



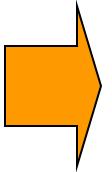
UNIVERSITY OF SCIENCE  
HO CHI MINH CITY

# Software Processes Traditional Methods

Nguyen V. Vu

with some materials adapted from Boehm and Sommerville

# Outline

- 
- Introduction to Software Processes
  - Planning Spectrum
  - Plan-driven Methods
    - Waterfall Model
    - Spiral Model
    - Rational Unified Process
    - Capability Maturity Model Integration (CMMI)

# What is an IS/software process?

- A set of activities whose goal is the development or evolution of software
- A process includes the answers on
  - What?
  - Who?
  - When?
  - How?
- Generic activities in software processes
  - Specification - what the system should do and its development constraints
  - Development - production of the software system
  - Validation - checking that the software is what the customer wants
  - Evolution - changing the software in response to changing demands.

# What is a software process model?

- A **simplified representation** of a software process, presented from a specific perspective
- Popular process models
  - Waterfall
  - Iterative development
  - Rational Unified Process (RUP)
  - Agile methods
    - Scrum
    - Kanban
    - XP
    - Test-driven development

# Two approaches of process model

- Two approaches to software development

- Traditional (document-based, heavy process)
  - Modern - Agile  
(XP, tacit knowledge, light process)



- Both have strengths and weaknesses

# Key Definitions

- Agile method
  - one which fully adopts the four value propositions and twelve principles in the Agile Manifesto
- Plan – (per Webster)
  - a method for achieving an end (a process plan);
  - an orderly arrangement of parts of an overall design (a product plan)
- Plan-driven
  - a description for disciplined methods (order is often defined in plans)

# The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it.  
Through this work we have come to value:

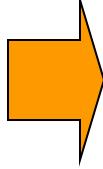
- **Individuals and interactions** over processes and tools
- **Working software** over comprehensive documentation
- **Customer collaboration** over contract negotiation
- **Responding to change** over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

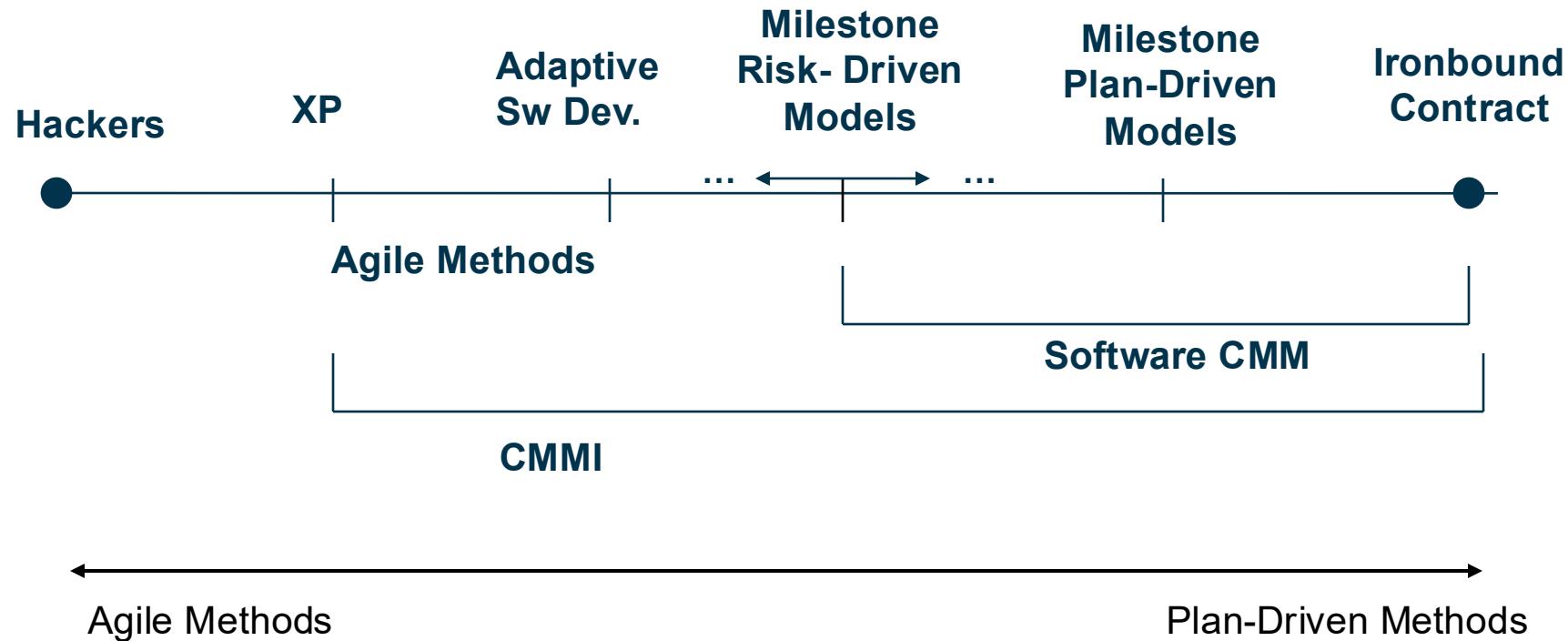
# Popular Plan-driven and Agile Methods

- Plan-driven methods
  - **Waterfall**
  - **Spiral**
  - **Rational Unified Process**
  - Personal Software Process
  - **CMMI**
- Agile methods
  - Extreme Programming
  - Scrum
  - Crystal
  - Feature Driven Development (FDD)

# Outline

- 
- Introduction to Plan-driven and Agile Methods
  - Planning Spectrum
  - Plan-driven Methods
    - Waterfall Model
    - Spiral Model
    - Rational Unified Process
    - Capability Maturity Model Integration (CMMI)

# Different Process Models



(Boehm 2002)

# Outline

- Introduction to Plan-driven and Agile Methods
- Planning Spectrum
- Plan-driven Methods
  - Waterfall Model
  - Spiral Model
  - Rational Unified Process
  - Capability Maturity Model Integration (CMMI)
  - Strengths and Weaknesses

# Plan-Driven Methods

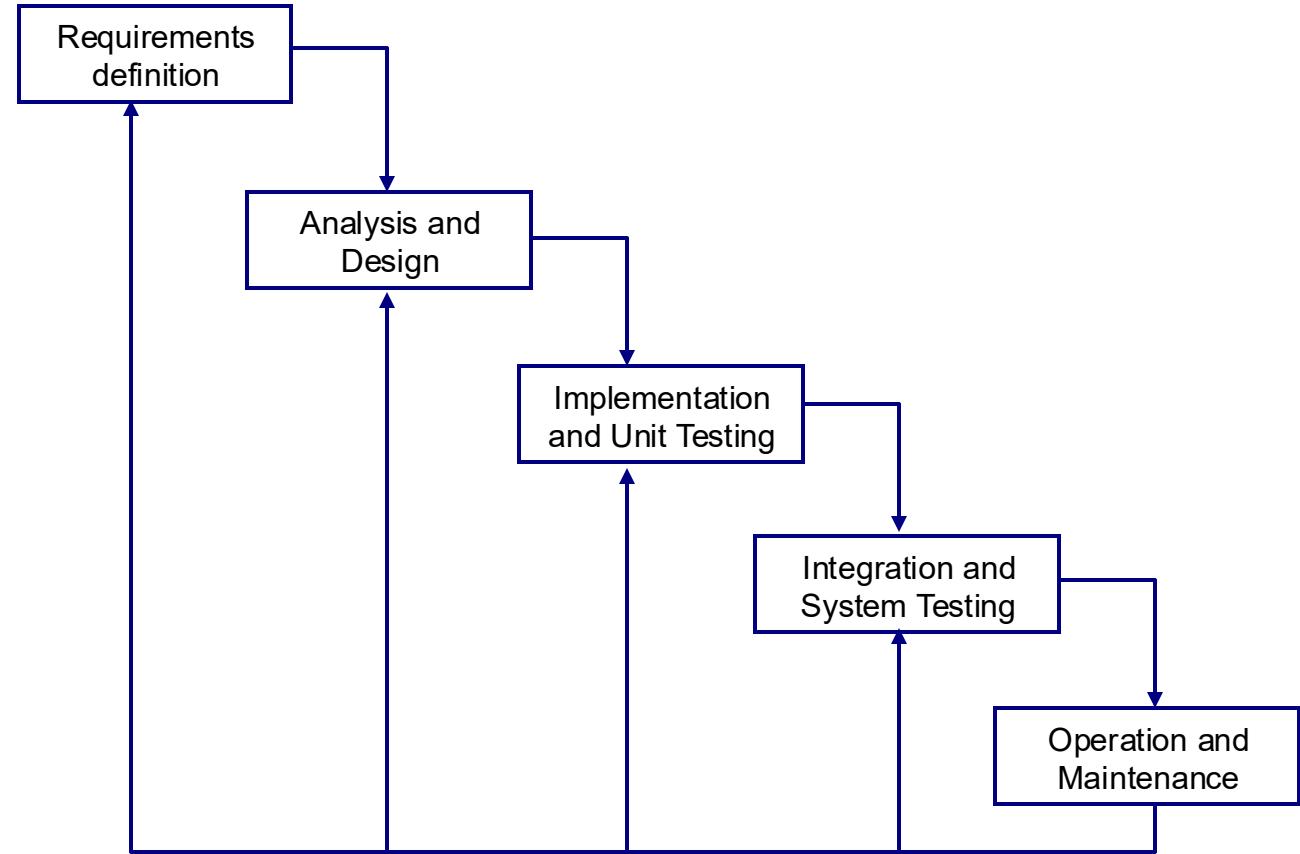
- Structured approaches to software development that are guided or driven by plans
- Also called as disciplined methods
- Often referred to as traditional methods
- Often defined in plans and relied on documentation

# Outline

- Introduction to Plan-driven and Agile Methods
- Planning Spectrum
- Plan-driven Methods
  - ➡ Waterfall Model
  - ➡ Spiral Model
  - ➡ Rational Unified Process
  - ➡ Capability Maturity Model Integration (CMMI)

# Waterfall Model - 1

- Key characteristics
  - Linear
  - Sequential
  - Well structured

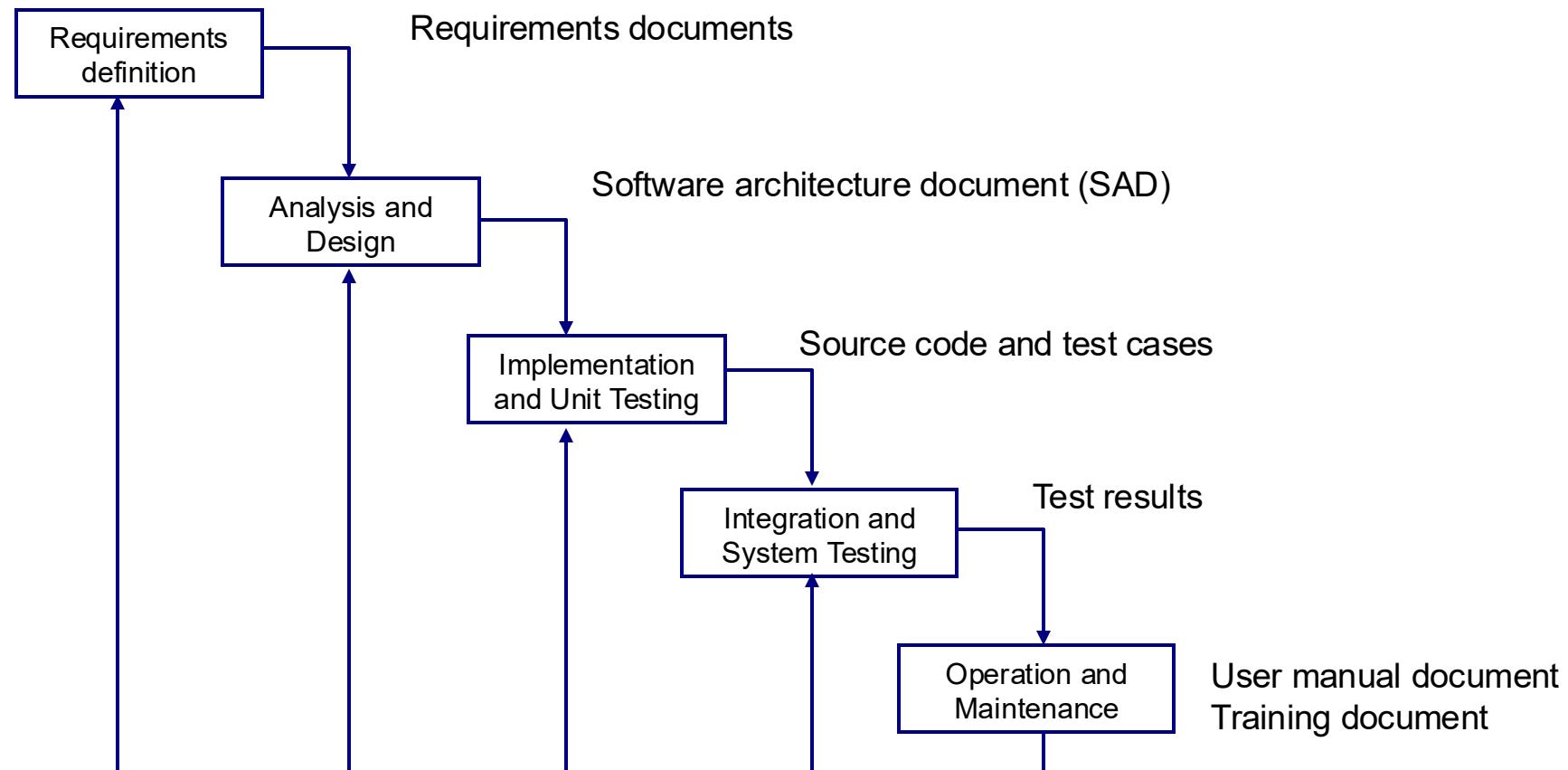


# Waterfall Model - 2

- Two stages cannot be performed at the same time
  - Development is not started when requirements are not baselined
  - Baseline requirements are ready to be based on
- Each stage has to be completed before doing the next stage
- A stage is completed if a set of criteria is met
  - Using checklists of criteria check
- Going back to the previous stage is possible

# Waterfall Model - 3

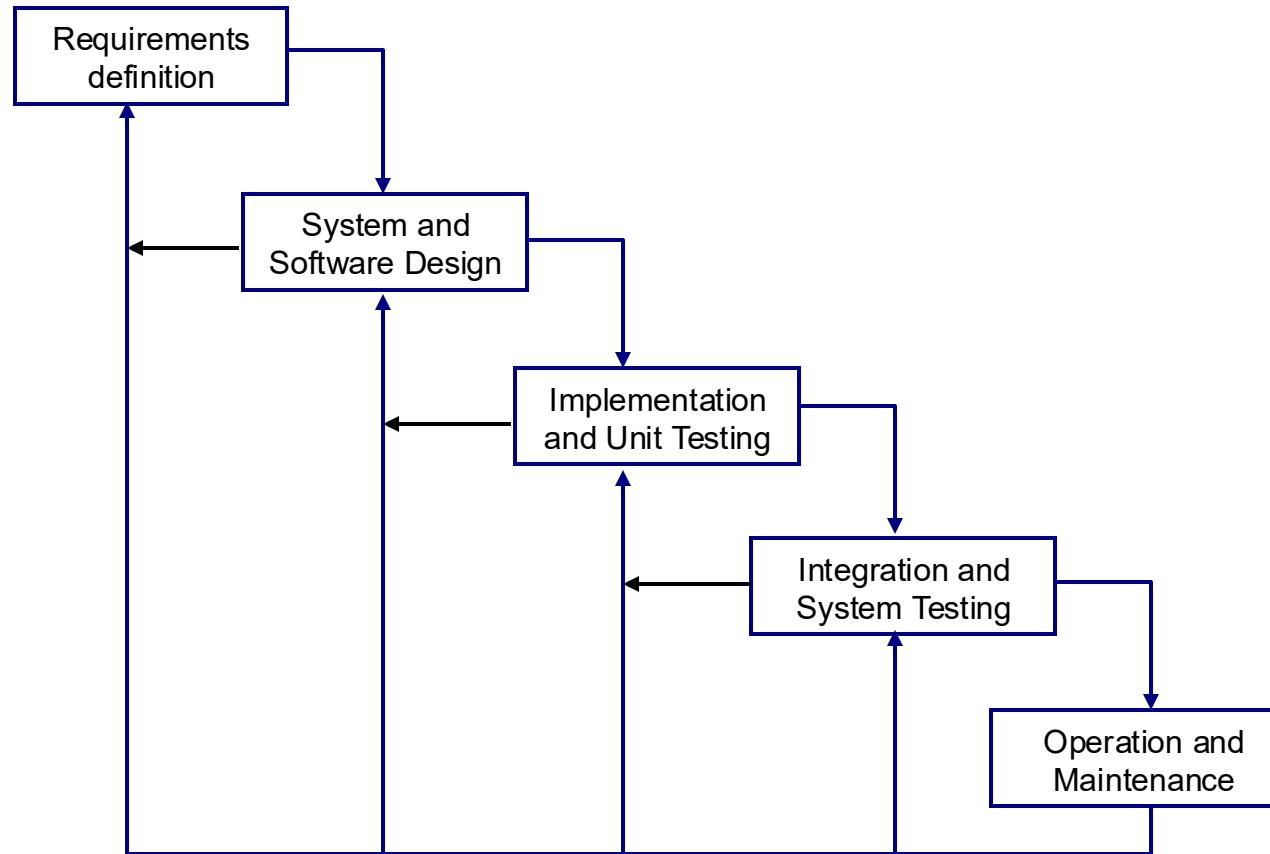
- A set of standard outcomes of each stage is defined



# Waterfall Model Problems

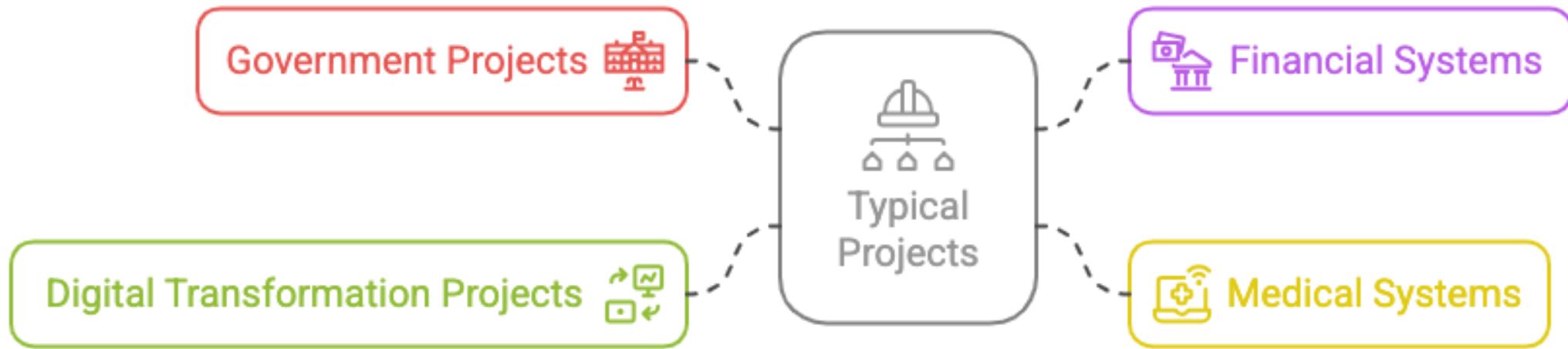
- Inflexible partitioning of the project into distinct stages
  - difficult to respond to changing requirements
- Only appropriate when the requirements are well-understood
  - changes will be fairly limited during the design process
  - But, few business systems have stable requirements
- Mostly used for large systems engineering projects

# Waterfall Model Revised



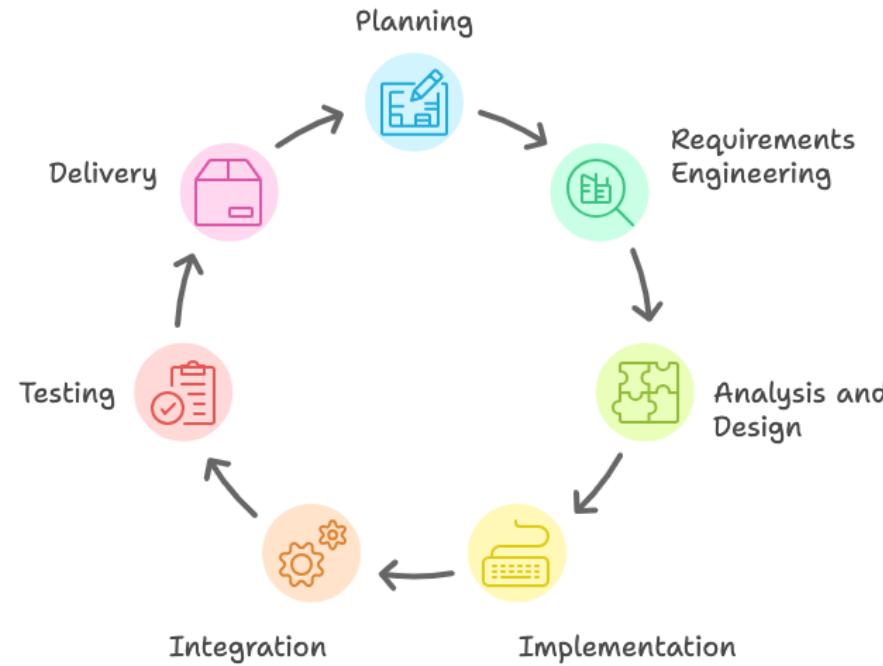
# When to use the Waterfall Model?

- Projects with stable, clear requirements



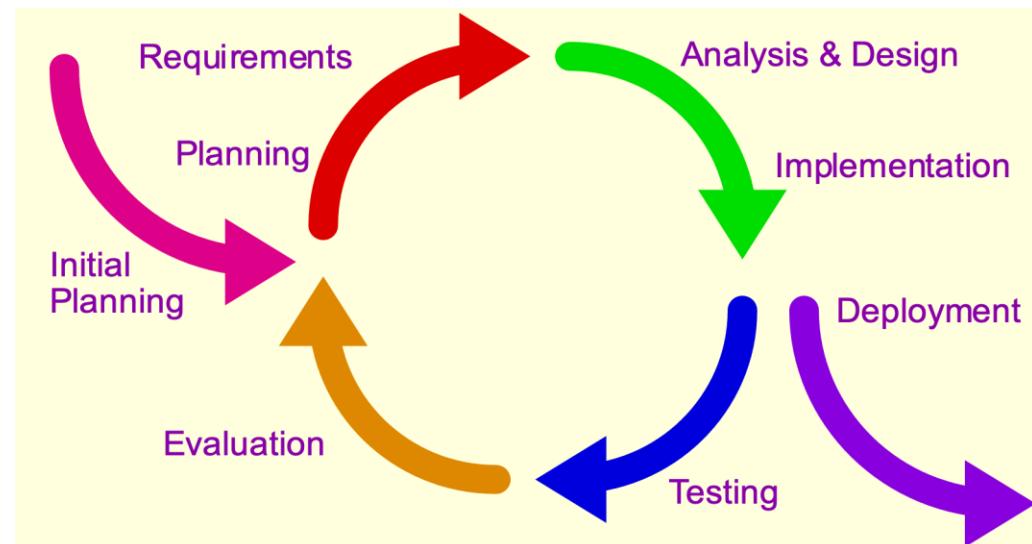
# Iterative development models

- Project is divided into many iterations
- Each iteration includes many different activities



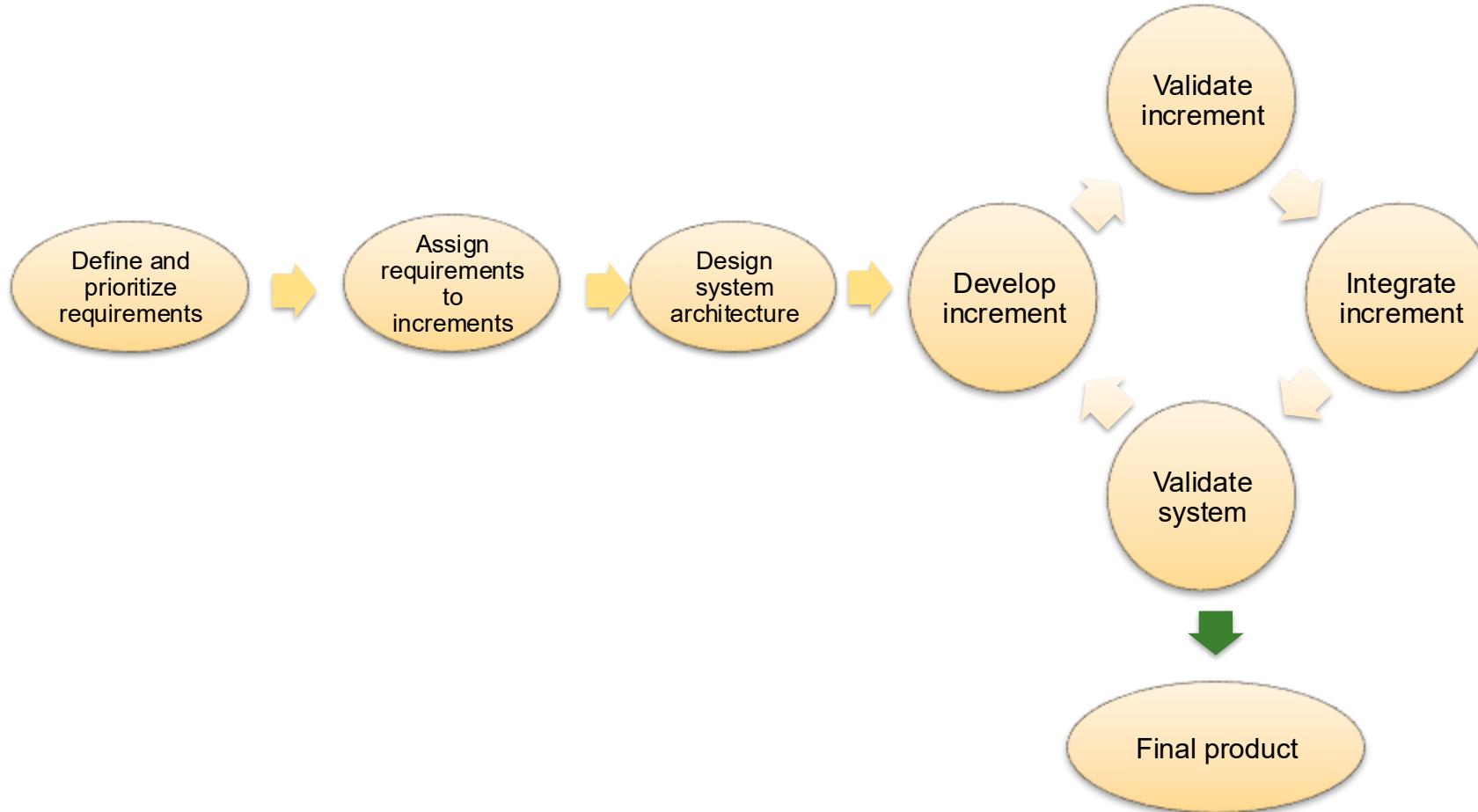
# Incremental development

- Development and delivery is broken down into increments
  - each increment delivering part of the required functionality
- Requirements are prioritized
  - Highest priority requirements are developed in early increments



Source: [https://en.wikipedia.org/wiki/Iterative\\_and\\_incremental\\_development](https://en.wikipedia.org/wiki/Iterative_and_incremental_development)

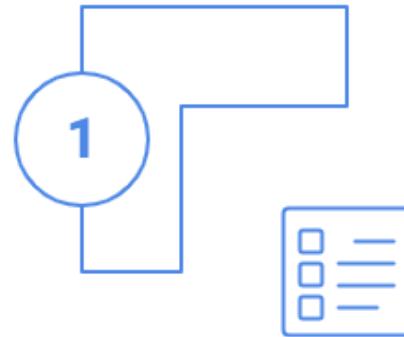
# Incremental development (cont'd)



# Advantages of Incremental Development

## Prioritized testing

Prioritized testing ensures critical requirements are thoroughly evaluated.



## Early prototype feedback

Early prototype feedback significantly shapes high-priority requirements.



## Lower project failure risk

Lower project failure risk is a low-impact, low-priority benefit.



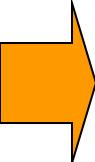
## Early functionality availability

Early functionality availability provides high-impact but low-priority advantages.

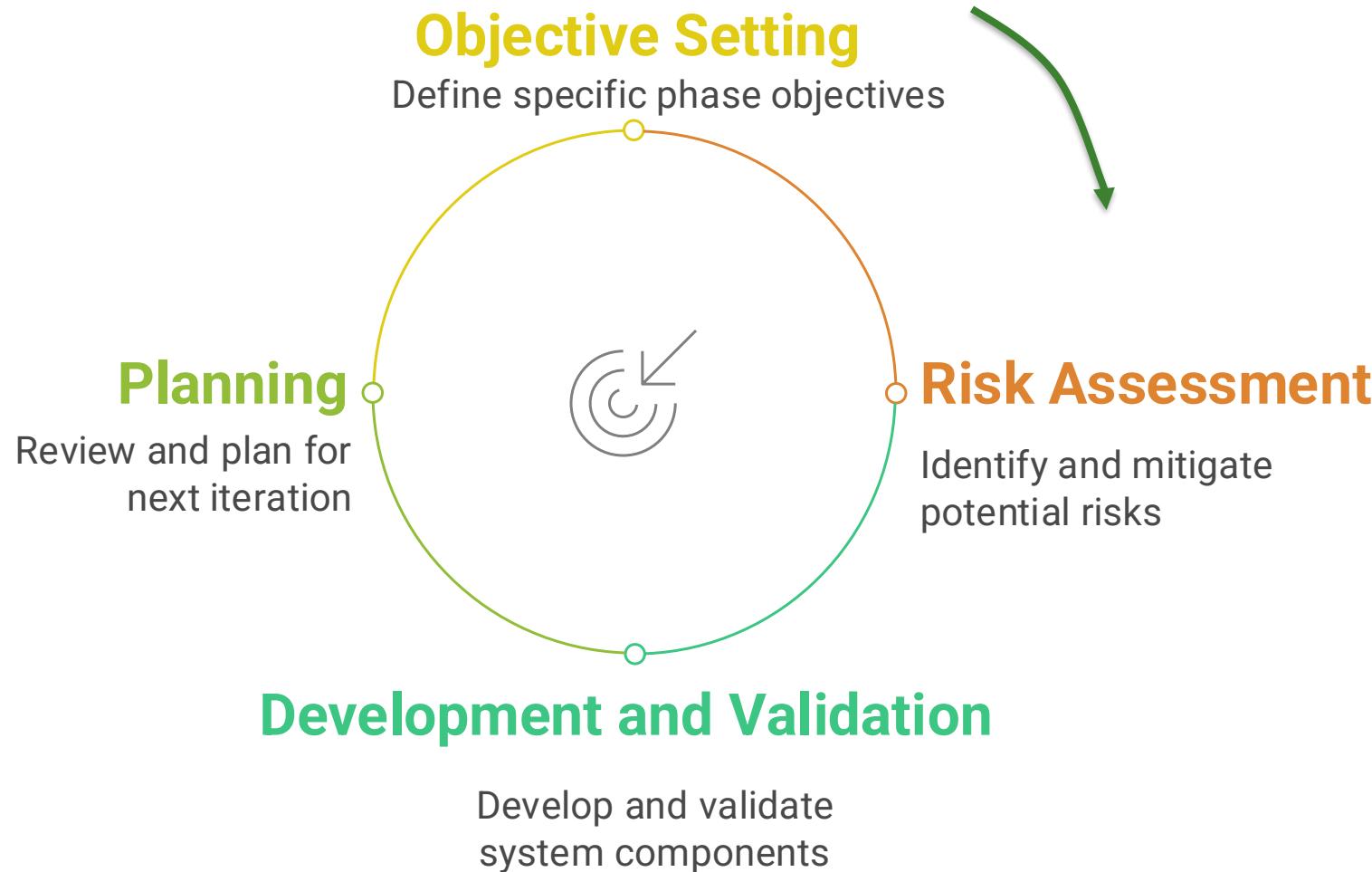


# Outline

- Introduction to Plan-driven and Agile Methods
- Planning Spectrum
- Plan-driven Methods
  - Waterfall Model
  - Spiral Model
  - Rational Unified Process
  - Capability Maturity Model Integration (CMMI)

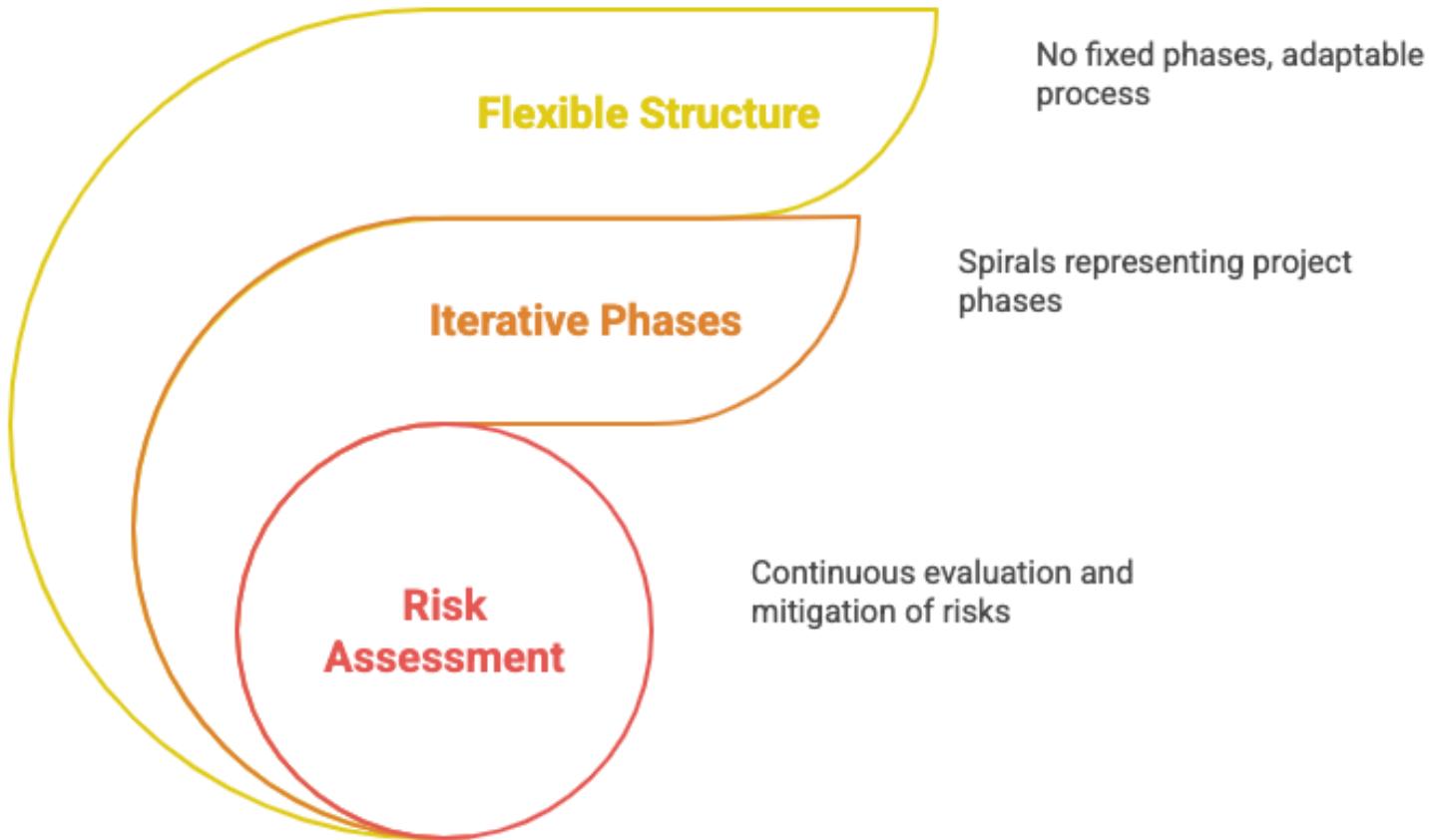


# Spiral Model Sectors

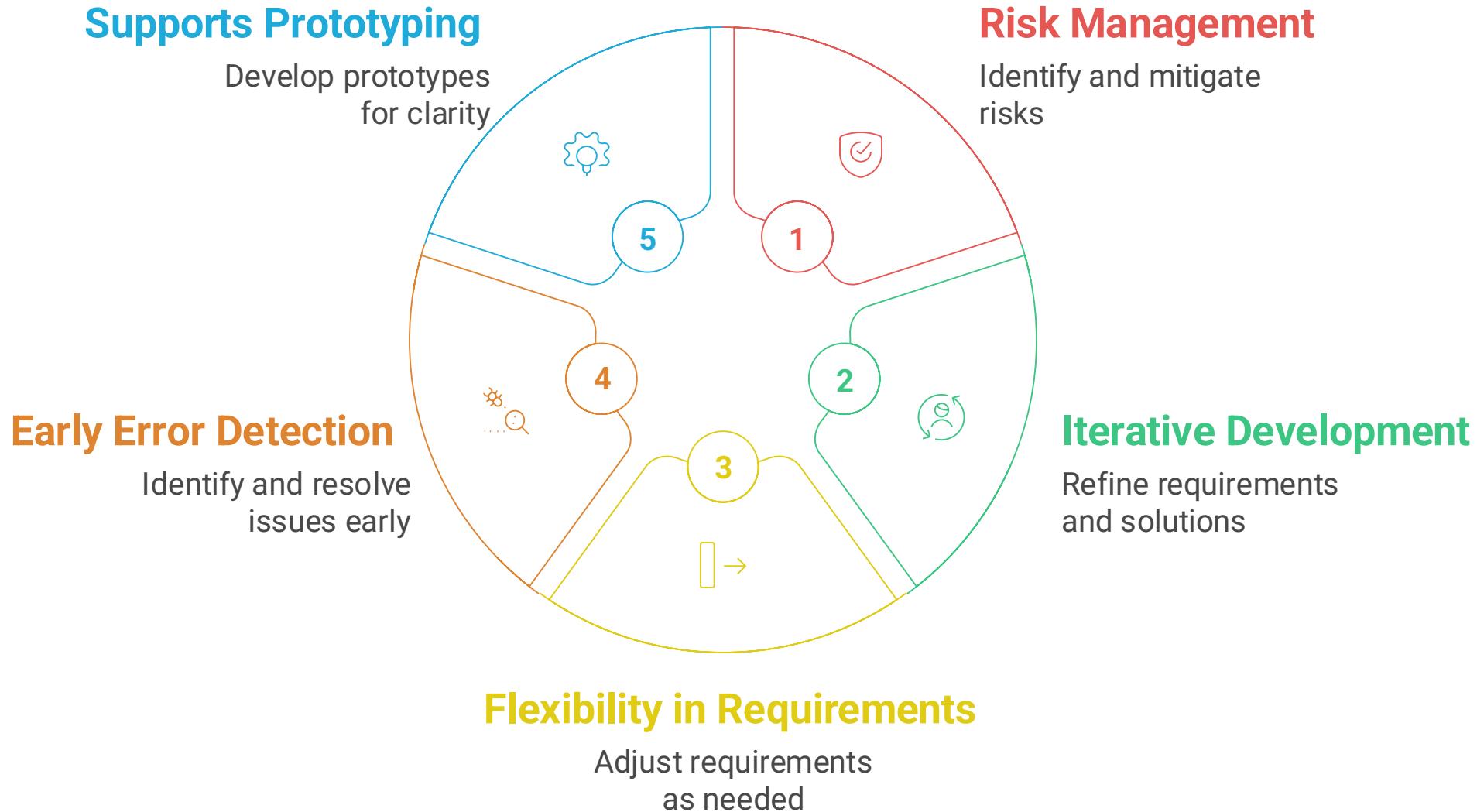


Are there other possible sectors?

# Spiral Model Characteristics

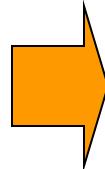


# Advantages of Spiral Model



# Outline

- Introduction to Plan-driven and Agile Methods
- Planning Spectrum
- Plan-driven Methods
  - Waterfall Model
  - Spiral Model
  - Rational Unified Process
  - Capability Maturity Model Integration (CMMI)

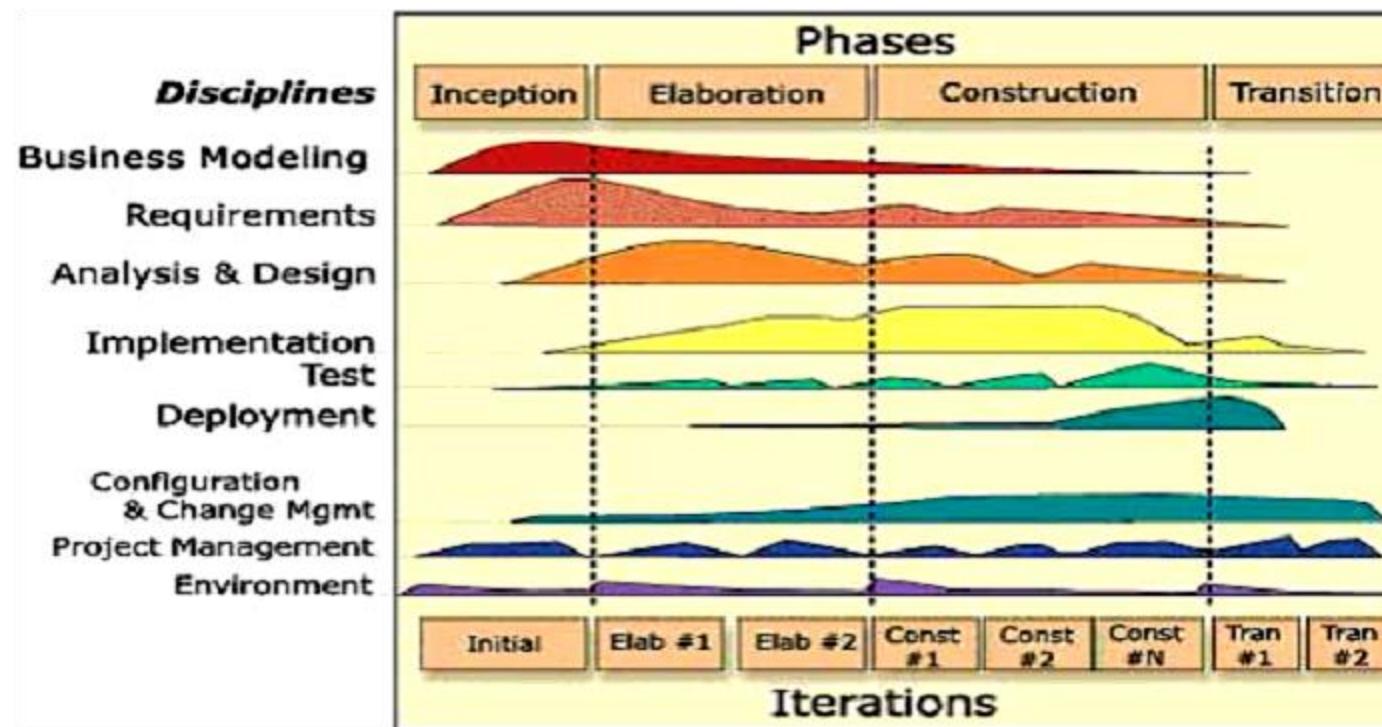


# The Rational Unified Process (RUP)

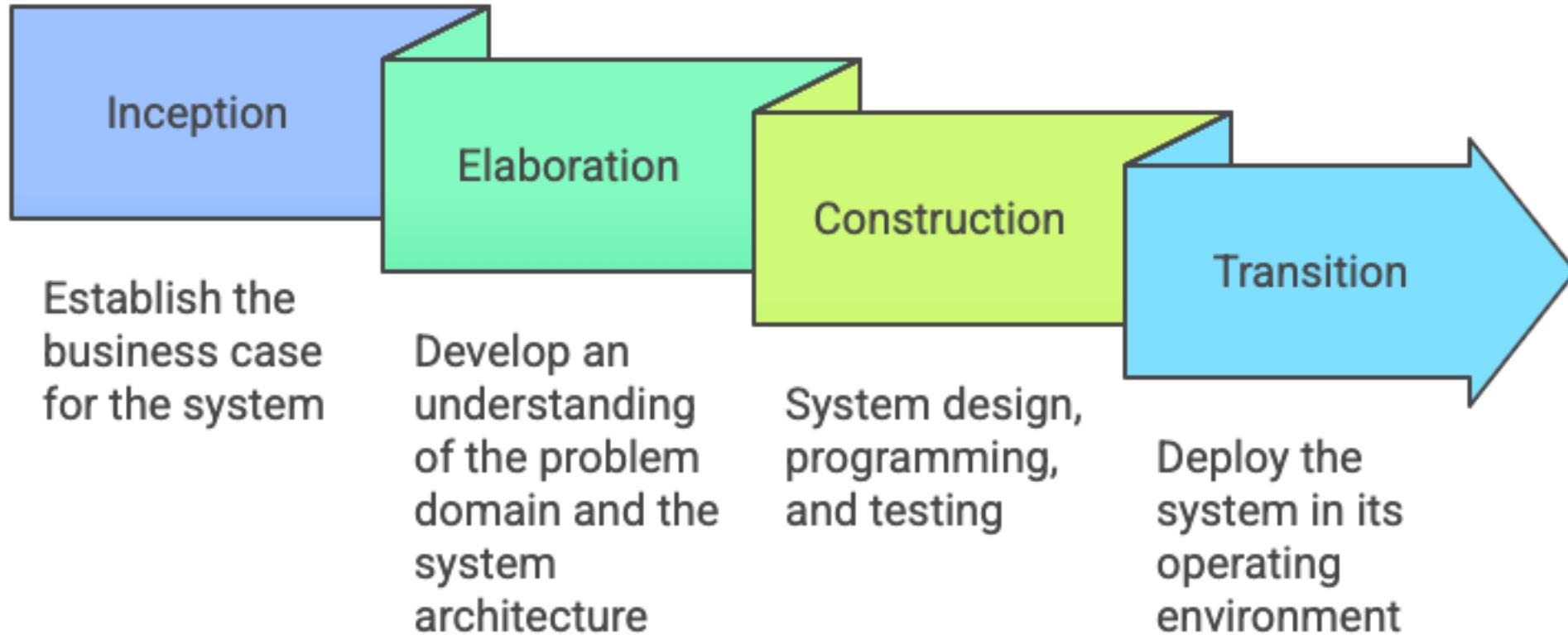
- A popular process model developed and introduced by Rational Software, now IBM Rational
- Normally described from 3 perspectives
  - Dynamic perspective: shows phases over time
  - Static perspective: shows process activities
  - Practical perspective: suggests good practice

# RUP

- Activities grouped into workflows
- Four phases Inception, Elaboration, Construction, and Transition
- Each phase divided into one or more iterations



# RUP Phases



# RUP Iteration

- Each phase is divided into one or more iterations
- Each iteration is considered a waterfall cycle
  - including requirements, analysis & design, implementation, testing, and deployment
- Each iteration has entry and exit criteria or check points
  - Iteration is completed if its check points are satisfied
- Typical number of iterations per phase
  - Inception: 1-2 iterations
  - Elaboration: 1-3
  - Construction: 2-3
  - Transition: 2-3

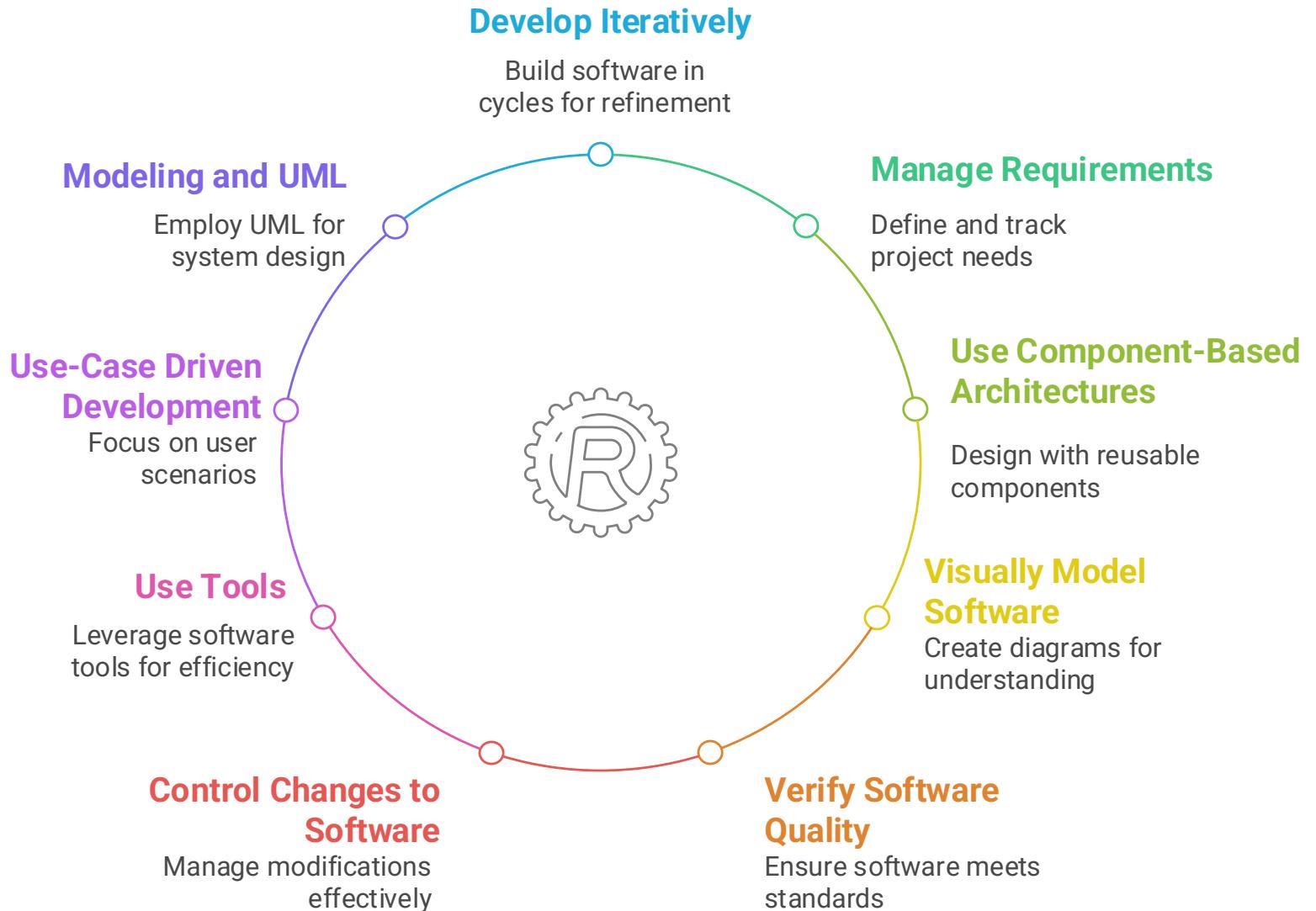
# RUP Workflows

- Business modeling
  - The business processes are modeled using business use cases
- Requirements
  - Requirements specification, use-case model
- Analysis and Design
  - Software architecture and design are created
  - Different software views (use-case view, design view, process view, implementation view, deployment view)
- Implementation
  - Software components are implemented

# RUP Workflows (cont'd)

- Testing
  - Includes testing activities performed to verify and validate the product against the requirements
- Deployment
  - Activities to release and deploy the product and help users to use the product successfully
- Configuration and Change Management (SCM)
  - Manages changes to the software product
- Project Management
  - Manages software development
- Environment
  - Prepares and ensures software tools, processes, and hardware for the development team

# RUP Best Practices



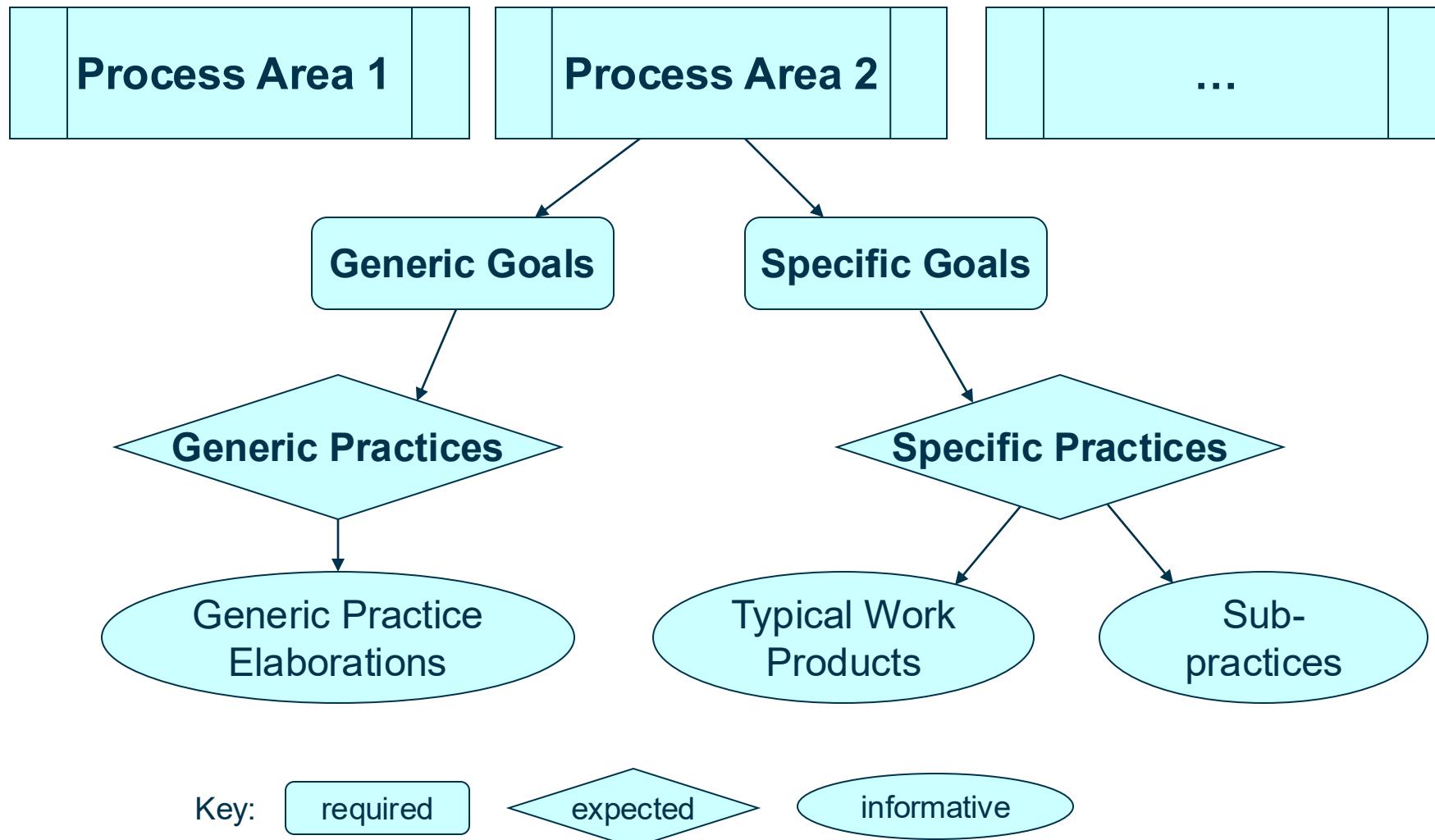
# Outline

- Introduction to Plan-driven and Agile Methods
- Planning Spectrum
- Plan-driven Methods
  - Waterfall Model
  - Spiral Model
  - Rational Unified Process
  - Capability Maturity Model Integration (CMMI)

# CMMI

- CMMI – Capability Maturity Model Integration
- Developed and promoted by CMU's Software Engineering Institute (SEI)
- **CMMI is a framework for software and systems process maturity**
- Allowing to support current and future software development trends, e.g.,
  - everything connected
  - rapid requirements change
  - adaptive process models
  - requirements, architecture, development are done concurrently
  - collaborative

# CMMI Model Components



# CMMI Model Components (cont'd)

## ■ Process Area

- ❑ A cluster of related practices in an area that satisfy a set of goals

## ■ Goal

- ❑ **Generic goal**: a required model component that describes the characteristics that must be present
- ❑ **Specific goal**: a required model component that describes the unique characteristics that must be present to satisfy the process area

# CMMI Model Components (cont'd)

## ■ Practice

- **Generic practice:** an expected model component that is considered important in achieving the associated generic goal
- **Specific practice:** an expected model component that is considered important in achieving the associated specific goal

## ■ Work product

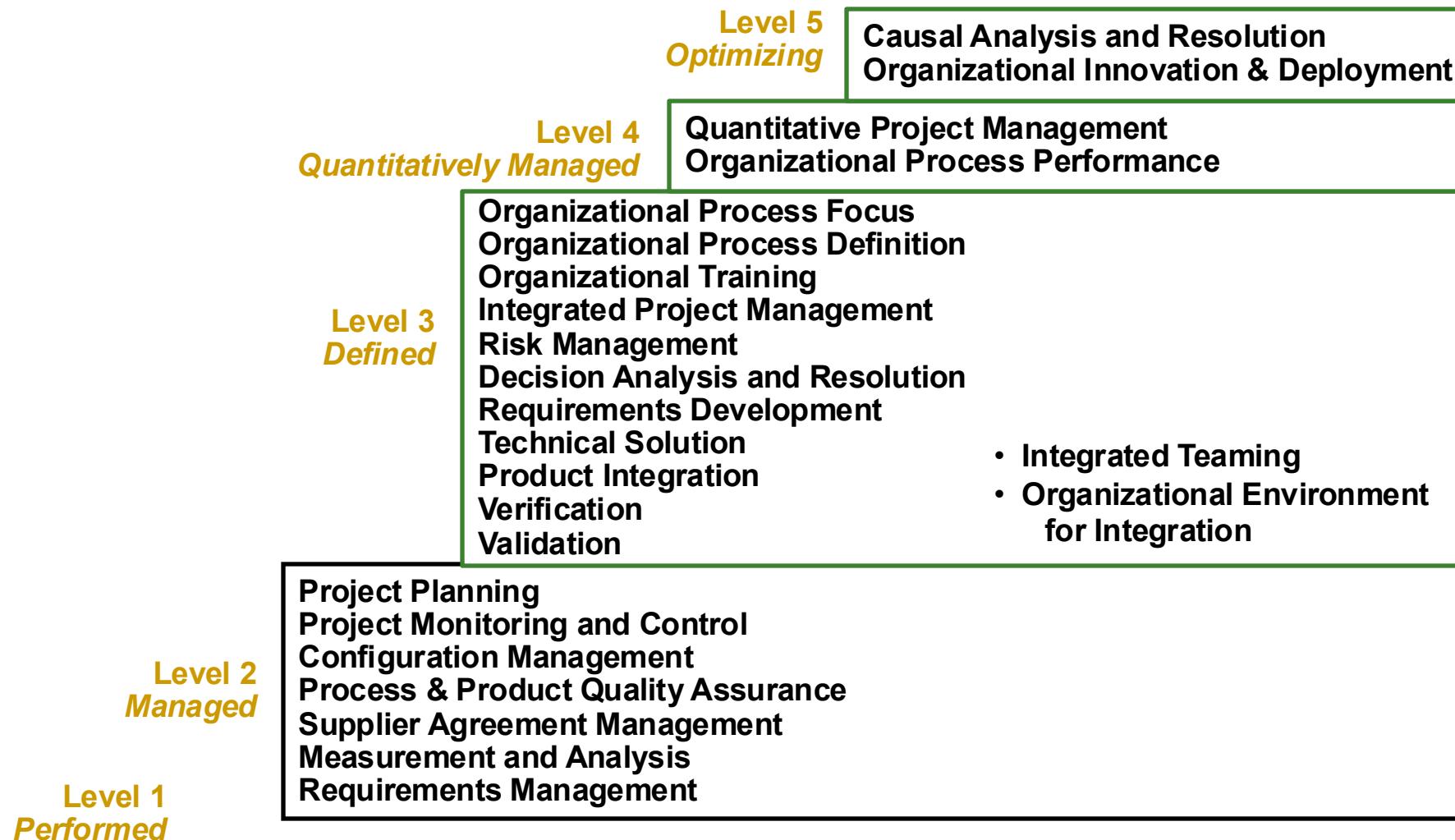
- Any artifact produced by a process

## ■ Sub-practice

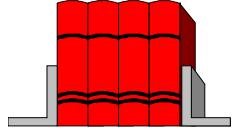
- Providing guidance for interpreting and implementing specific or generic practices

# CMMI Process Areas

## *Staged Representation*



# Common Features (for all Process Areas)



- **Commitment to Perform** includes practices that ensure the process is established and will endure.
  - Organizational policies



- **Ability to Perform** includes practices that establish the necessary conditions for implementing the process completely.

- Plans, resources, responsibility, and training.



- **Activities Performed** includes practices that directly implement a process area.



- **Directing Implementation** includes measurement practices that are necessary to collect and analyze data related to the process.

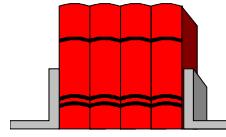
- Configuration management, stakeholder involvement, monitor and control



- **Verification** includes practices that ensure compliance with the process that has been established.

- Senior management reviews, quality audits

# Organizational Assets (for all Process Areas)



- Organizational policies



- Organizational training office



- Organizational procedures

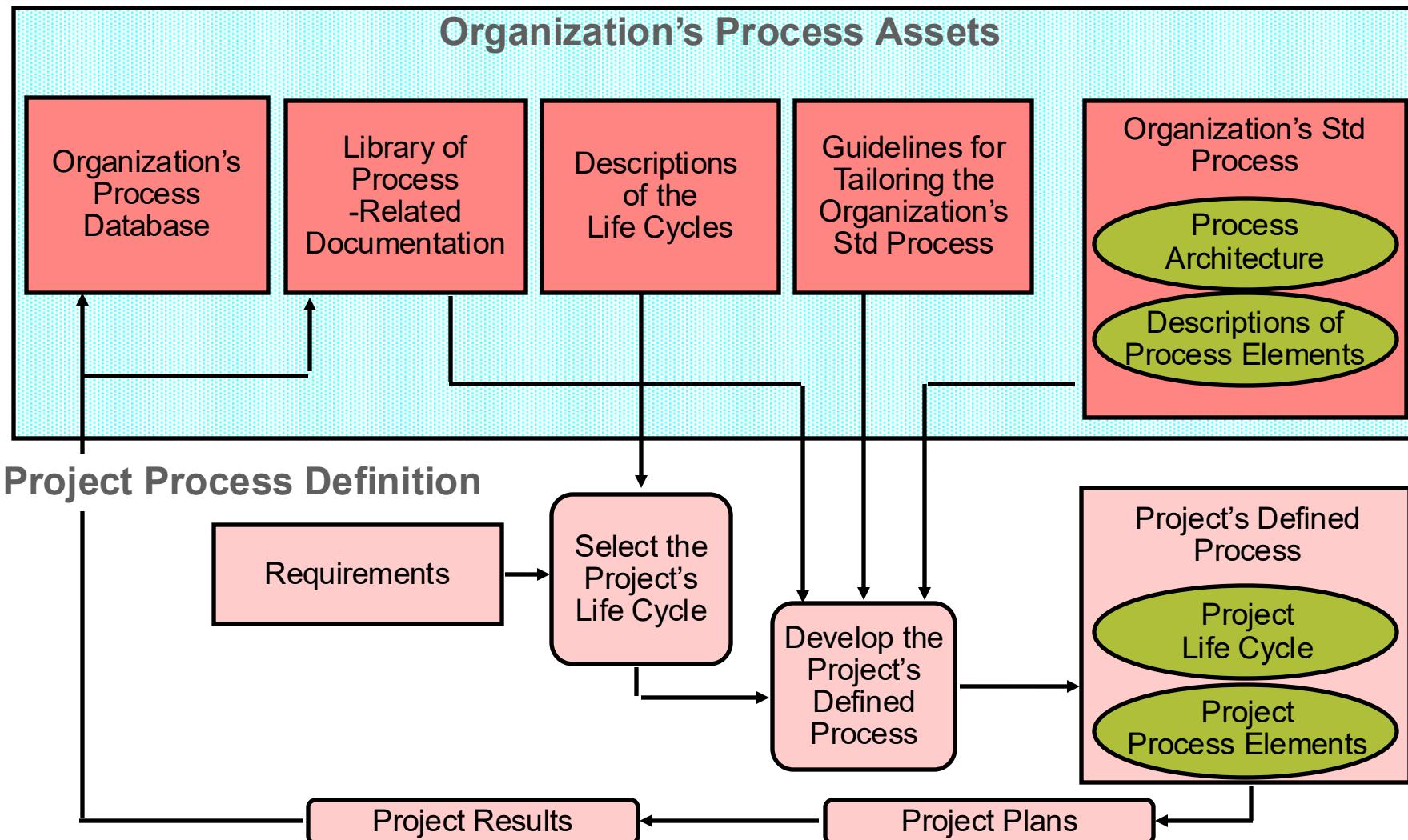


- Standard metrics
- Estimation process, tools & support



- Senior management review process
- Independent quality assurance organization

# Conceptual Software Process Framework



# Summary

- Two Software Development Approaches
  - Plan-driven
  - Agile
- Each has strengths and weaknesses
- Plan-driven methods
  - Waterfall
  - Spiral
  - Rational Unified Process
  - CMMI

# Summary (cont'd)

- Strengths of plan-driven methods
  - High assurance, suitable for safety-critical software systems
  - Suitable for large software systems
  - Documentation available for training new staff
  - Making sure everyone to work in certain ways through clear processes
- Weaknesses of plan-driven methods
  - Documentation overhead
  - Outdated and useless documents
  - Inflexible to changes
  - Obstruct innovations
  - Engineers don't like documentation