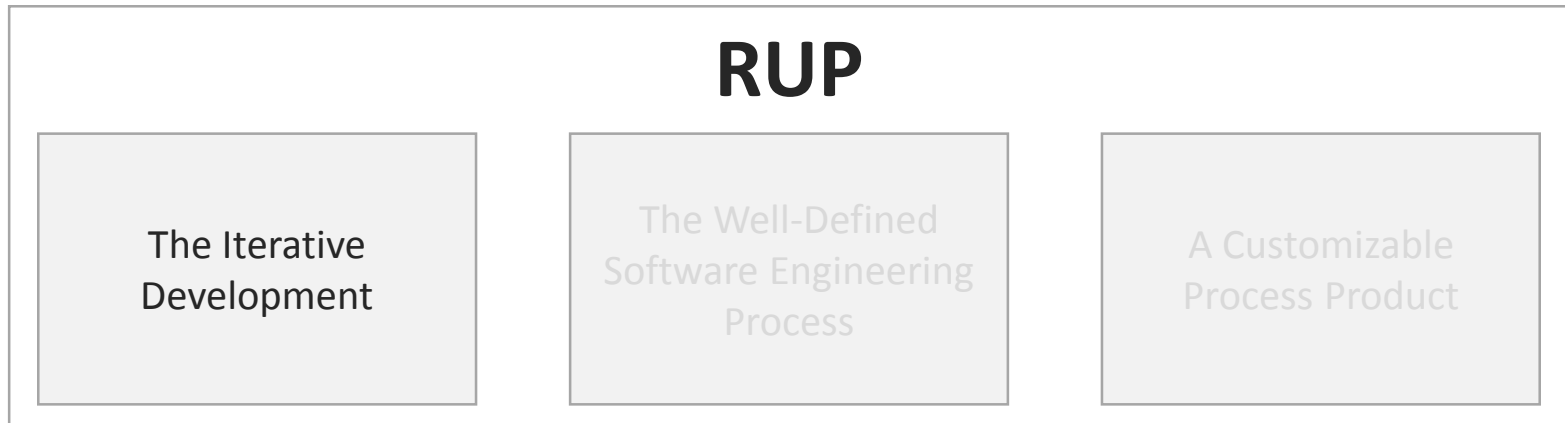
What is the RUP?

The RUP is

- A **software development approach** that is iterative, architecture-centric, and use-case driven

- A **well-defined and well-structured software engineering process**

- A **process product** that provides you with a **customizable process framework** for software engineering
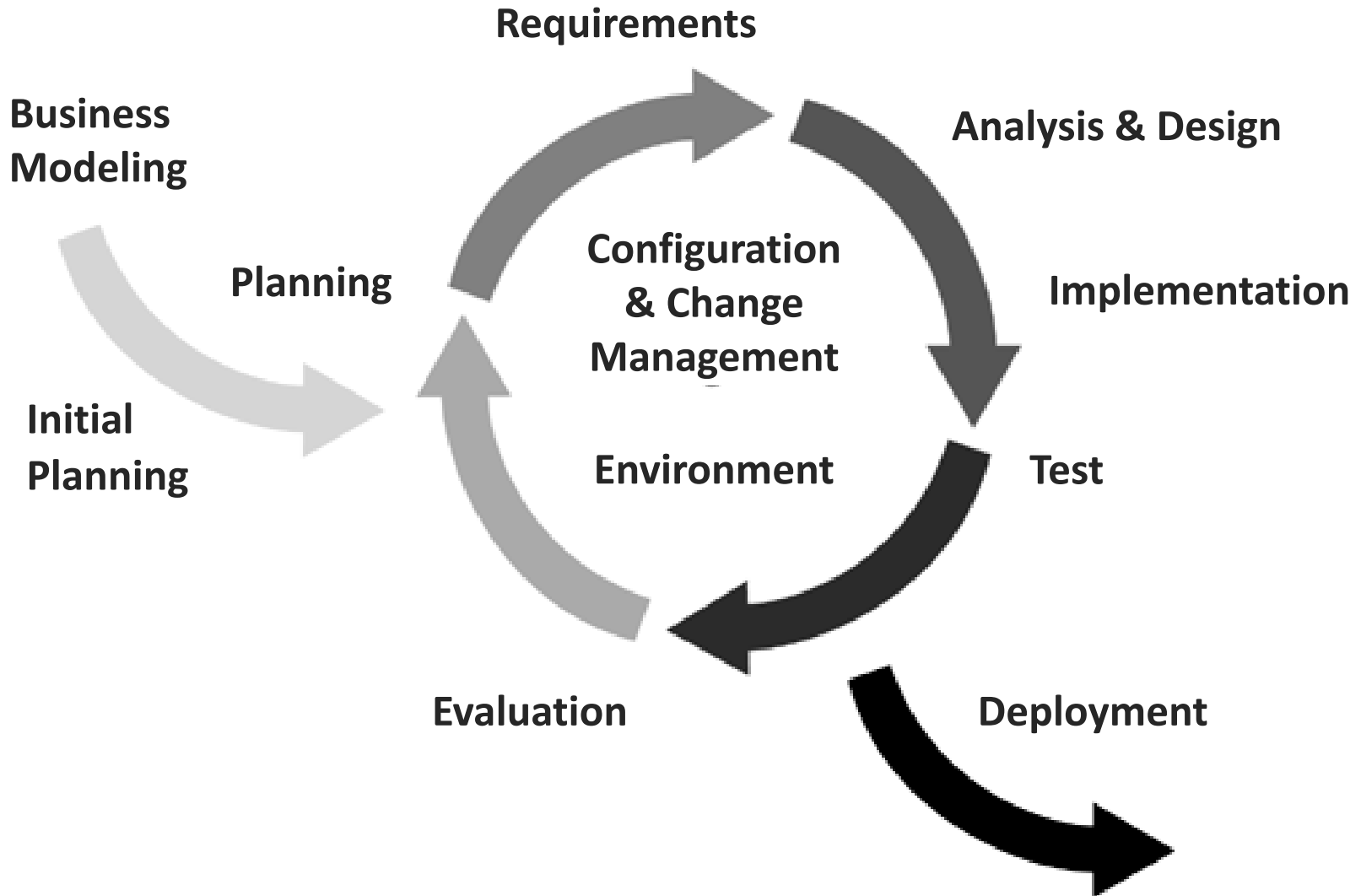
RUP		
The Iterative Development	The Well-Defined Software Engineering Process	A Customizable Process Product

What is the RUP?

The RUP uses iterative approach

- The RUP uses an iterative approach that is a sequence of incremental steps or iterations
  - It accommodates changing requirements
  - Integration is not one "big bang" at the end of a project
  - Risks are usually discovered or addressed during early integrations
  - Management has a means of making tactical changes to the product
  - Defects can be found and corrected over several iterations
  - It is a better use of project personnel
  - Team members learn along the way
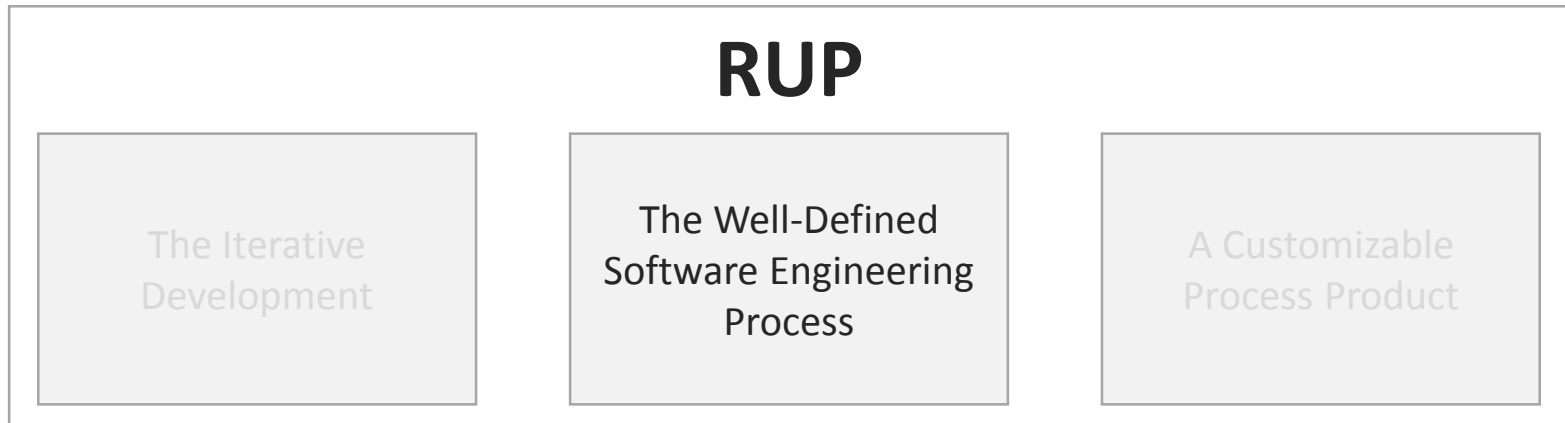 - The development process itself is improved and refined along the way

RUP

The Iterative Development	The Well-Defined Software Engineering Process	A Customizable Process Product

What is the RUP?

Requirements

Business Modeling

Analysis & Design

Planning

Implementation

Configuration & Change Management

Initial Planning

Environment

Test

Evaluation

Deployment

What is the RUP?

The RUP is a well-defined software engineering process

- Modeling
    - It creates and maintain **models**, and effectively **use the UML** (Unified Modeling Language)
    - It comprised of two structures or dimensions: **Dynamic** and **Static**
- Dynamic Structure
    - It deals with the lifecycle or time dimension of a project
    - It comprised of Inception, Elaboration Construction and Transition phases
    - Dynamic aspects: cycles, phases, iterations, and milestones
- Static Structure
    - It describes how process elements are logically grouped into workflows
   - Static aspects: activities, disciplines, artifacts, and roles

RUP

The Iterative Development	The Well-Defined Software Engineering Process	A Customizable Process Product

KU KONKUK UNIVERSITY DEPENDABLE SOFTWARE LABORATORY

What is the RUP?

Dynamic Structure
- Phases
  - Inception
  - Elaboration
  - Construction
  - Transition
- Iterations
- Etc.

Static Structure
- Disciplines
  - Business modeling
  - Requirements
  - Analysis & Design
  - Etc.
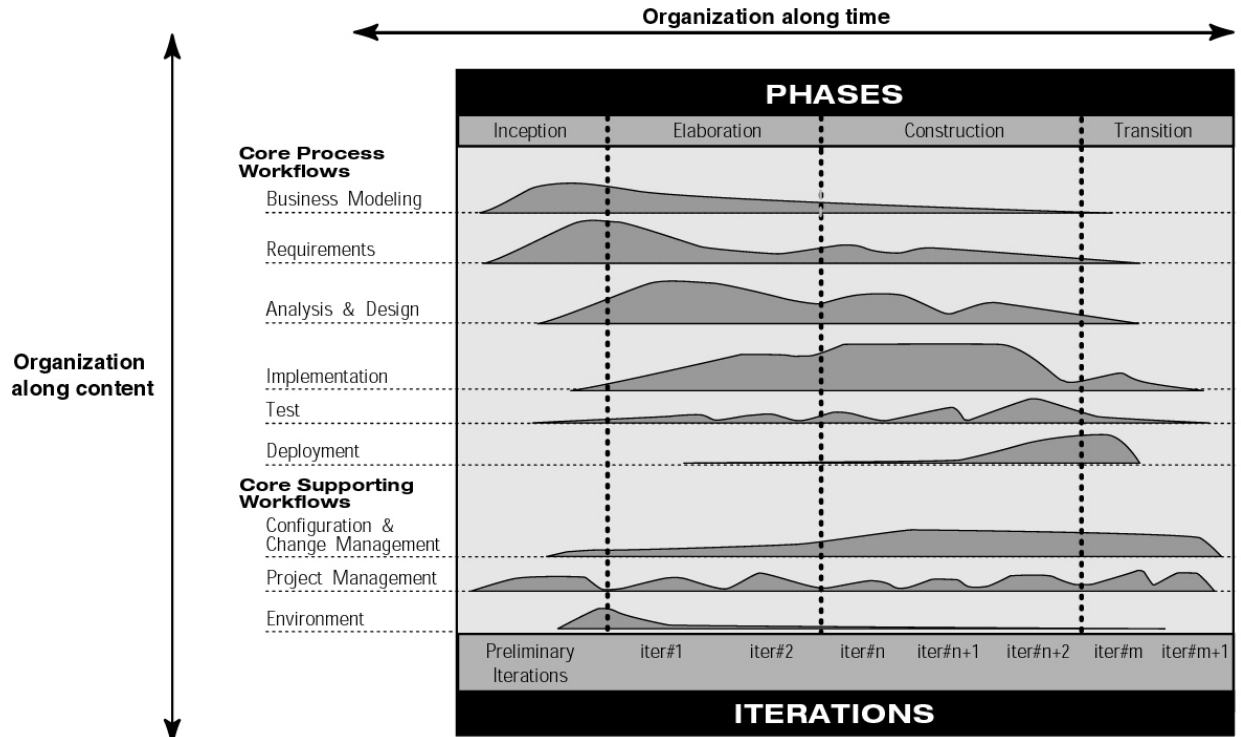
What is the RUP?

**Dynamic Structure: The lifecycle of a RUP project going through:**

- The Inception Phase
  - Establish a good understanding of what system to build

- The Elaboration Phase
  - Take care of many of the most technically difficult tasks, and address major technical risks

- The Construction Phase
  - Do most of the implementation as you move from an executable architecture to the first operational version of your system

- The Transition Phase
 - Ensure that software addresses the needs of its users
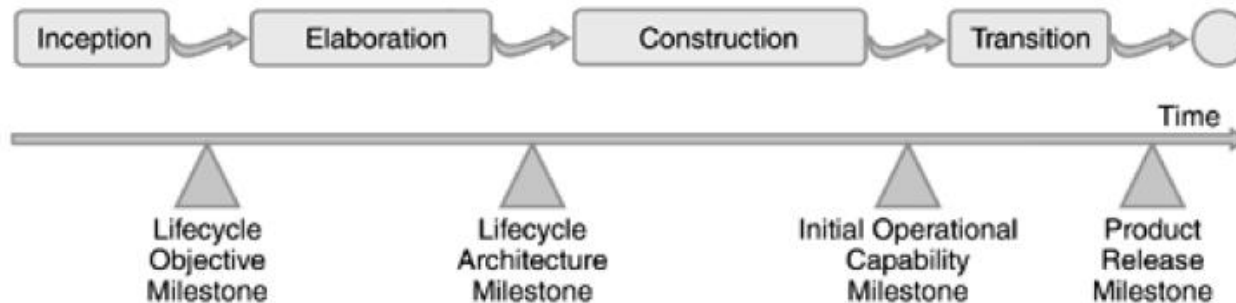
What is the RUP?

**Static Structure: The process describes who is doing what, how, and when**

- Roles – The *who*
  - It is like a 'hat' that an individual (or group) wears during a project

- Activities – The *how*
  - It is a unit of work that an individual in that role may be asked to perform

- Artifacts – The *what*
  - It is a piece of information that is produced, modified, or used by a process

- Workflows – The *when*
  - It is a process to describe meaningful sequence of activities that produce some valuable result and show interactions between roles
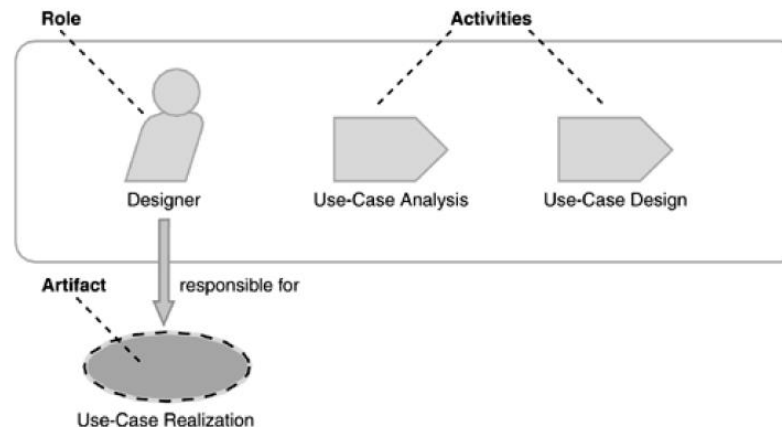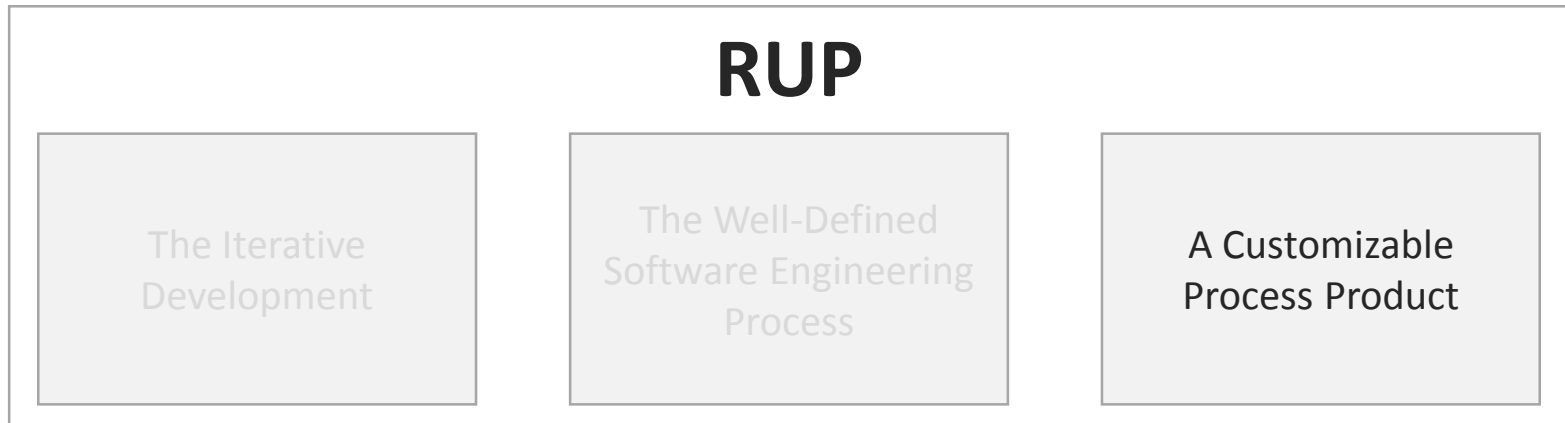
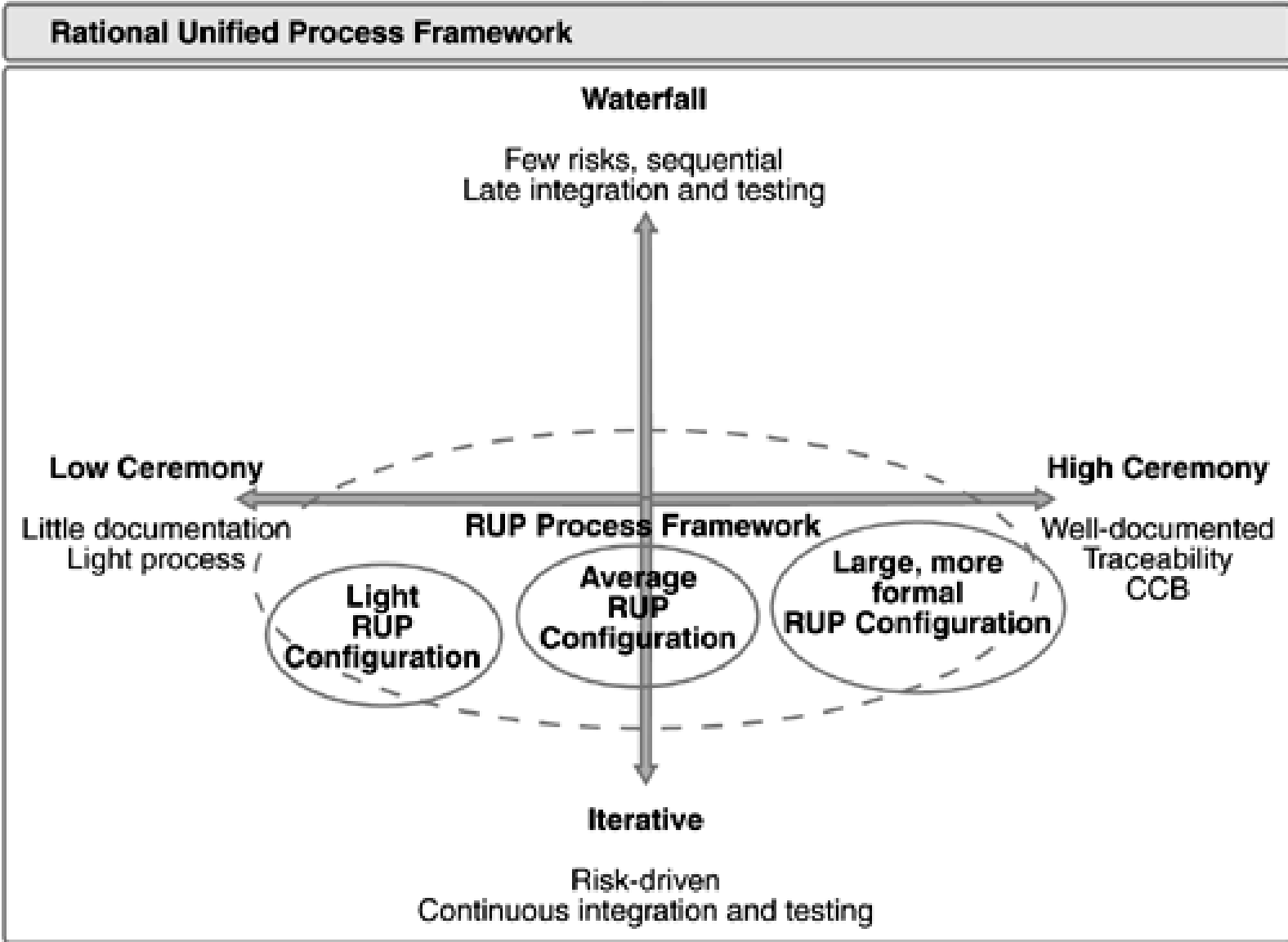# What is the RUP?
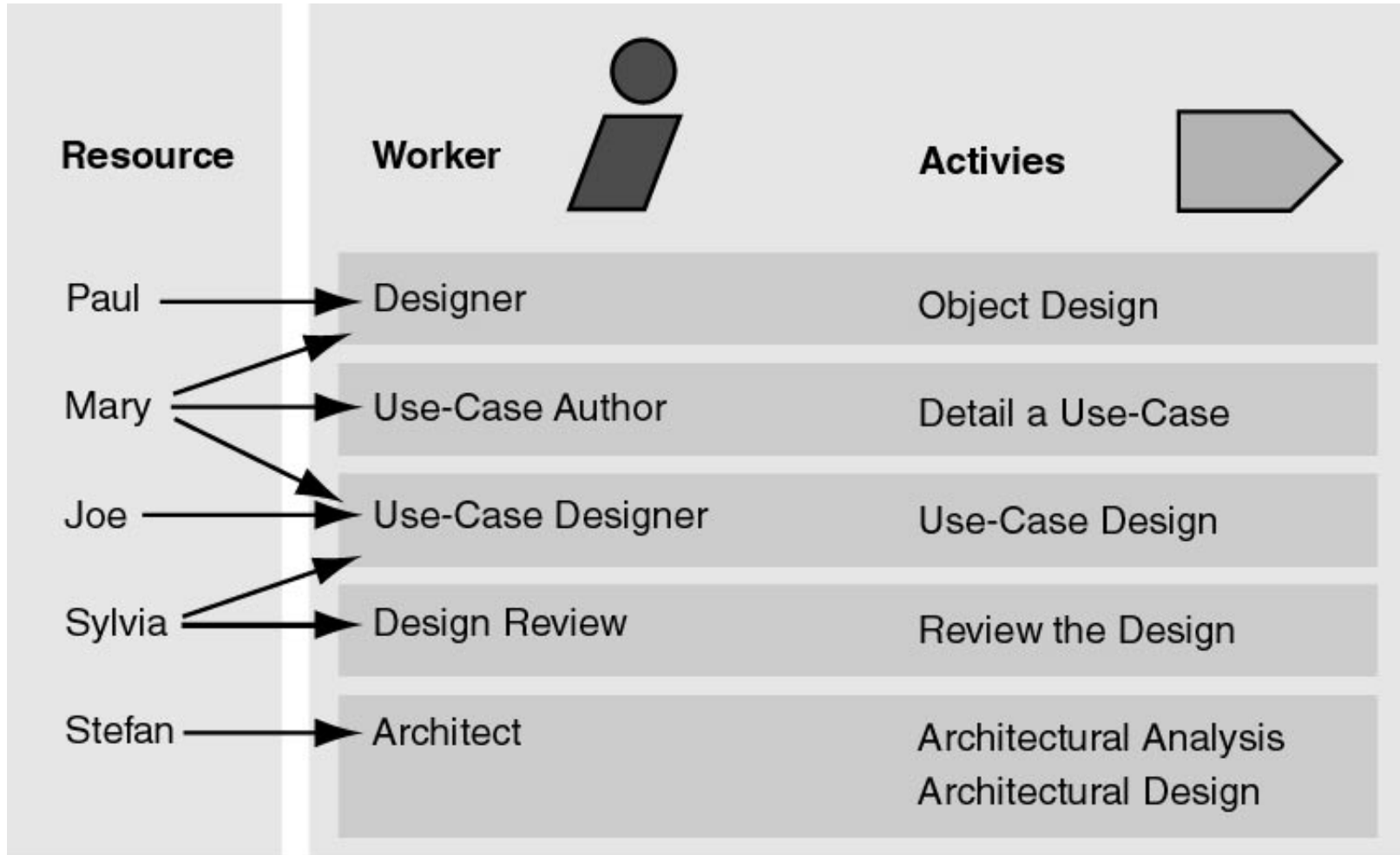
What is the RUP?
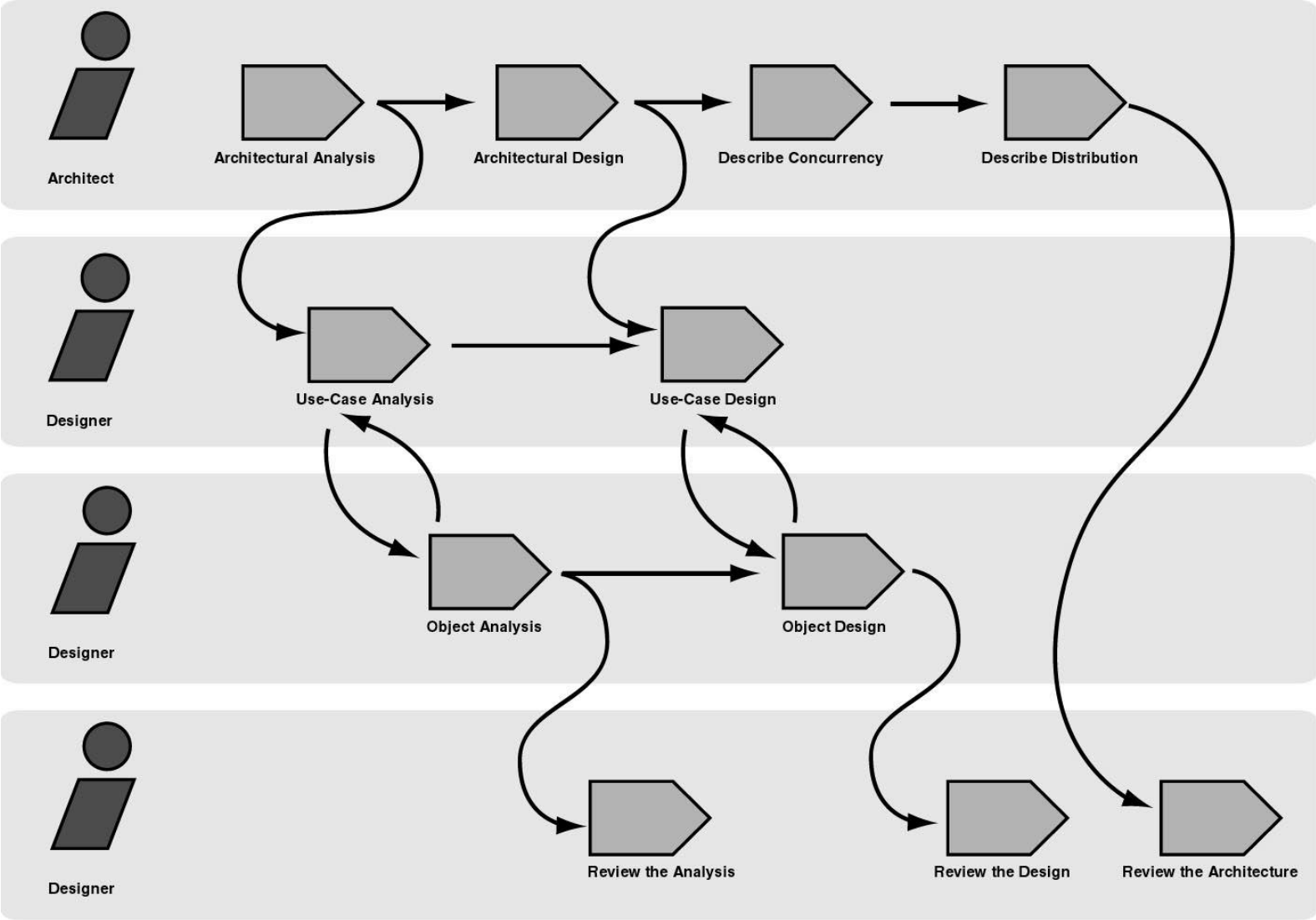
What is the RUP?

The RUP is a customizable process product

- You can configure the RUP product to support small or large teams and disciplined or less formal approaches to development
    - Best practices
    - Process delivery tools
    - Configuration tools
    - Process authoring tools
    - Community/Marketplace

RUP		
The Iterative Development	The Well-Defined Software Engineering Process	A Customizable Process Product

What is the RUP?

Rational Unified Process Framework

Waterfall
Few risks, sequential
Late integration and testing

Low Ceremony
Little documentation
Light process

High Ceremony
Well-documented
Traceability
CCB

RUP Process Framework

Light RUP Configuration

Average RUP Configuration

Large, more formal RUP Configuration

Iterative
Risk-driven
Continuous integration and testing

What is the RUP?

Essential principles of the RUP:

- Attack major risks early and continuously… or they will attack you

- Ensure that you deliver value to your customer

- Stay focused on executable software

- Accommodate change early in the project

- Baseline an executable architecture early on

- Build your system with components

- Work together as one team

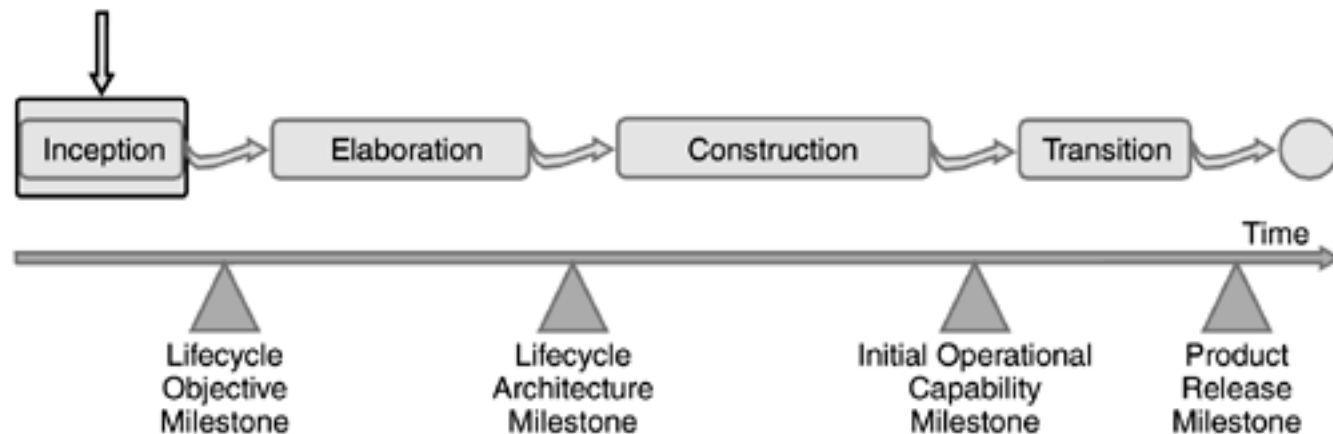- Make quality a way of life, not an afterthought

The Inception Phase

The Inception Phase

Objectives of the inception phase:

- Understand what to build*

- Identify key system functionality*

- Determine at least one possible solution*

- Understand the costs, schedule, and risks associated with the project

- Decide what process to follow and what tools to use

The Inception Phase

Objective 1: Understand what to build

- Produce a vision document

- Generate a "Mile-wide, inch-deep" description
  - Identify and briefly describes actors
  - Identify and briefly describe use cases
  - Identify and describe how each actor will interact with the system

- Hold a workshop or brainstorming session

- Detail key actors and use cases

The Inception Phase

Objective 2: Identify key system functionality

- The functionality is the core of the application, or it exercises key interfaces of the system

- The functionality **must** be delivered

- The functionality covers an area of the architecture that is not covered by any other critical use cases

The Inception Phase

Objective 3: Determine at least one possible solution

- Make sure that there is at least one potential architecture that will allow you to build the system with a sensible amount of risk and at reasonable cost

Thinner client, thicker server
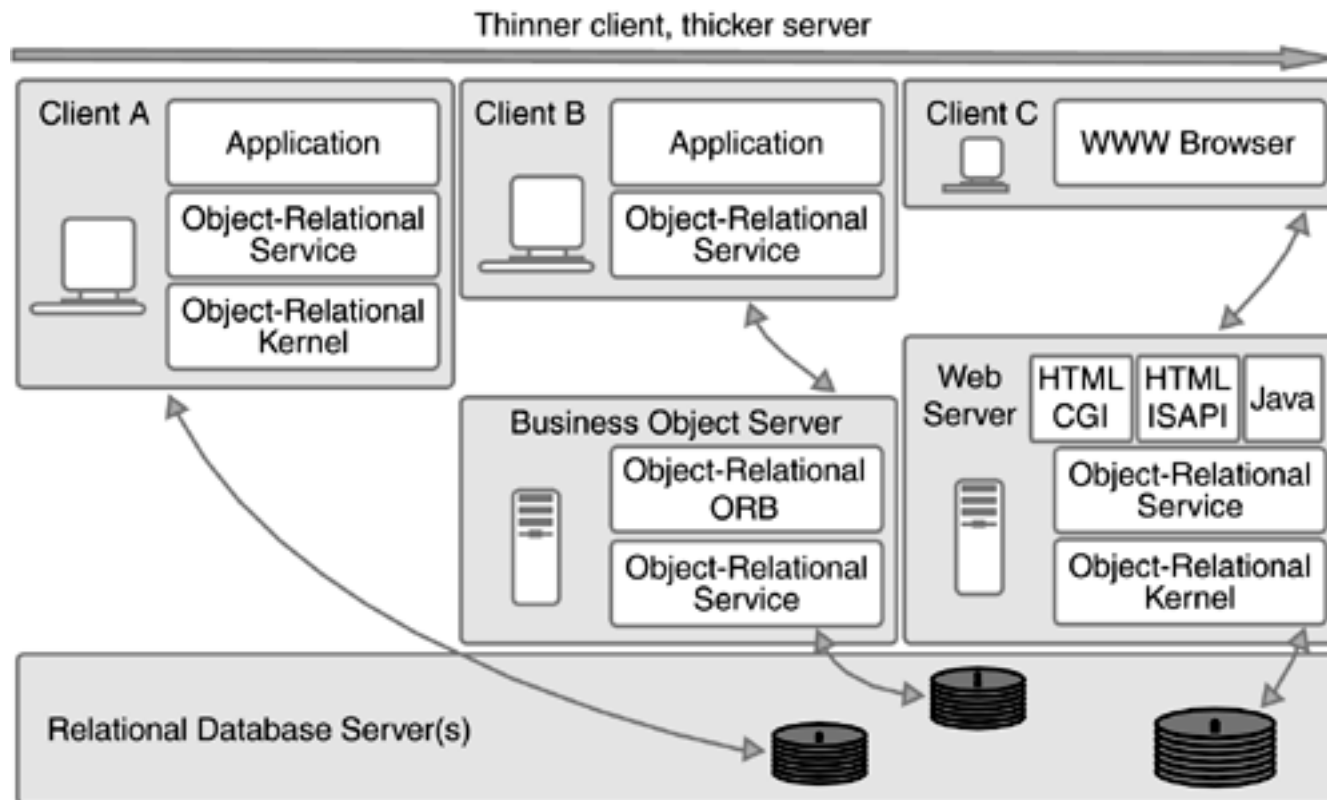
The Inception Phase

The outcome of the inception phase:

- A vision document

- A initial use-case model (10 – 20%) complete

- An initial project glossary

- An initial business case, which includes business context, success criteria, and financial forecast

- An initial risk assessment

- A project plan, showing phases and iterations

- A business model, if necessary

- One or several prototypes

The Inception Phase

Lifecycle objective milestone

- Stakeholder concurrence on scope definition and cost/schedule estimates

- Requirements understanding as evidenced by the fidelity of the primary use cases

- Credibility of the cost/schedule estimates, priorities, risks, and development process

- Depth and breadth of any architectural prototype that was developed

- Actual expenditures versus planned expenditures

**The project may be cancelled or considerably re-thought**

The Elaboration Phase

The Elaboration Phase
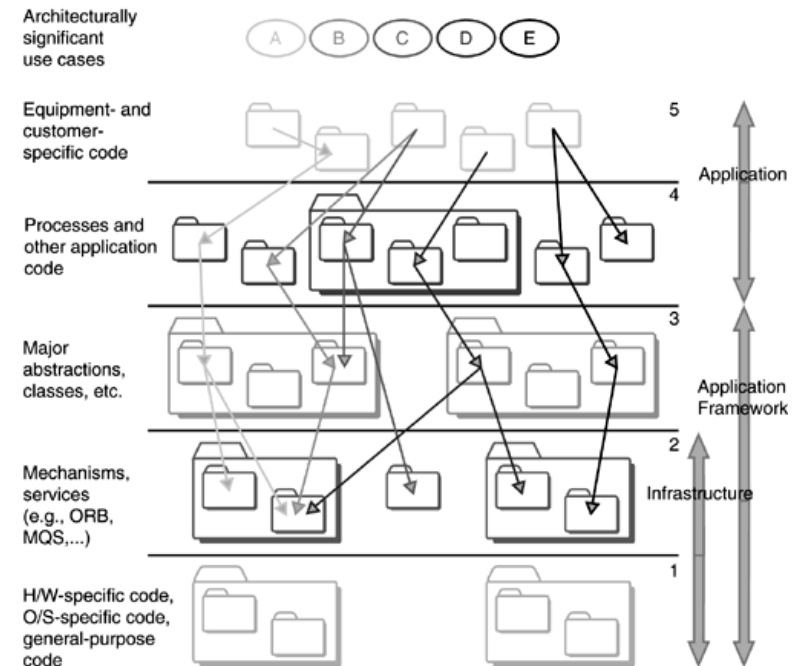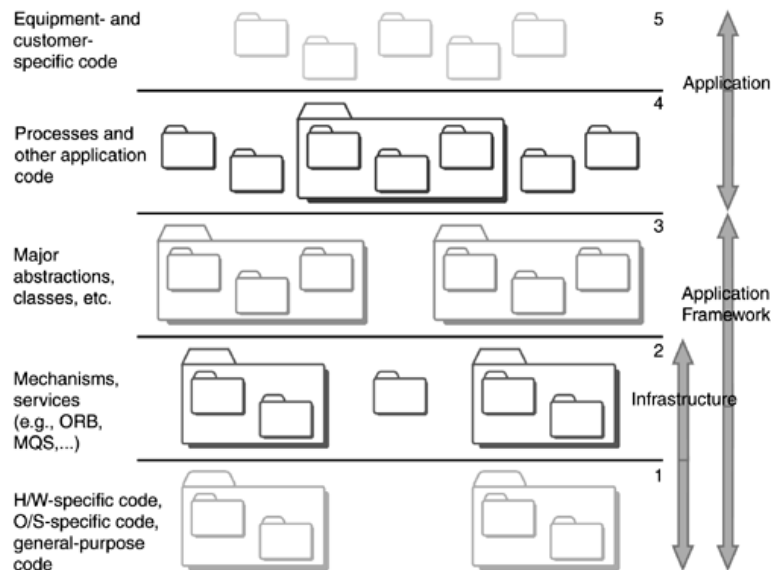
Objectives of the elaboration phase

- Get a more detailed understanding of the requirements

- Design, implement, validate, and baseline the architecture*

- Mitigate essential risks, and produce more accurate schedule and cost estimates

- Refine the development case and put the development environment in place
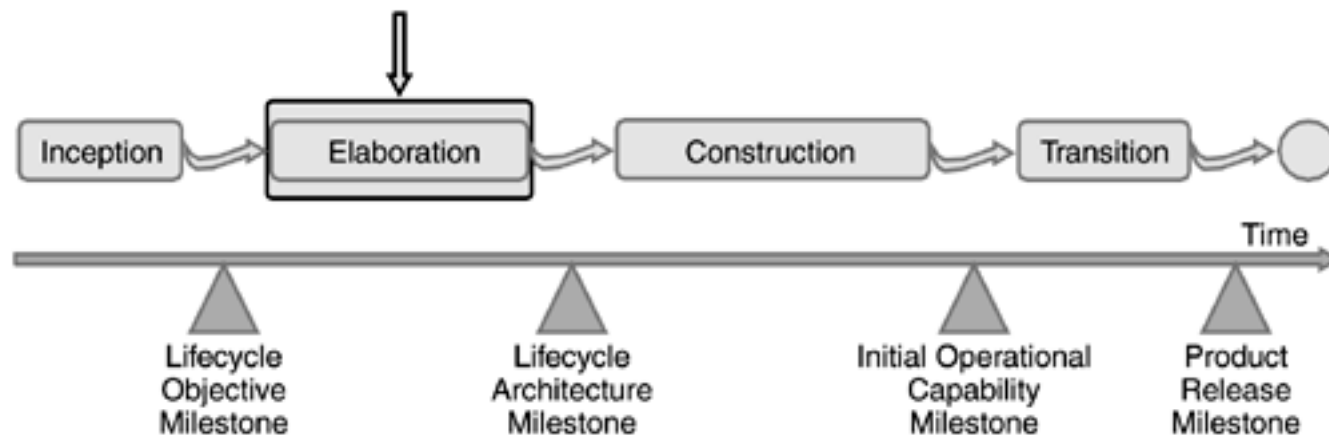
The Elaboration Phase

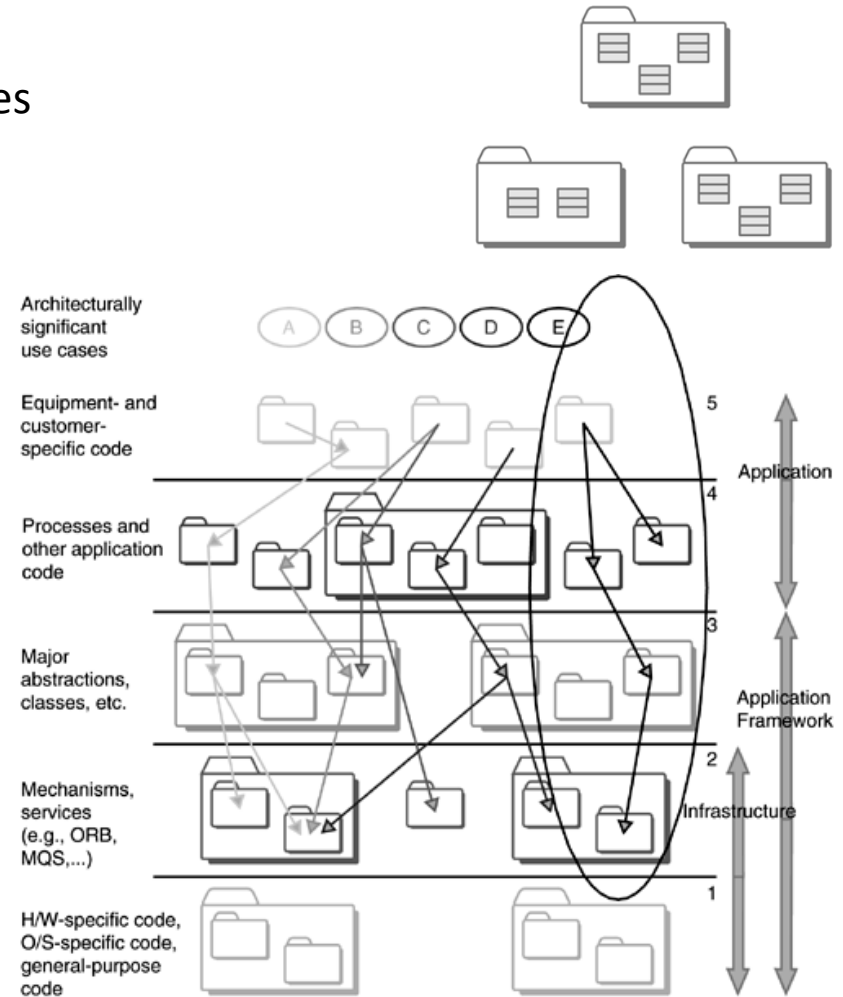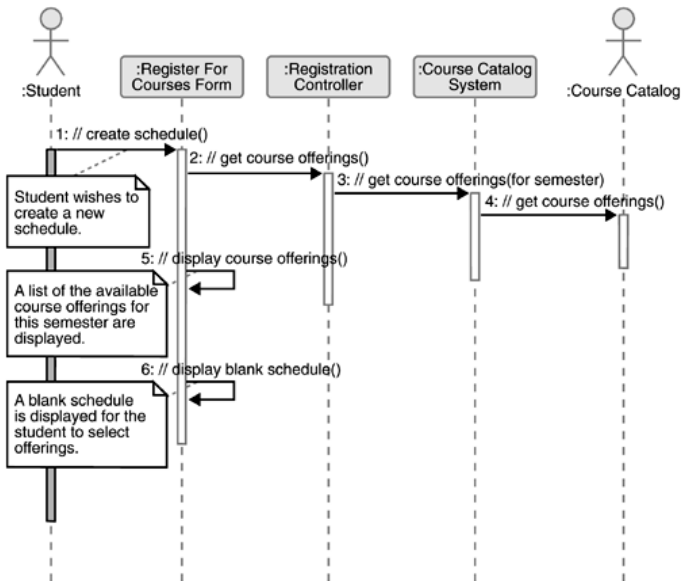**Objective 2: Design, implement, validate, and baseline the architecture**

- Architecture: Defining subsystems, key components, and their interfaces

- Use architecturally significant use cases to drive the architecture

The Elaboration Phase

Objective 2: Design, implement, validate, and baseline the architecture(Cont'd)

- Design critical use cases

- Consolidate and package identified classes

- Ensure architectural coverage

The Elaboration Phase

Objective 2: Design, implement, validate, and baseline the architecture (Still cont'd..)

- Design databases

- Outline concurrency, processes, threads, and physical distribution

- Identify architectural mechanisms

- Implement critical scenarios

- Integrate components

- Test critical scenarios

The Elaboration Phase

The outcome of the elaboration phase:

- A use-case model (at least 80%), all use cases and actors have been identified, and most use-case descriptions have been developed

- Supplementary requirements capturing the non functional requirements and any requirements that are not associated with a specific use case

- A software architecture description

- An executable architectural prototype

- A revised risk list and a revised business case

- A development plan for the overall project, including the coarse-grained project plan, showing iteration and evaluation criteria for each iteration

- An updated development case specifying the process to be used

- A preliminary user manual

The Elaboration Phase

Lifecycle architecture milestone

- Is the vision of the product stable?

- Is the architecture stable?

- Does the executable demonstration show the major risk elements have been addressed and credibly resolved?

- Is the plan for the construction phase sufficiently detailed and accurate? And it is backed up with a credible basis of estimates?

- Do all stakeholders agree that the current vision can be achieved if the current plan is executed to develop the complete system, in the context of the current architecture?

- Is the actual resource expenditure versus planned expenditure acceptable?

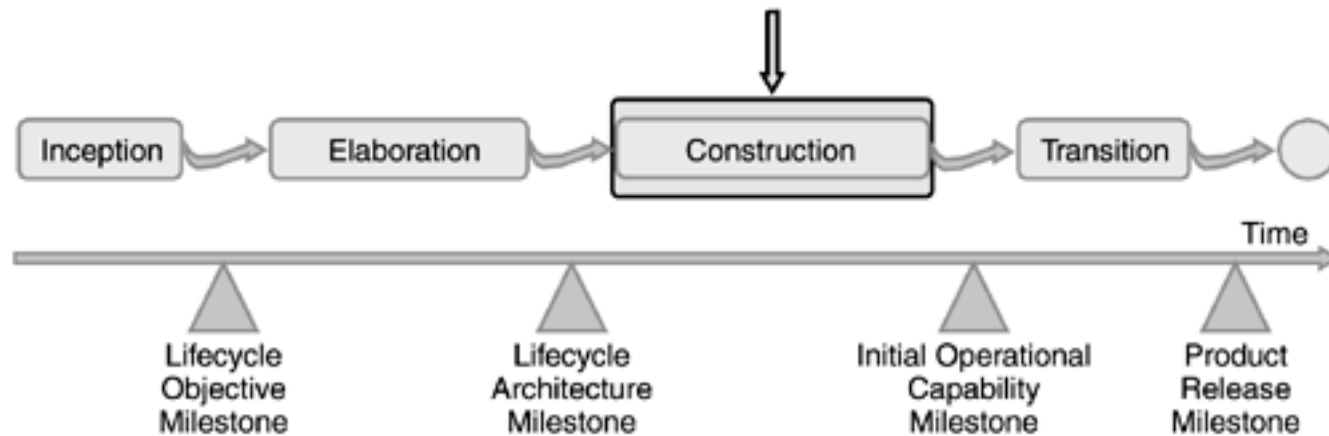**The project may be aborted or considerably re-thought**

The Construction Phase

The Construction Phase
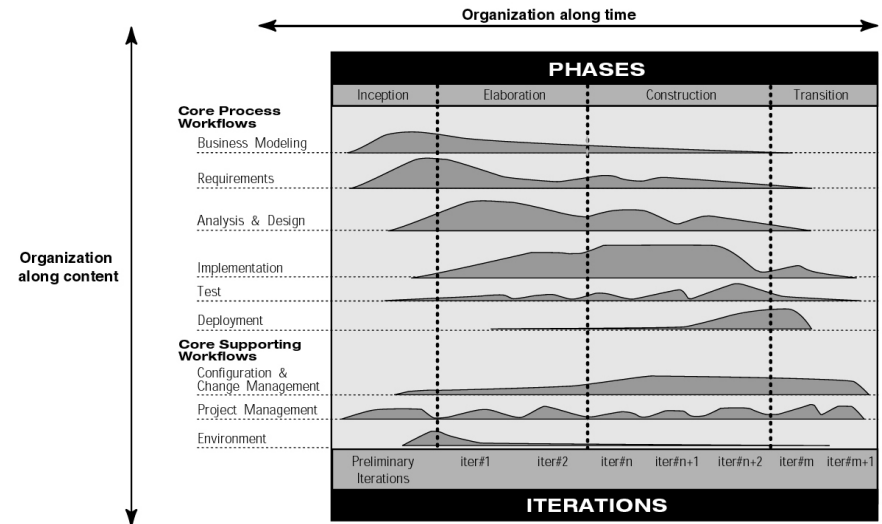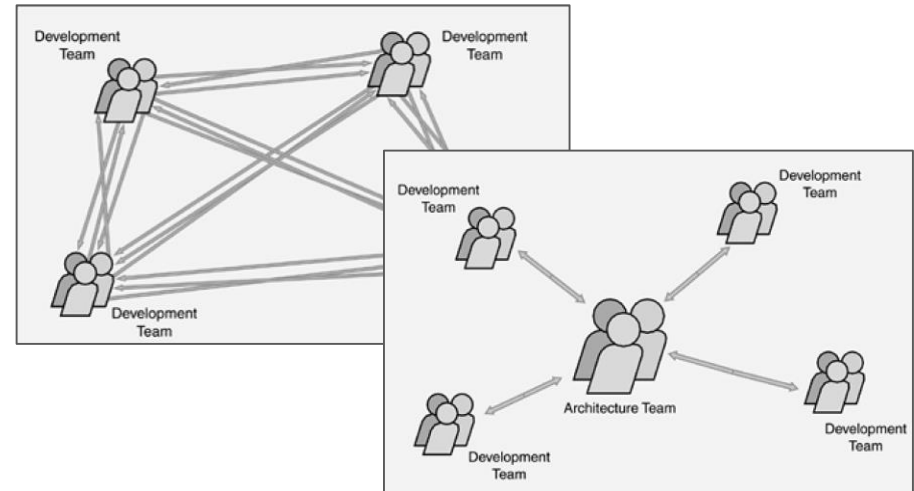
Objectives of the construction phase

- Minimize development costs and achieve some degree of parallelism in the work of the development teams

- Iteratively develop a complete product that is ready to transition to its user community

The Construction Phase

Objective 1: Minimize development costs and achieve some degree of parallelism

- Organize around architecture

- Configuration management
  - Integration planning for frequent build

- Enforce the architecture

- Ensure continual progress

The Construction Phase

Objective 2: Iteratively development a complete product that is ready to transition to its user community

- Describe the remaining use cases and other requirements

- Fill in the design

- Design the database

- Implement and unit-test code

- Do integration and system testing

- Prepare deployment
    - Early deployments and feedback loops
    - Prepare for beta deployment
    - Prepare for final deployment

The Construction Phase

The outcome of the Construction phase:

- The software product integrated on the adequate platforms

- The user manuals

- A description of the current release

The Construction Phase

Initial operational capability milestone

- Is this product release stable and mature enough to be deployed in the user community?

- Are all stakeholders ready for the transition into the user community?

- Are the actual resource expenditures versus planned expenditures still acceptable?

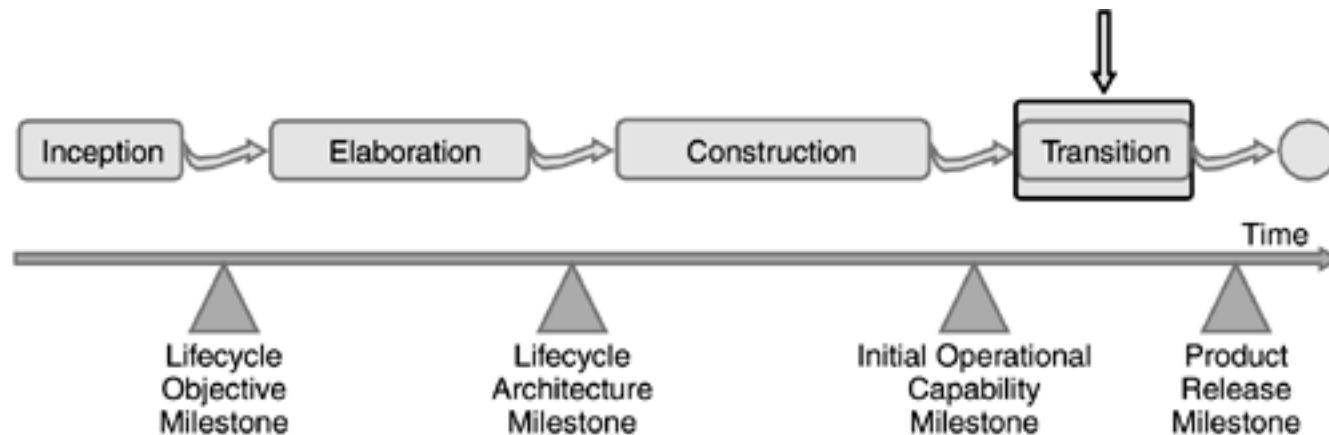**Transition may have to be postponed by one release**

The Transition Phase

The Transition Phase

Objectives of the transition phase:

- Beta test to validate that user expectation are met*
- Train users and maintainers to achieve user self-reliability
- Prepare deployment site and covert operational databases
- Prepare for launch-packaging, production and marketing roll-out; release to distribution and sales forces; field personnel training
- Achieve stakeholder concurrence that deployment baselines are complete and consistent with the evaluation criteria of the vision
- Improve future project performance through lessons learned*

The Transition Phase

Objective 1: Beta test to validate that user expectation are met

- Capturing, analyzing, and implementing change requests

- Transition testing
  - Continued test design and implementation
  - Regression testing
  - Acceptance testing (may not require additional test development)

  - Validate build stability
  - Test and evaluate
  - Achieve your test objectives, or in RUP parlance, achieve an acceptable mission
  - Improve test assets

- Patch releases and additional beta releases

The Transition Phase

Objective 5: Achieve stakeholder concurrence that deployment is complete

- Product acceptance Test
    - Formal acceptance testing
    - Informal acceptance testing
    - Beta testing

The Transition Phase

Product release milestone

- Is the user satisfied?

- Are the actual resources expenditures versus planned expenditures still acceptable?

The Transition Phase

Evolution cycles

- If the objectives were not met, you should start another development cycle

- There is an often overlap between two development cycles, transition phase and Inception phase