2009 Spring

Software Modeling & Analysis

- Fundamentals of Software Engineering
- Software Process Model

Lecturer: JUNBEOM YOO
jbyoo@konkuk.ac.kr

What is Software Engineering?


IEEE Std
610.12-1990

- [IEEE Standard 610.12-1990]
  Software Engineering: (1) The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. (2) The study of approaches as in (1).


NATO 1969

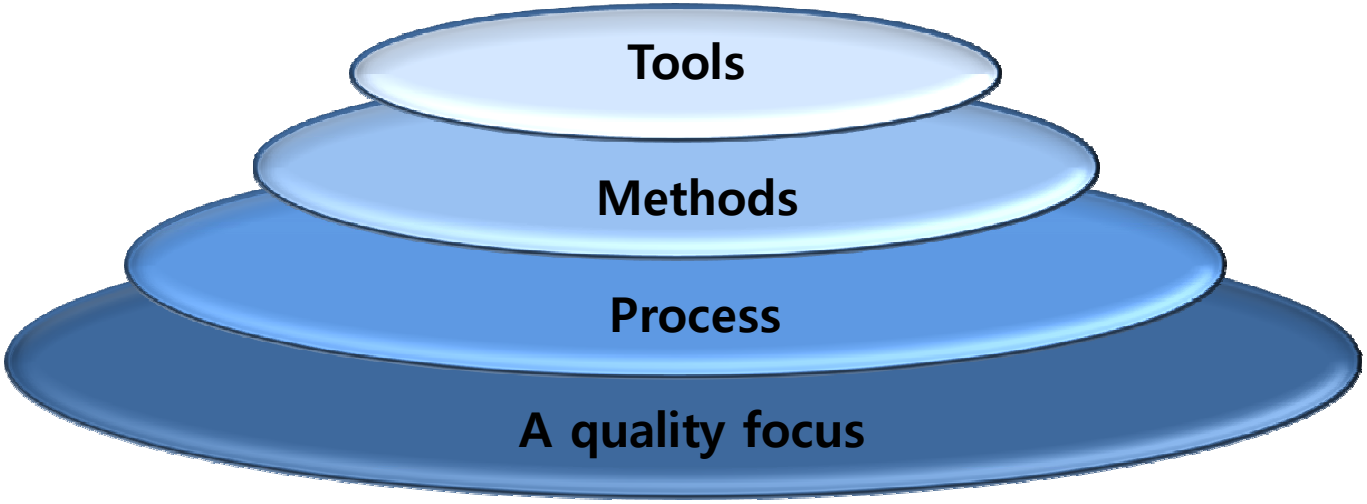
- [Fritz Bauer (1969)]
  Software Engineering is the establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines.

- [Watts Humphrey (1995)]
  The disciplined application of engineering, scientific, and mathematical principles and methods to the economical production of quality software.

What is Software Engineering?

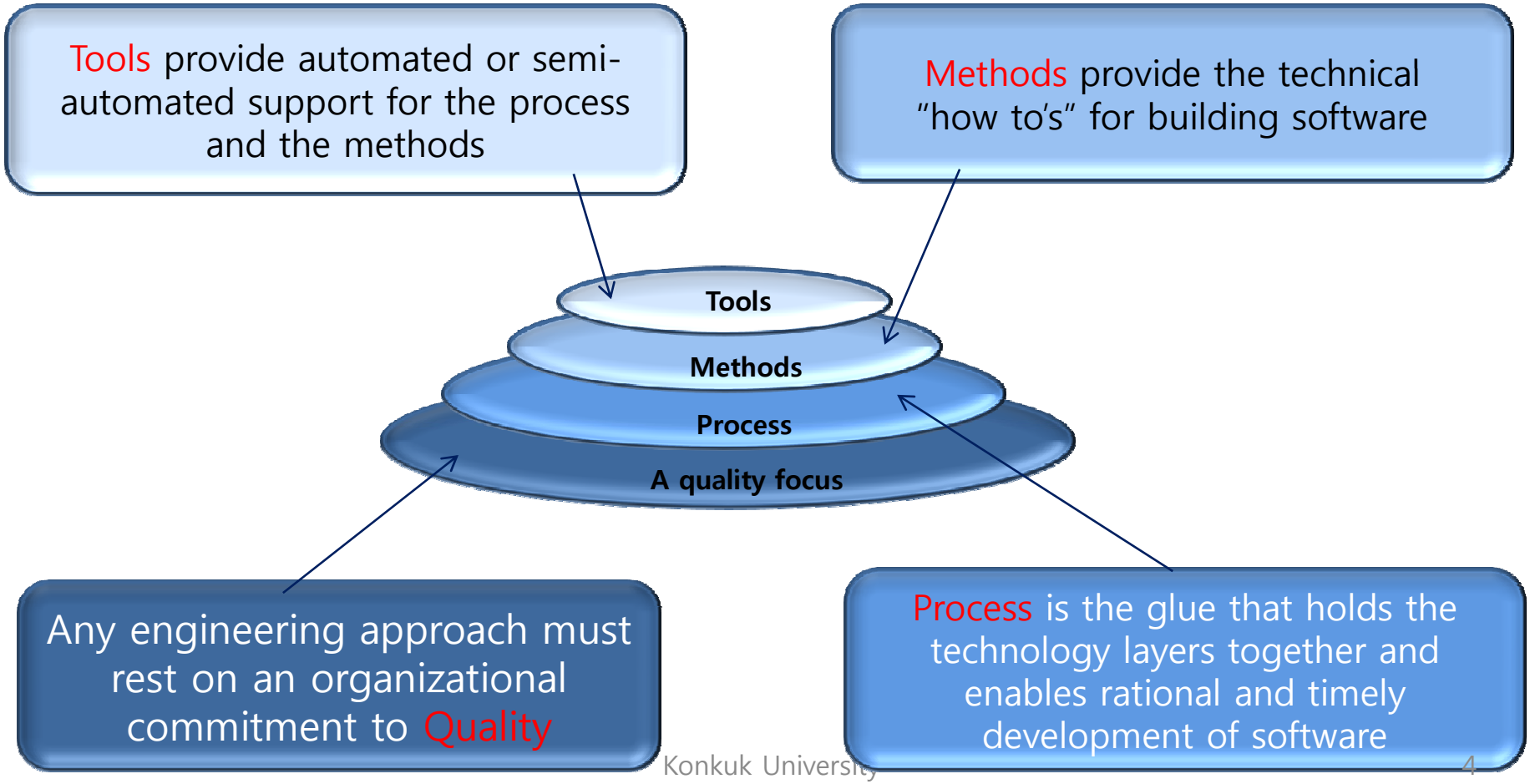
- [Junbeom Yoo (2008)]
  All activities to develop and manage software well

- [Roger S. Pressman (2005)]
  Software engineering is a layered technology.



What is Software Engineering?

- A layered technology



Software Engineering Technologies

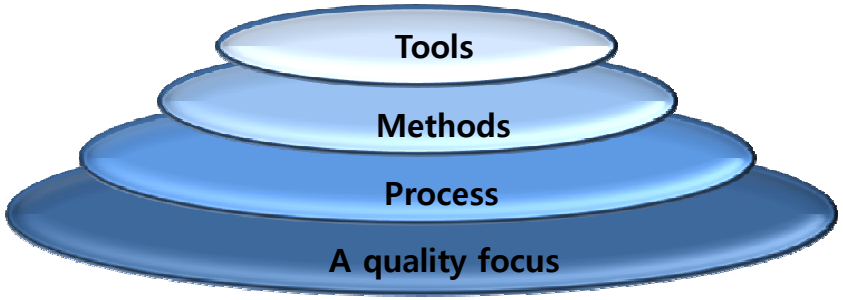
Requirements Engineering (Elicitation & Specification)
Reliability Engineering
Safety Analysis

Structured Analysis and Design
Object-Oriented Analysis and Design

Test Driven Development (TDD)
XP (eXtreme Programming)
RUP (Rational Unified Process)
UML (unified Modeling Language)

Software Process Improvement (SPI)
Configuration Management
Project Management
Software Architecture
Software Product Line

Design Patterns
Formal Methods (Formal Verification)
.NET J2EE XML WEB



Telelogic Tau, ClearCase, ClearQuest, SDL
Rational Rose
Visual Studio
SMV, SPIN, VIS

IEEE Software Engineering Standards

- Process standards
  - IEEE-Std 1074-1997,
    IEEE Standard for Developing Software Life Cycle Processes
  - IEEE Std 1012-1998,
    IEEE Standard for Software Verification and Validation


IEEE Std
1074-1997


IEEE Std
1012-1998

- Product standards
  - IEEE Std 829-1998,
    IEEE Standard for Software Test Documentation
  - IEEE Std 830-1998,
    IEEE Recommended Practice for Software Requirements Specifications


IEEE Std
829-1998


IEEE Std
830-1998

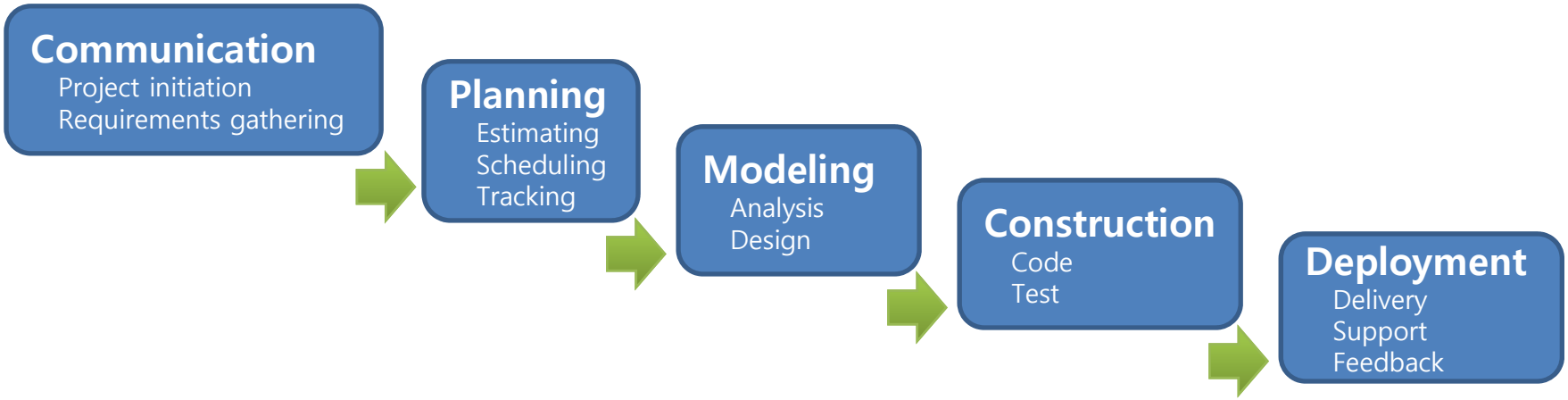
Software Process Model

- What is it?
  - Process models define a distinct set of activities, actions, tasks, milestones, and work products that are required to engineer high-quality software.
  - Defines Who is doing What, When to do it, and How to reach a certain goal.
  - Process models were originally proposed to bring order to the chaos of software development.

- Typical Process Models
  - Waterfall Model / Incremental Model
  - Evolutionary Models (Prototyping, Spiral)
  - Specialized Model (Component-Based Development, Formal Methods)
  - Unified Process (RUP)

Waterfall Model

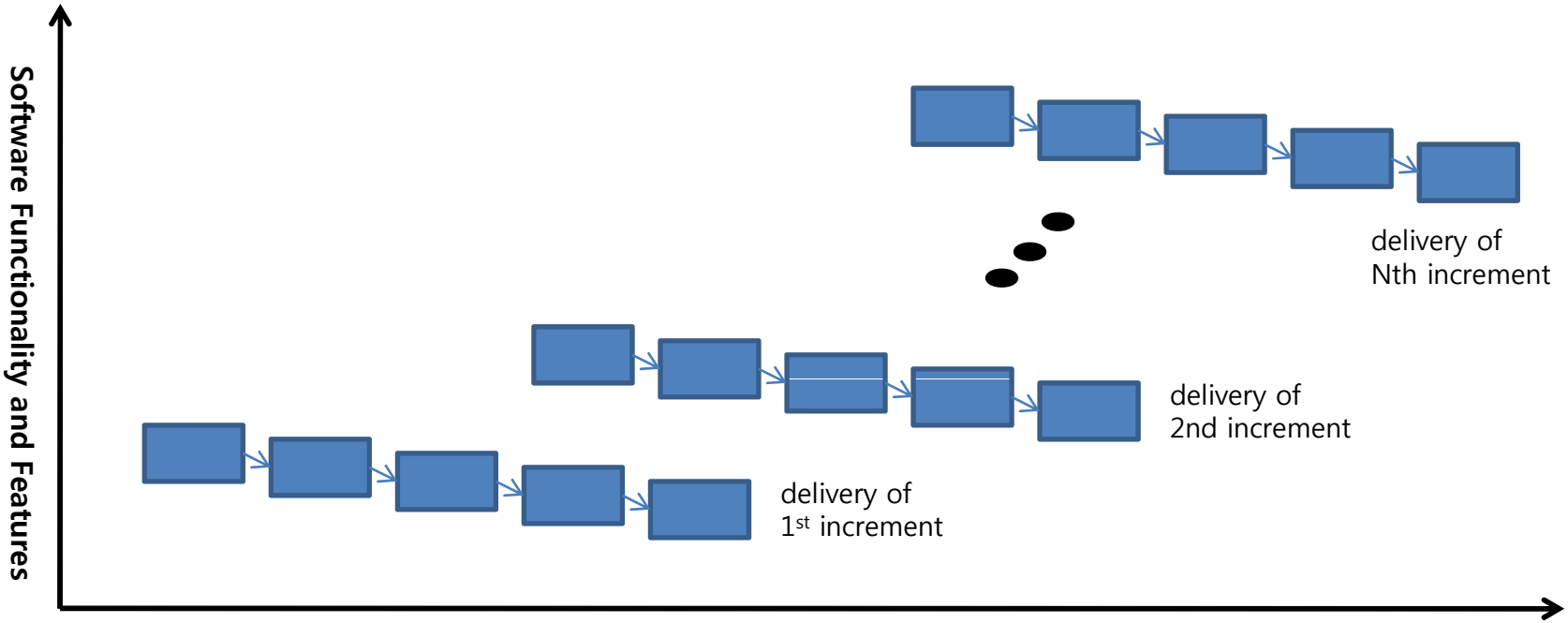
- A classic life cycle model
  - Suggests a systematic, sequential approach to software development
  - The oldest paradigm
  - Useful in situations where requirements are fixed and work is to proceed to completion in a linear manner



More Info.: http://en.wikipedia.org/wiki/Waterfall_model

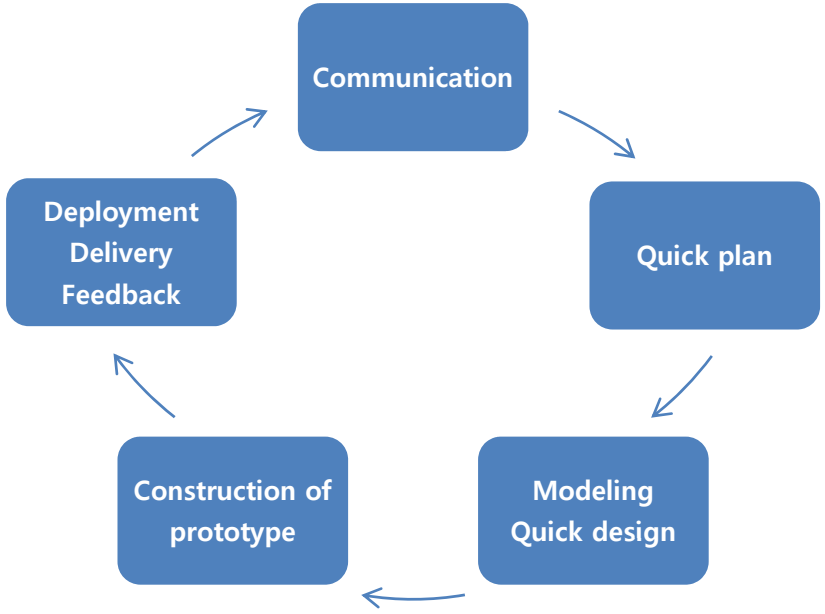
Incremental Model

- What is it?
  - Combines elements of the waterfall model applied in an iterative fashion
  - Delivers a series of releases(increments) that provides progressively more functionality for the customer as each increment is delivered



Evolutionary Model

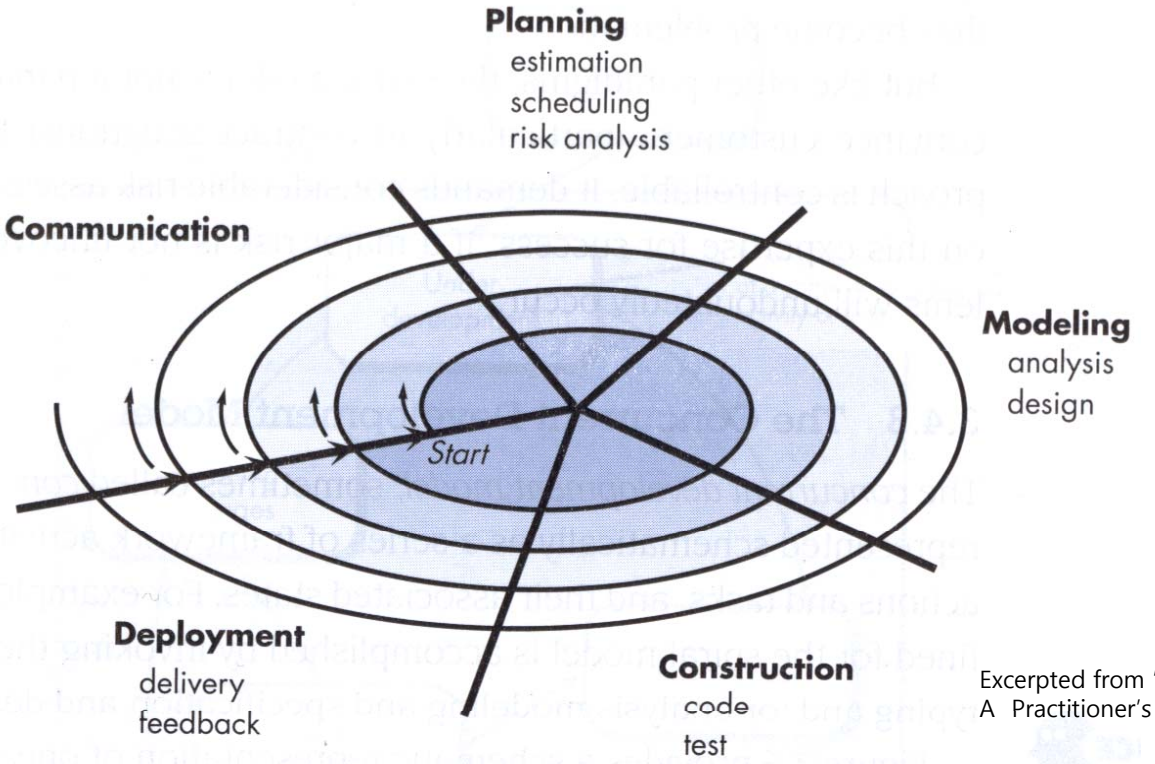
- Prototyping Model
  - Used when customer does not indentify detailed requirements
  - Used when developers may be unsure of the efficiency of the algorithm, adaptability of OS, or the form of HMI should take
  - Commonly used within the context of other process models



More Info.: http://en.wikipedia.org/wiki/Software_prototyping

Evolutionary Model

- Spiral Model
  - Software is developed in a serious of evolutionary releases



Excerpted from "Software Engineering: A Practitioner's Approach" by Roger S. Pressman

More Info.: http://en.wikipedia.org/wiki/Spiral_model

Specialized Model

- Component-Based Development (CBD)
  - Use commercial off-the-shelf(COTS) software components
  - CBD-based Software development steps:
    1. Available component-based products are researched and evaluated
    2. Component integration issues are considered
    3. A software architecture is designed to accommodate the component
    4. Components are integrated into the architecture
    5. Comprehensive testing is conducted to ensure proper functionality

- Let's give your opinion upon CBD !!!

Specialized Model

- Formal Methods
  - Formal Specification : write software requirements mathematically (logically) with support of automatic tools
  - Formal Verification : prove its correctness mathematically
  - Aiming at defect-free software
  - Highly recommended to use to develop safety-critical systems
    - Nuclear Power Plants
    - Railroad Control
    - Satellite Control
    - Aerospace Industry (i.e. NASA)

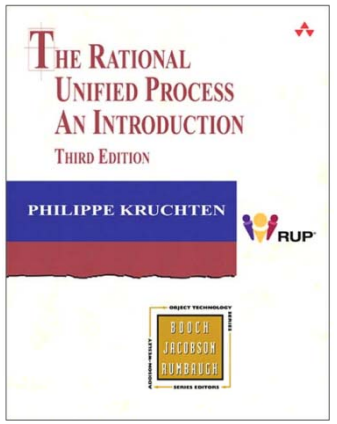
A specifiers
duction to formal

Formal
ecification Roadm

- Let's give your opinion upon Formal Methods !!!

Unified Process

- called as Rational Unified Process (RUP)



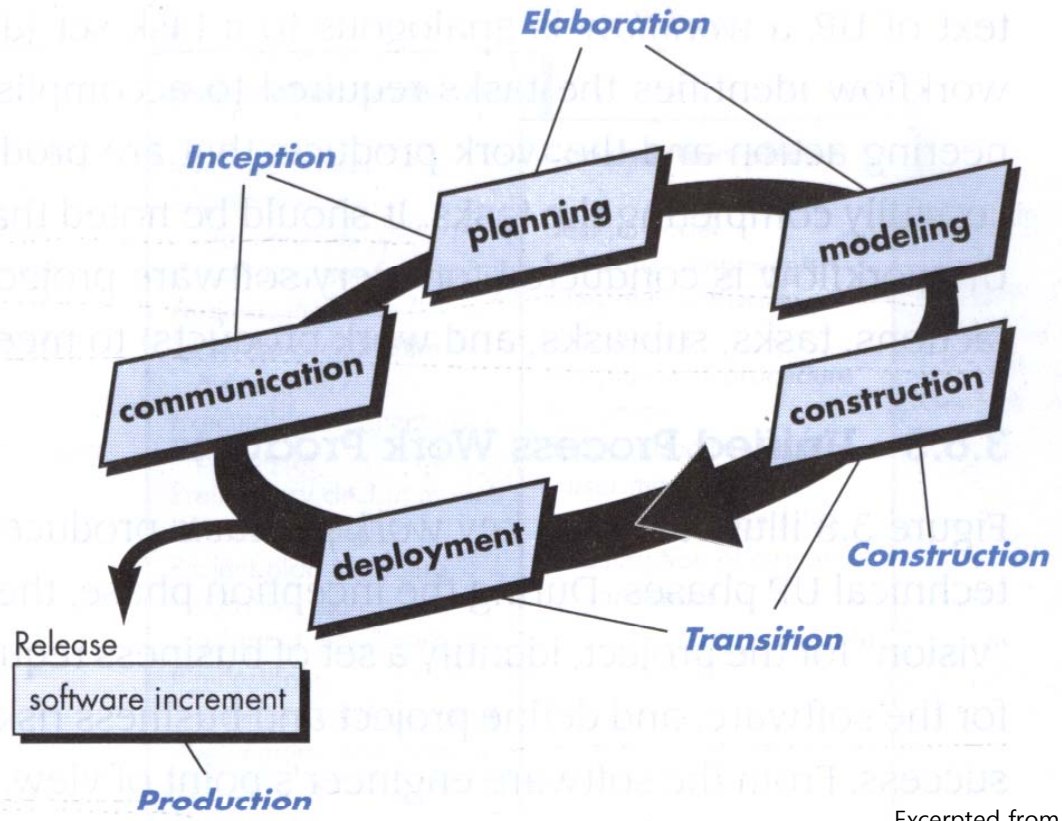
- What is it?
  - A Software development approach that is
    - Iterative,
    - Architecture-centric, and
    - Use-case-driven
  - A Well-defined and well-structured software engineering process
  - A Process product that provides you with a customizable process framework for software engineering

The Spirit of the RUP

- Essential Principles
  - 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.

RUP : A Software Development Process

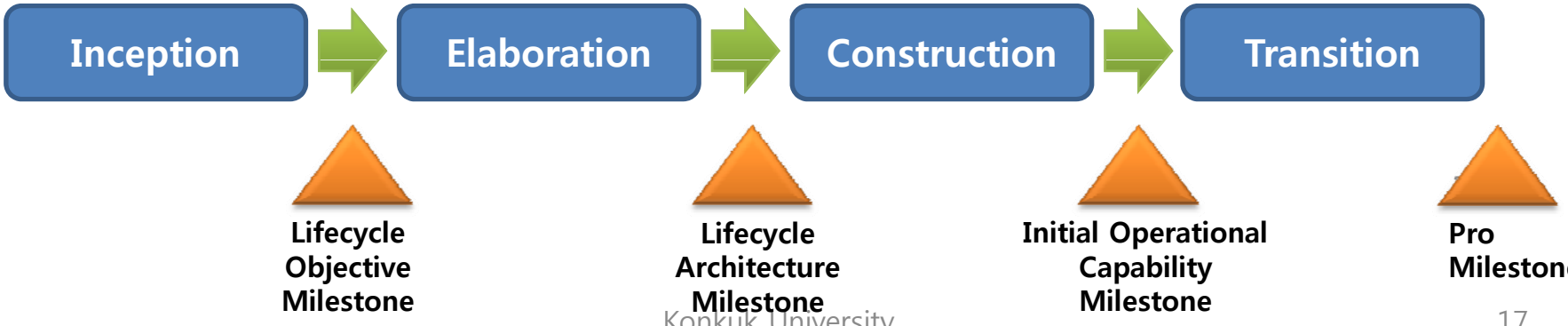
- An Iterative Development



Excerpted from "Software Engineering:
A Practitioner's Approach" by Roger S. Pressman

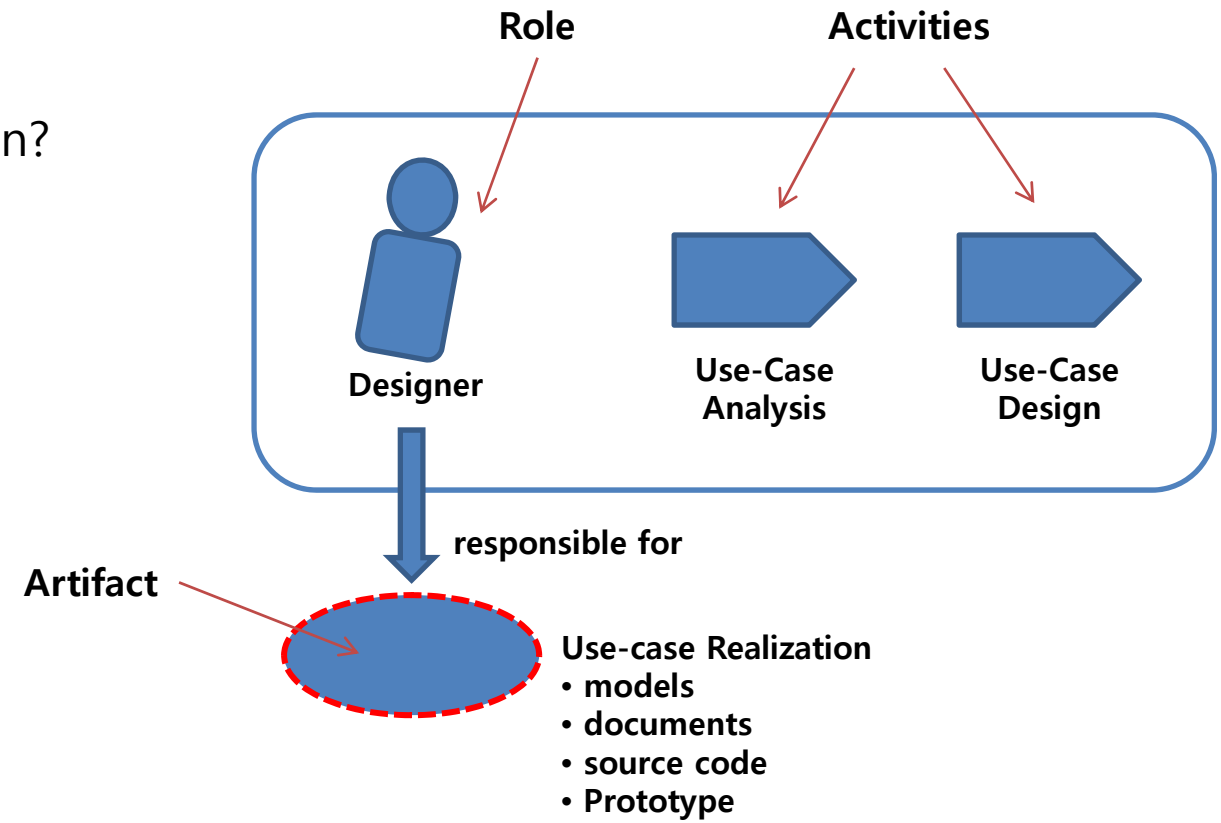
RUP :
A Well-Defined Software Engineering Process

- Dynamic Structure of RUP
  1. Inception Phase:
     - Define the scope and lifecycle of the project
  2. Elaboration Phase:
     - Mitigate risks and create a stable baseline architecture
  3. Construction Phase:
     - Develop the remainder of the system as efficiently as possible
  4. Transition Phase:
     - Get customer acceptance of the product



RUP :
A Well-Defined Software Engineering Process

- Static Structure of RUP
  1. Role : Who?
  2. Activity : How?
  3. Artifact : What?
  4. Workflow : When?

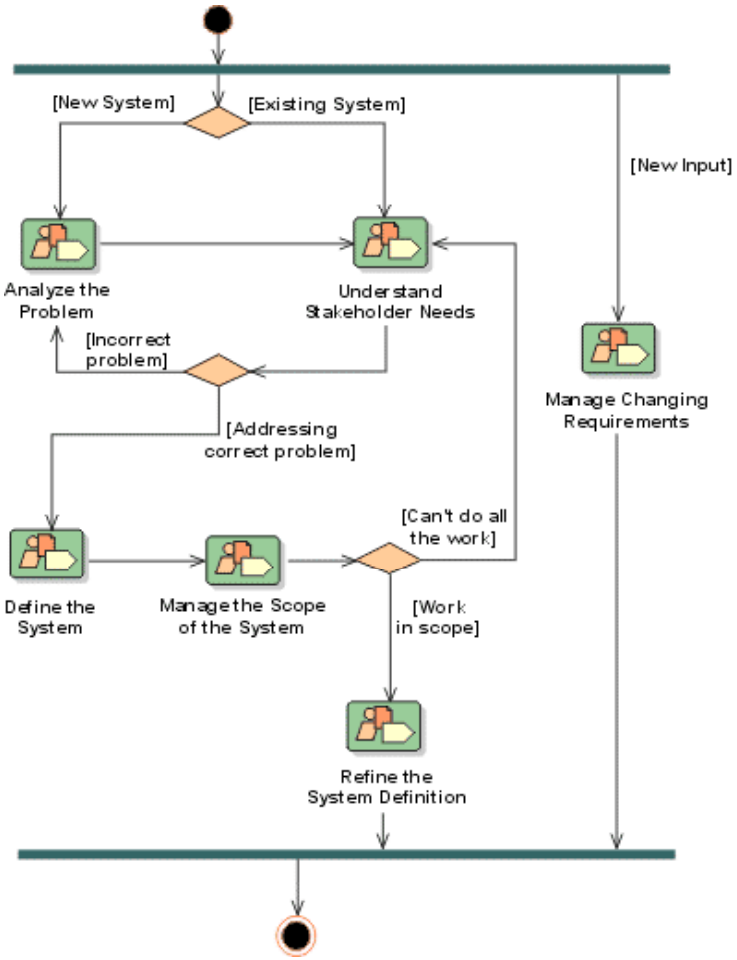


RUP :
A Well-Defined Software Engineering Process

- Workflow
  - A way to describe meaningful sequences of activities
  - A way to show interactions between roles
  - 2 forms of workflows

1. Disciplines :
- High-level workflows

2. Workflow Details :
- Workflows within a discipline



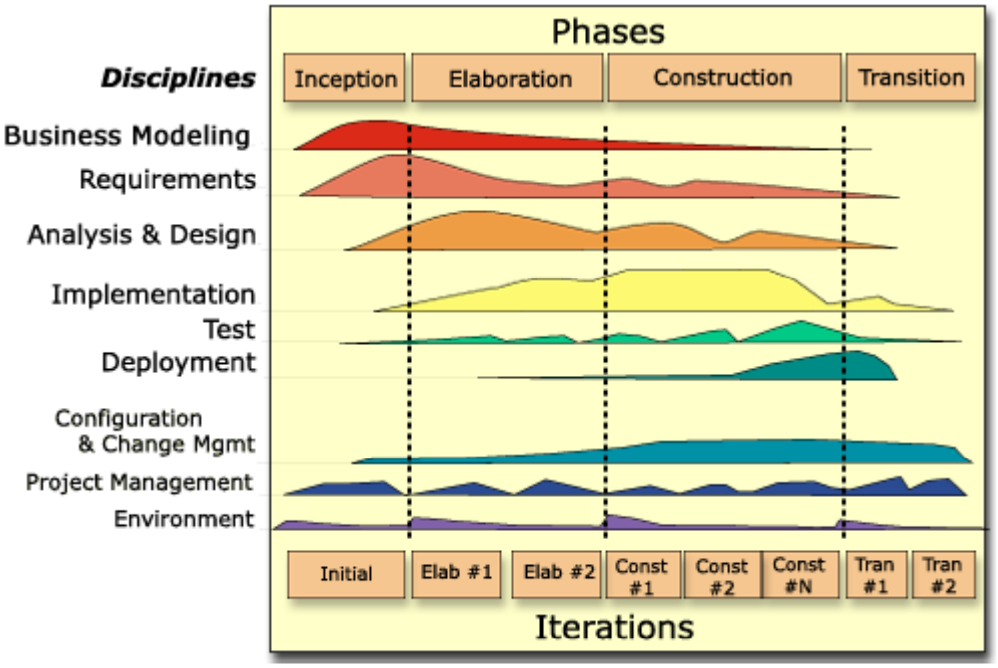
Excerpted from "RUP iteration planning" in www.ibm.com

RUP :
A Well-Defined Software Engineering Process

- 9 Disciplines
  - Logical containers of all process elements

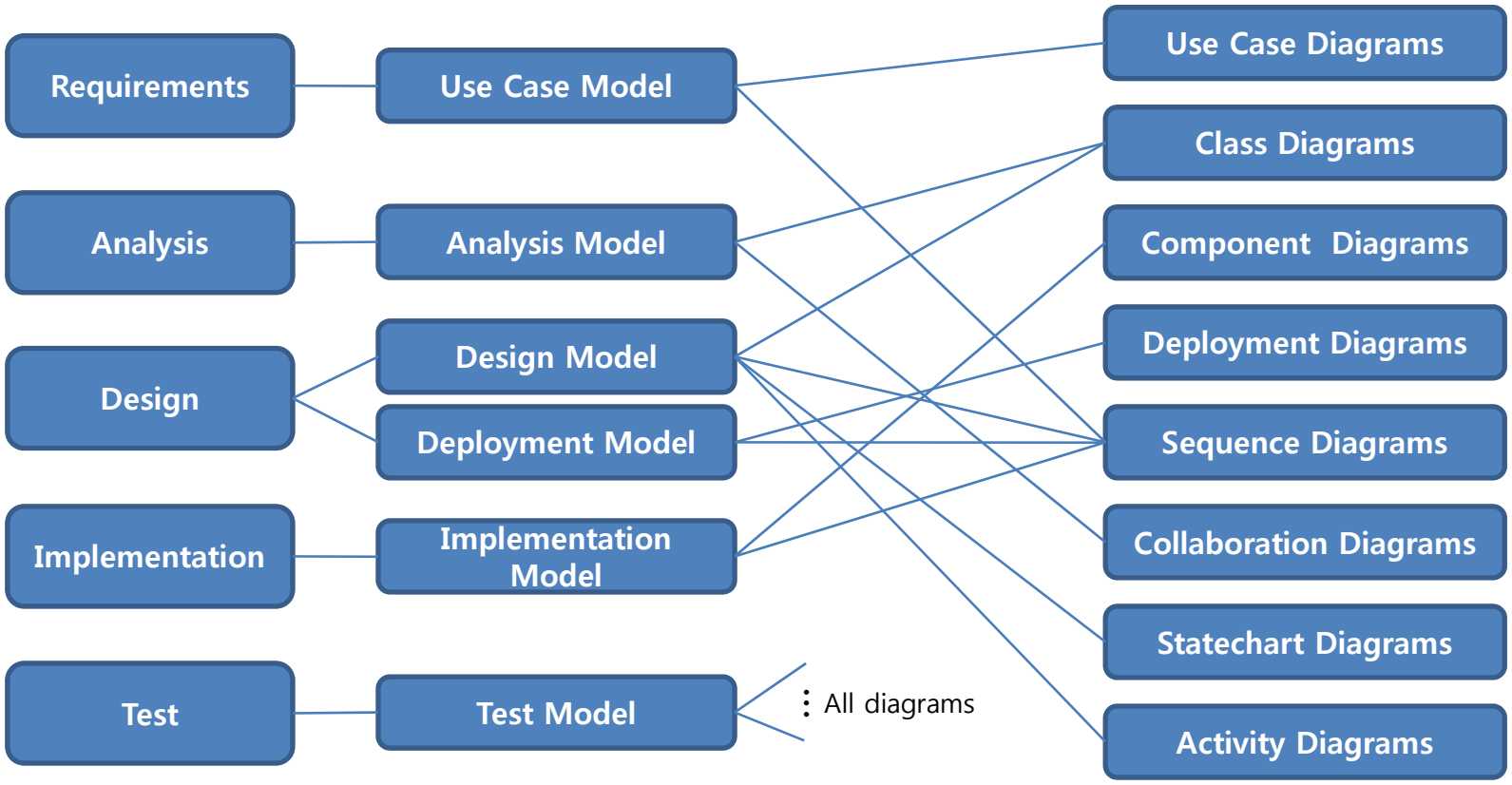
1. Business modeling
2. Requirements managemen
3. Analysis and design
4. Implementation
5. Test
6. Deployment

7. Change management
8. Project management
9. Environment



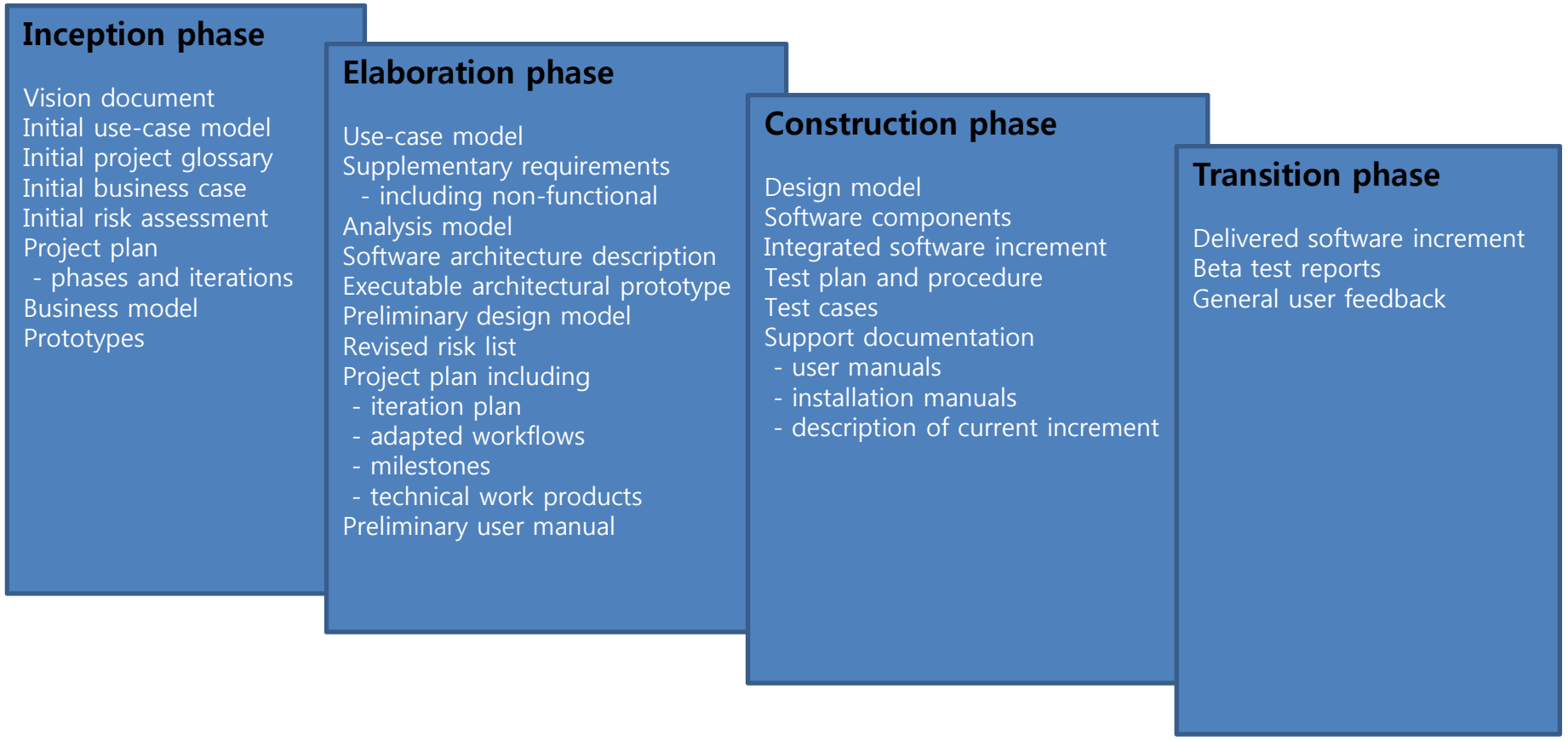
RUP :
A Well-Defined Software Engineering Process

- Workflow Details
  - Each workflow creates one or more models implemented with UML



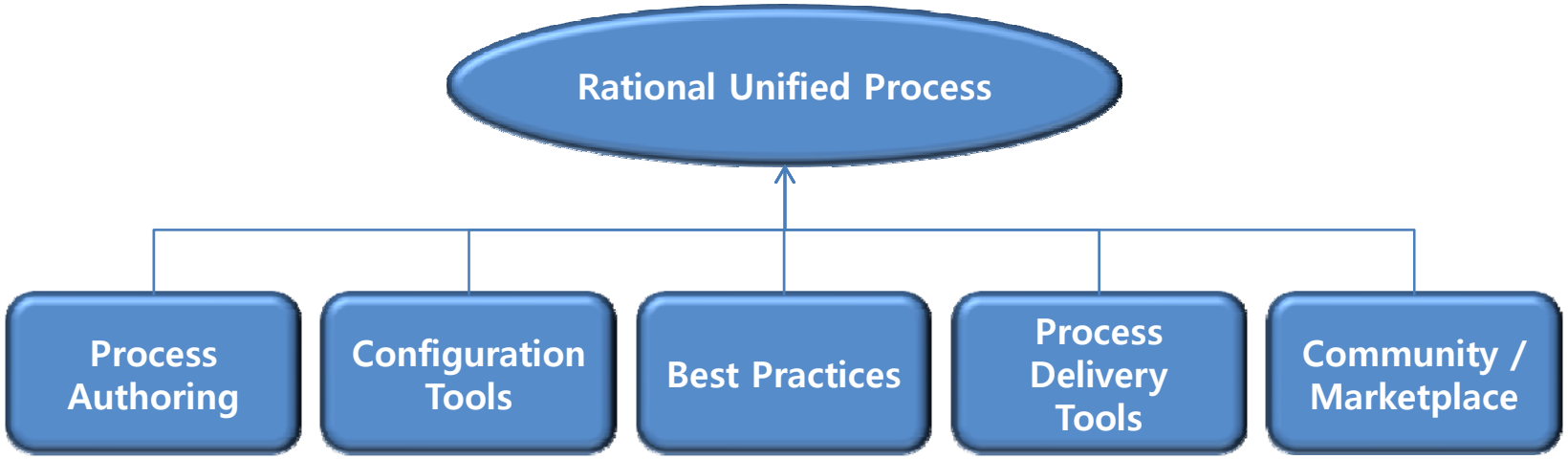
RUP :
A Well-Defined Software Engineering Process

- Major work products produced for each RUP phase



RUP : A Customizable Process Product

- To accommodate various needs requiring a process that is adapted to their specific situation
  1. Best practices
  2. Configuration tools: selecting appropriate best practices
  3. Process delivery tools: accessing selected best practices
  4. Online community: exchanging artifacts and experiences with others
  5. Process authoring tools: adding new best practices to the RUP



Summary

- What is Software Engineering?
- What is Software Process Model?
  - Why you have to use the software process model?
  - Can you clarify the difference between typical process models?
- What is the RUP?
  - What is the relationship between UML and RUP?

Recommended Supplementary Text