Informatik-Projektentwicklung
– Lecture 2 –

Prof. Dr. Peter Müller

Software Component Technology

Wintersemester 04/05

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

What is a Project?

Every project has a definite beginning and a definite end

- Definition:
  A project is a <u>temporary</u> endeavor undertaken to create a <u>unique</u> product or service

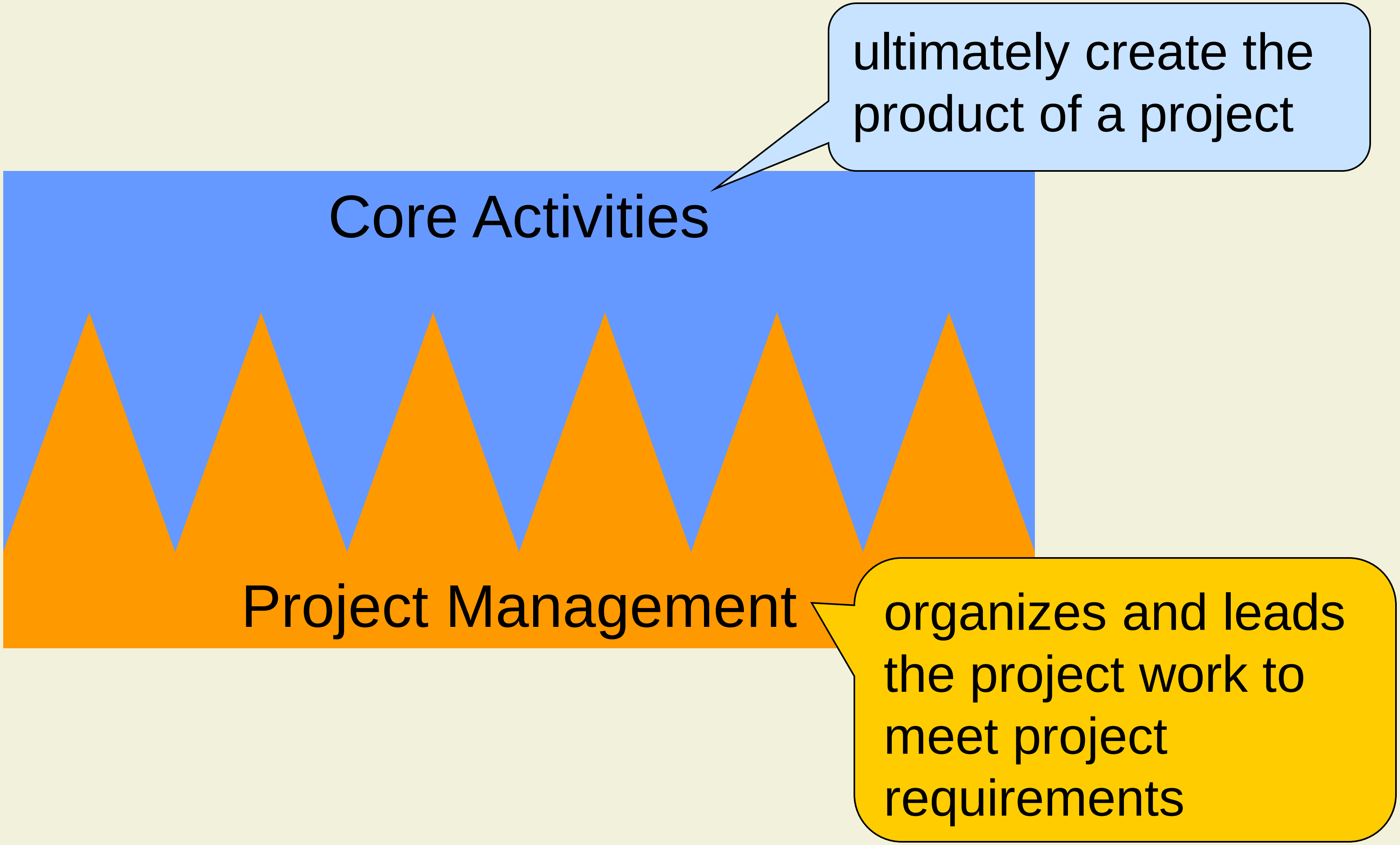
The product or service is different in some distinguishing way from all similar products and services

- In contrast: *Operations* are ongoing and repetitive

Characteristics of Projects

- **Temporary** endeavor
- **Unique** product or service
- Performed by **people**
- **Constrained** by limited resources
  - Budget, time, staff
- **Planned**, **executed**, and **controlled**
- Have their own **organization**

Core Activities and Project Management



ultimately create the product of a project

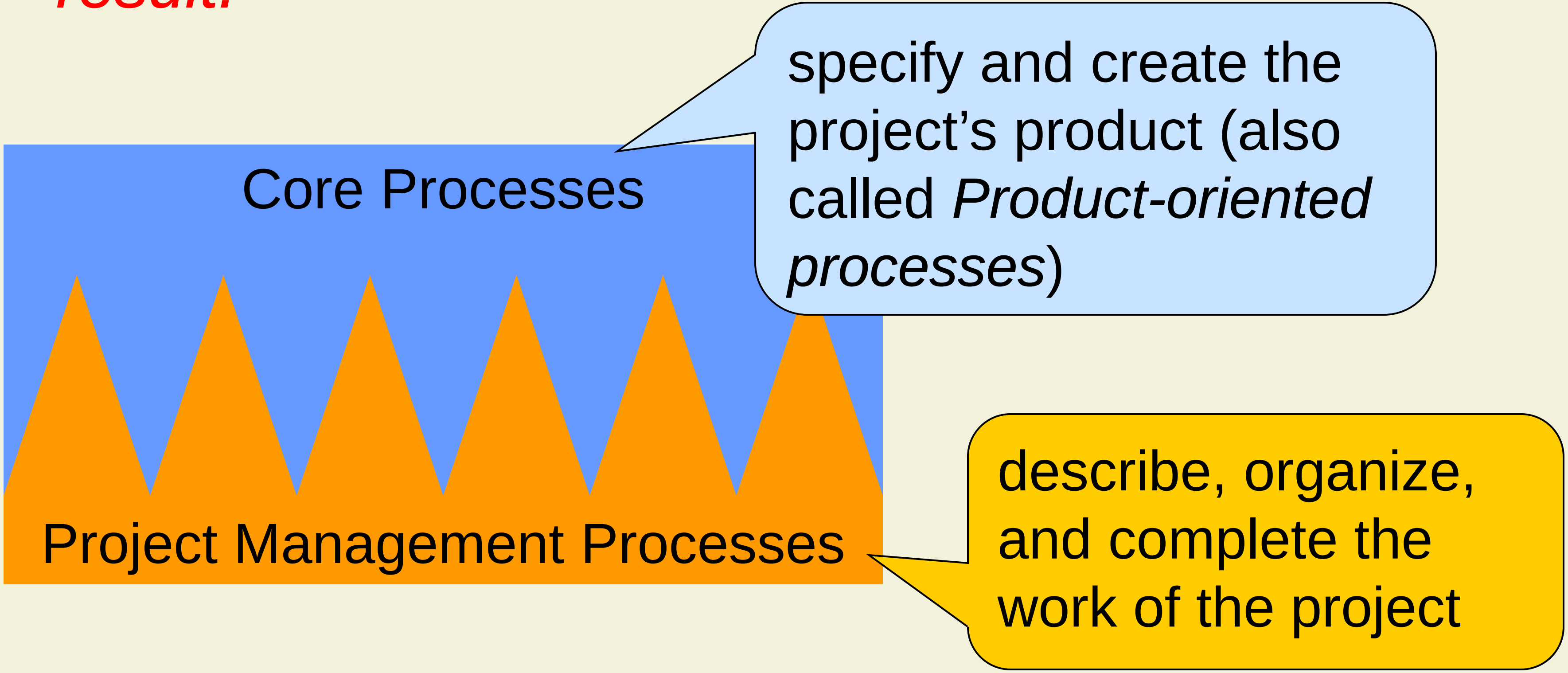
Core Activities

Project Management

organizes and leads the project work to meet project requirements

Processes

- Definition of Process:
  A process is a series of actions bringing about a result.



Core Processes

specify and create the project's product (also called *Product-oriented processes*)

Project Management Processes

describe, organize, and complete the work of the project

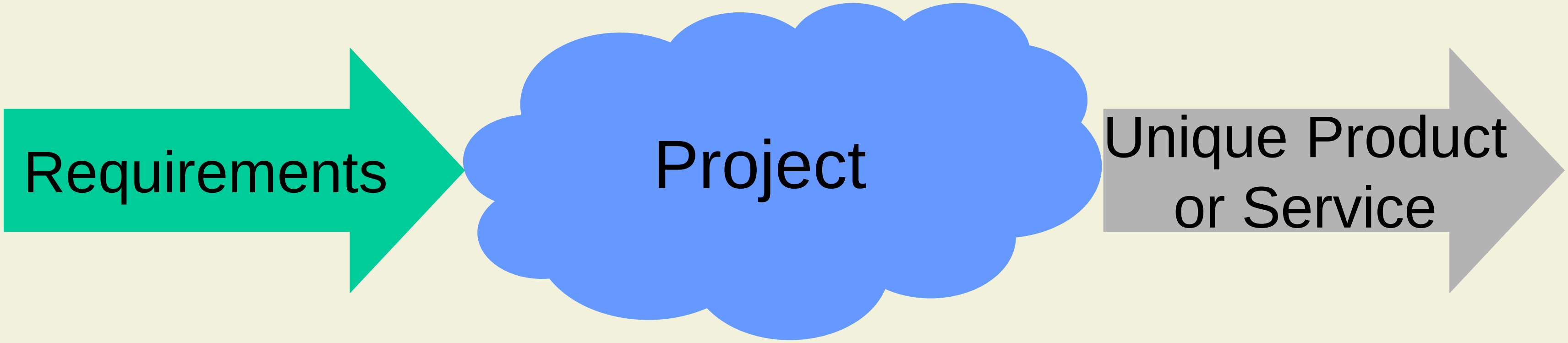
Agenda for Today

2. Project Life Cycle and Project Management Life Cycle
    2.1 Project Life Cycle
    2.2 Project Management Life Cycle
    2.3 Development Models

2. Project Life Cycle and Project Management Life Cycle

- **2.1 Project Life Cycle**
- 2.2 Project Management Life Cycle
- 2.3 Development Models

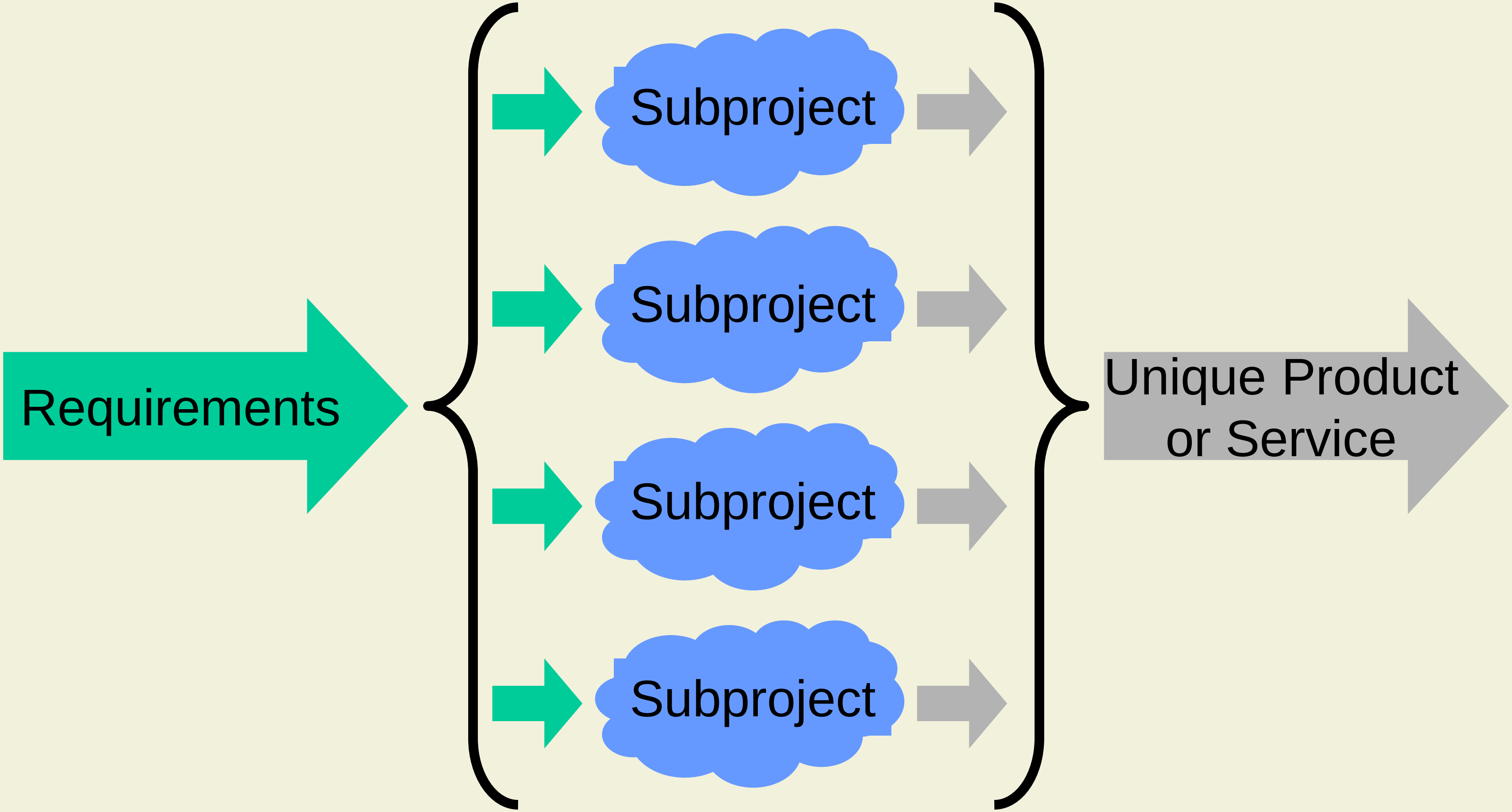
Projects are Complex



- At project start, only broad information about characteristics of product are available
- Average size of IT projects is 500-2000 person days
- Different tasks have to be performed such as designing a GUI, testing a module, installing hardware, training users, or negotiating with customers

➔ How can we handle this complexity?

Decomposition According to Product



Subprojects

- Decomposition usually follows structure of product
- Subprojects are **easier to manage**
- Subprojects enable one to use **specialized staff**
- Remaining and new problems
  - Only broad information about product characteristics
  - Managing the interfaces between subprojects
  - Integrating the results of the subprojects
  - Increased need for communication
- Subprojects are **still complex**

Progressive Elaboration

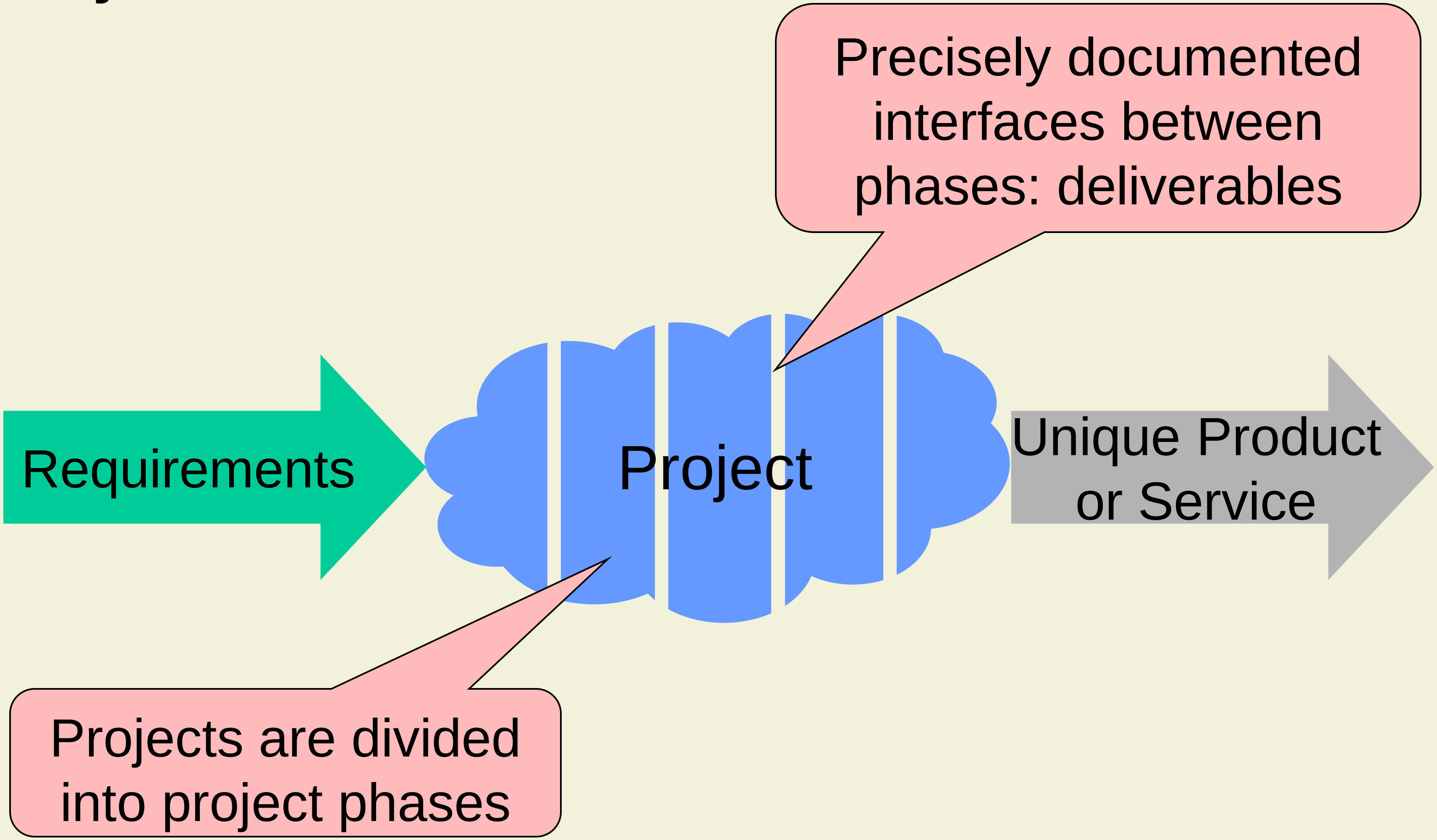
Characteristics of a unique product or service must be progressively elaborated

Continuing steadily by increments

Worked out with care and detail

- During the project, characteristics are defined in more detail as the project team develops a better and more complete understanding of the product

Project Phases



Precisely documented interfaces between phases: deliverables

Requirements

Project

Unique Product or Service

Projects are divided into project phases

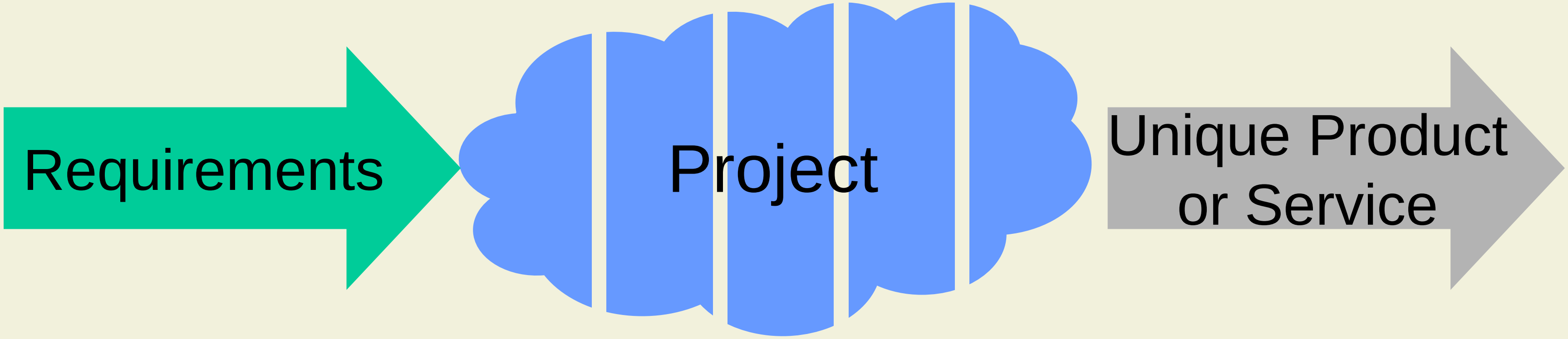
Deliverables

- Definition:
  Any measurable, tangible, verifiable outcome, result, or item that must be produced to complete a project or part of a project

- Examples
  - An object-oriented design, described by a UML diagram
  - A project schedule as MS Project file
  - A user guide for a new application
  - Software, delivered as compiled binary

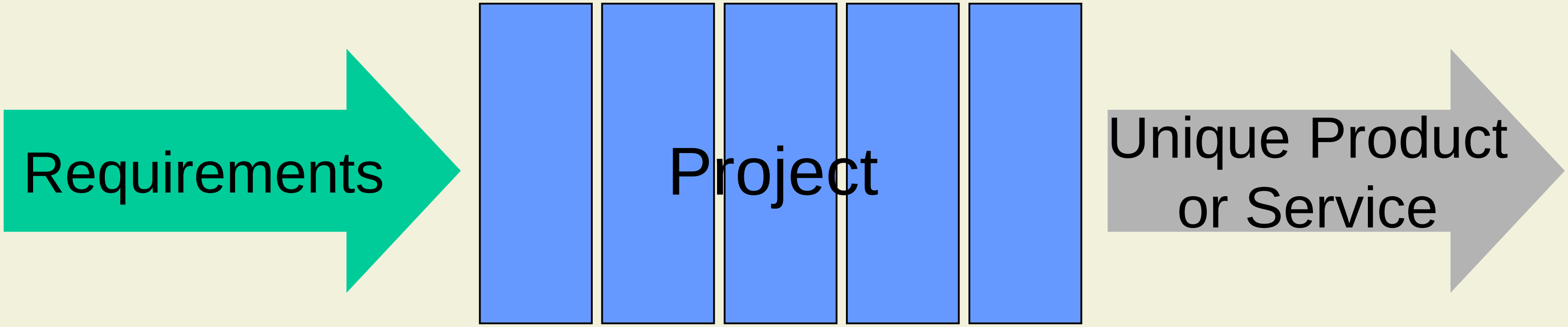
Project Phases

- Definition:
  A collection of logically related project activities, usually culminating in the completion of a major deliverable

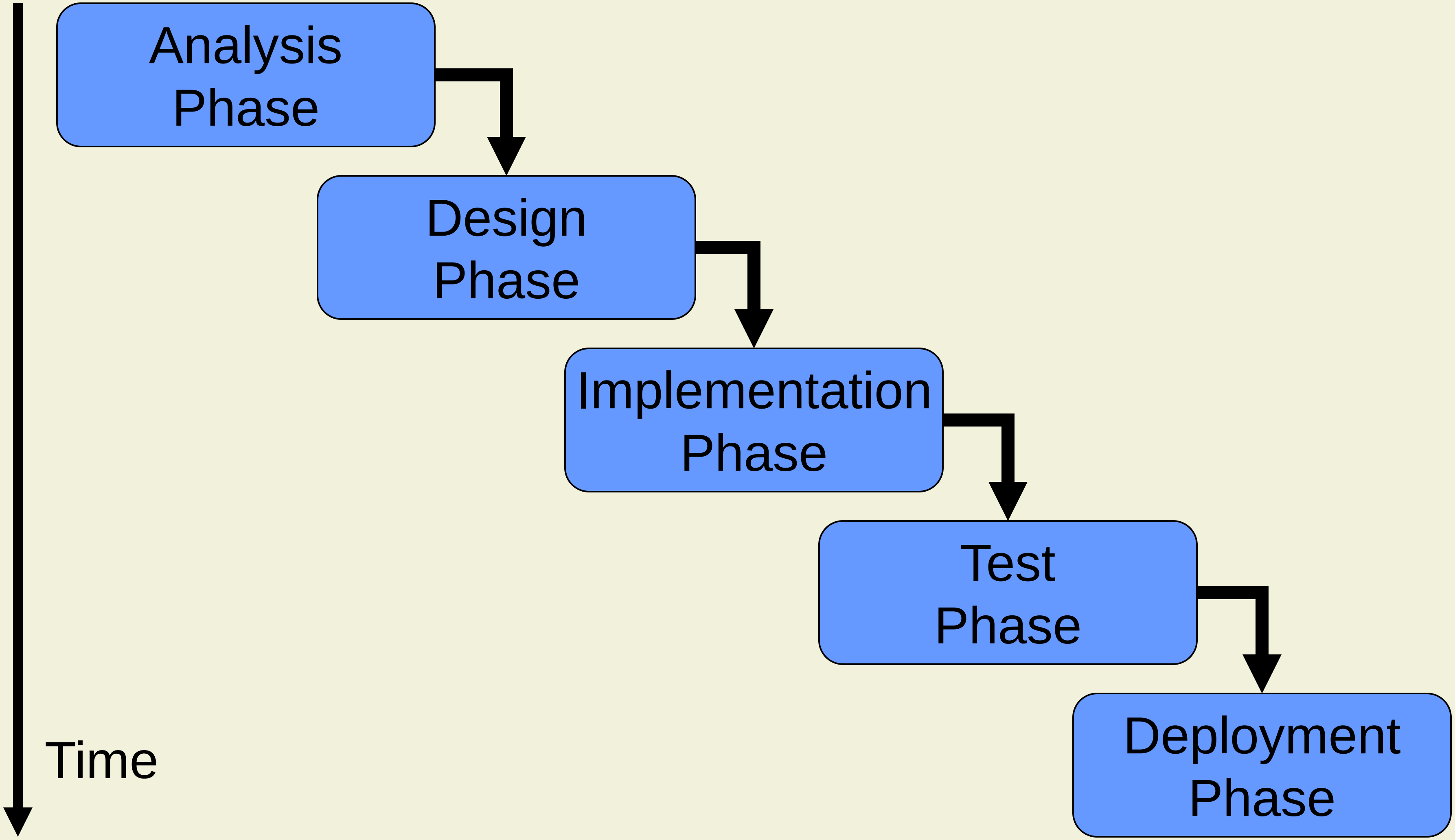


Project Phases

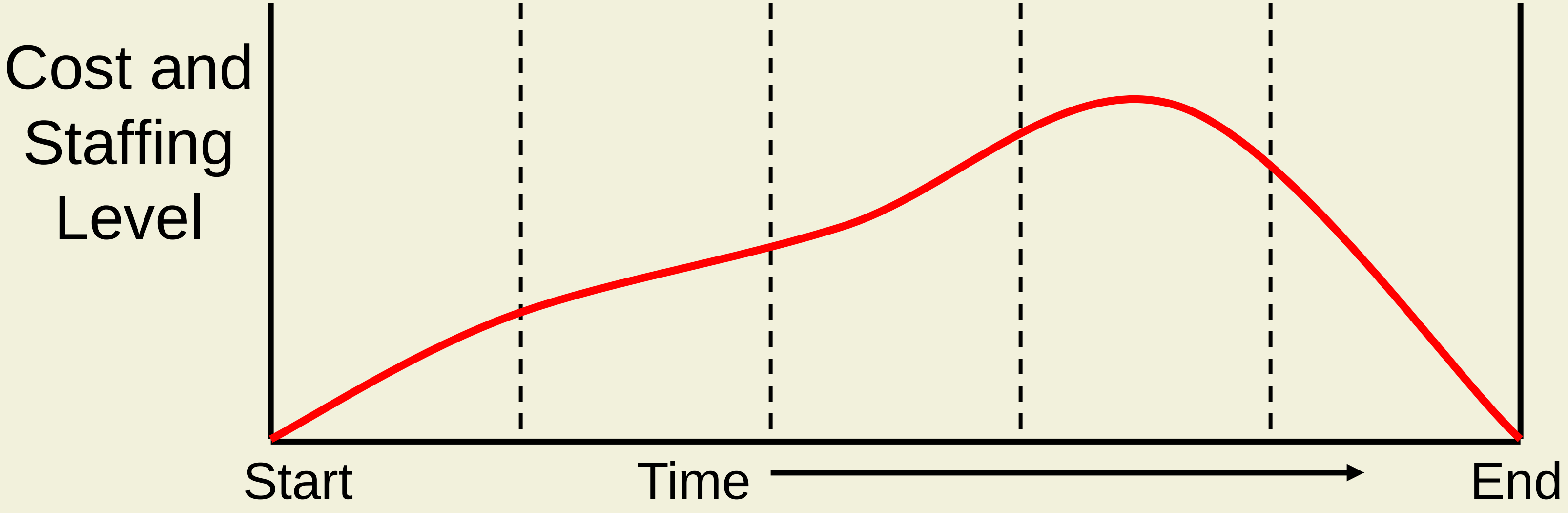
- Definition:
  A collection of logically related project activities, usually culminating in the completion of a major deliverable



Waterfall Model of Project Life Cycle

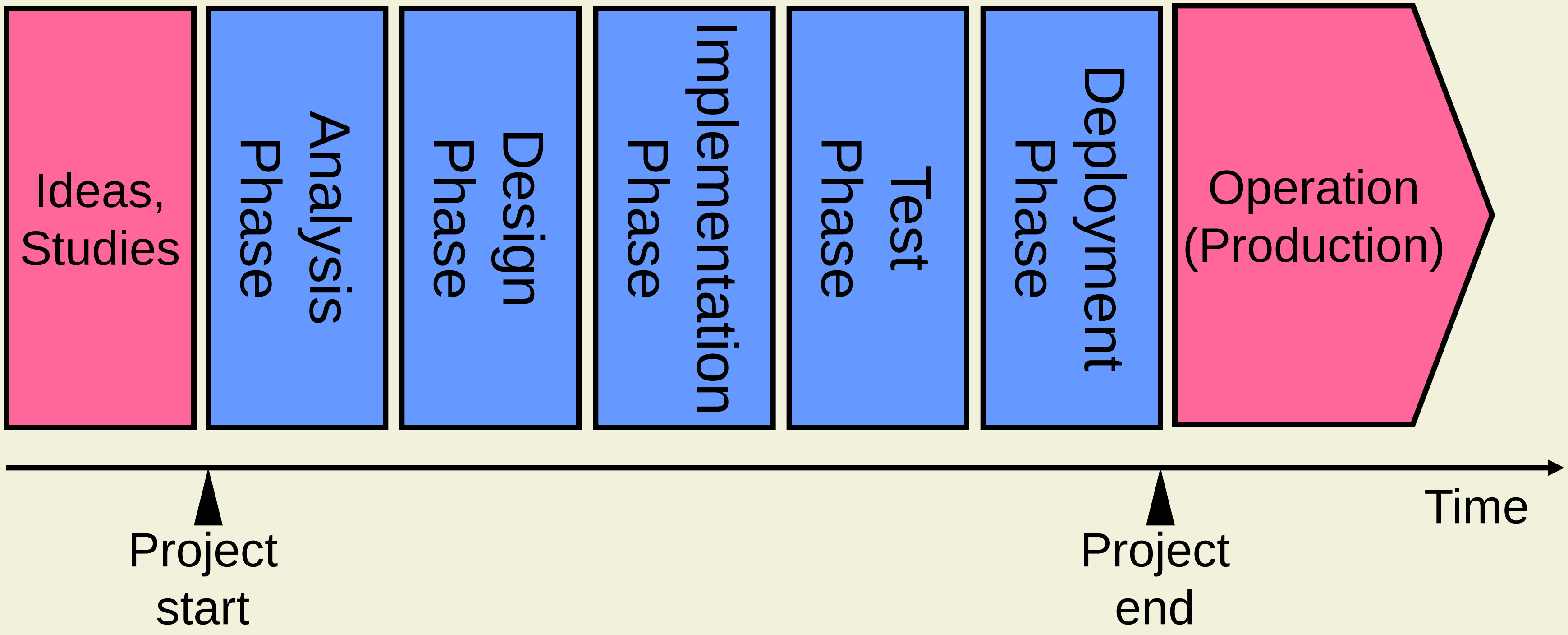


Properties of the Project Life Cycle



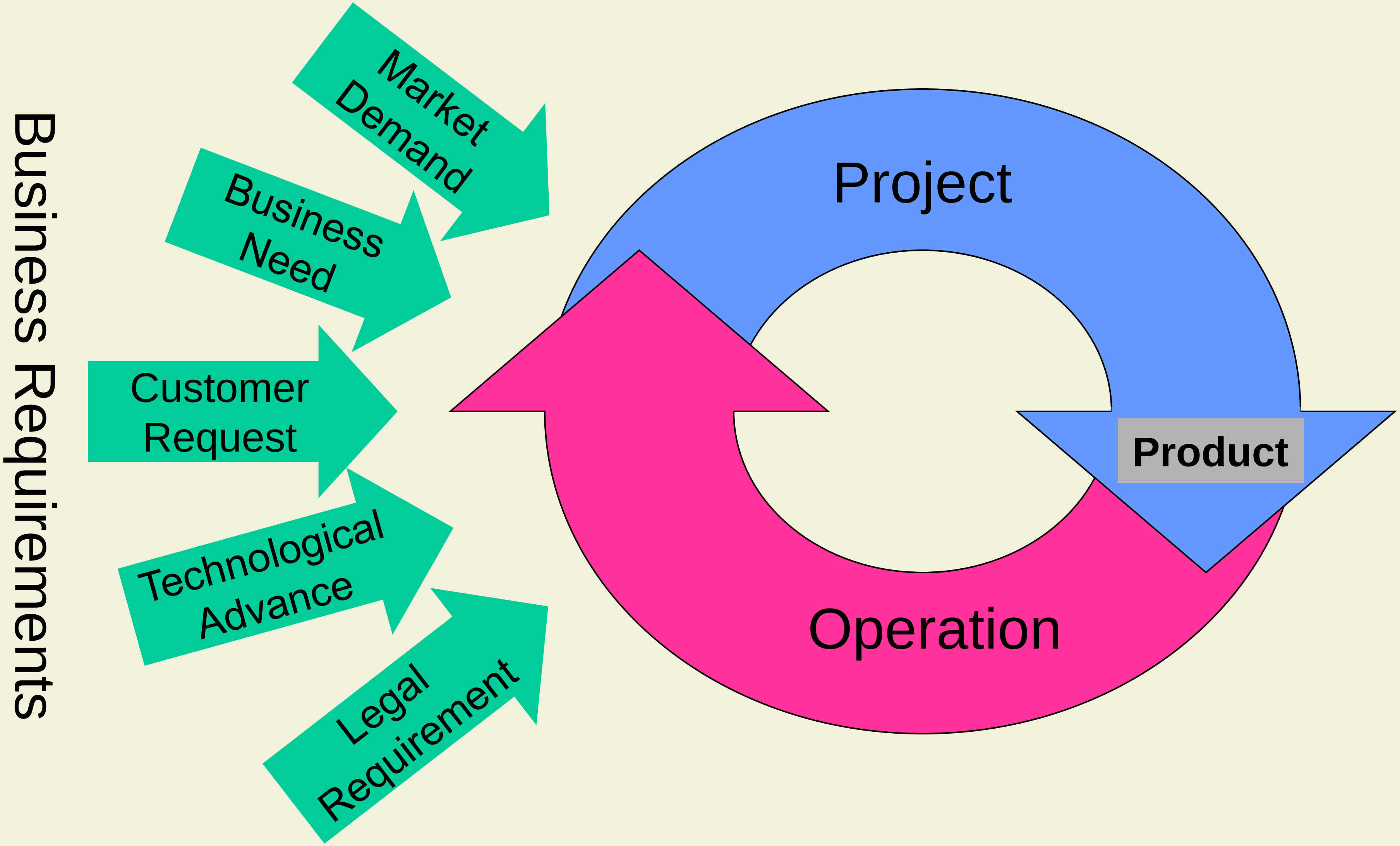
- Stakeholders’ influence on product characteristics and final cost is highest at project start and decreases progressively
- Cost of changes and error correction increases during the project life cycle

From Projects to Operations



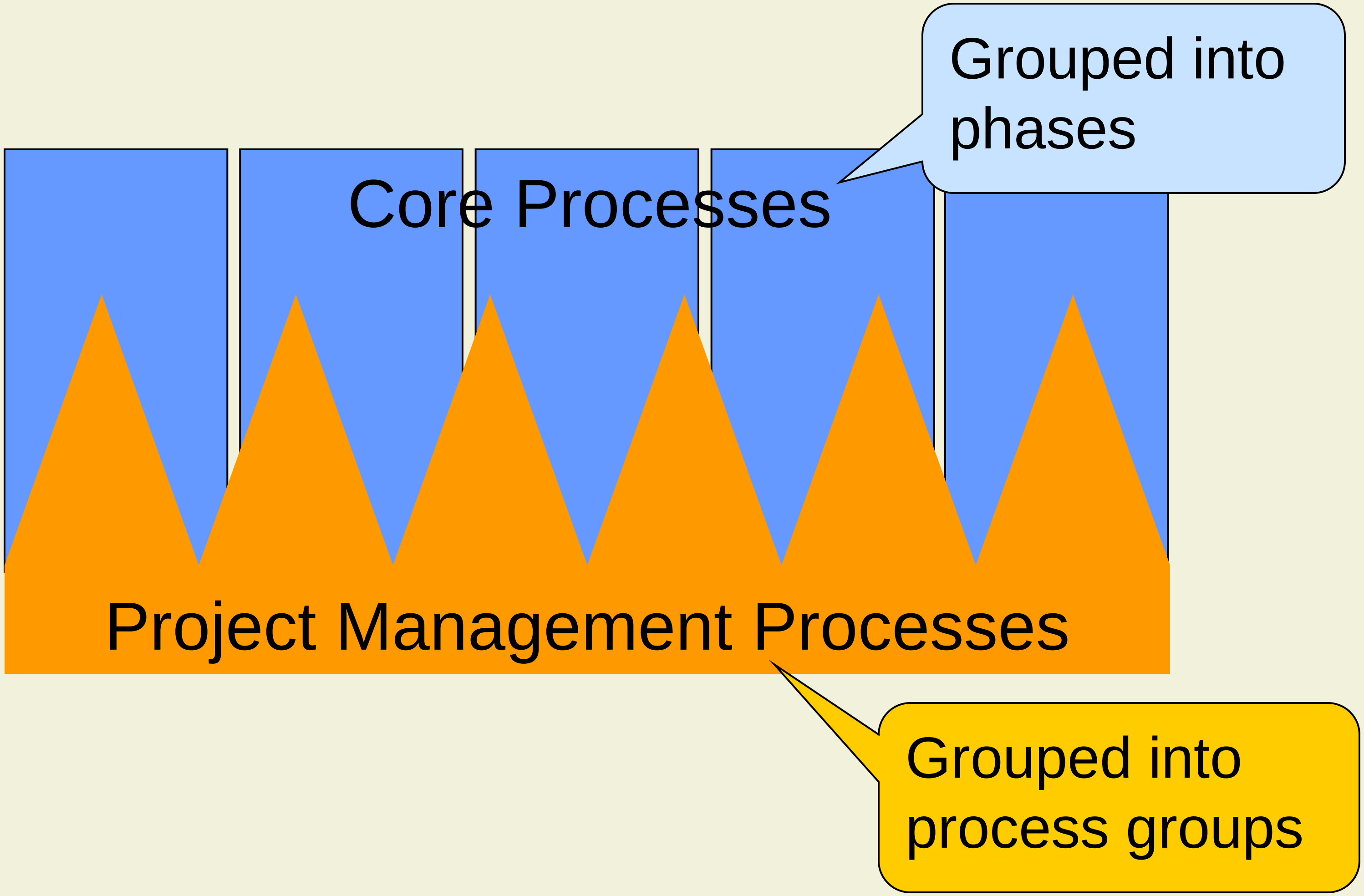
- Project phases are surrounded by related activities that are not part of the project

Product Life Cycle



2. Project Life Cycle and Project Management Life Cycle

Core and Project Management Processes



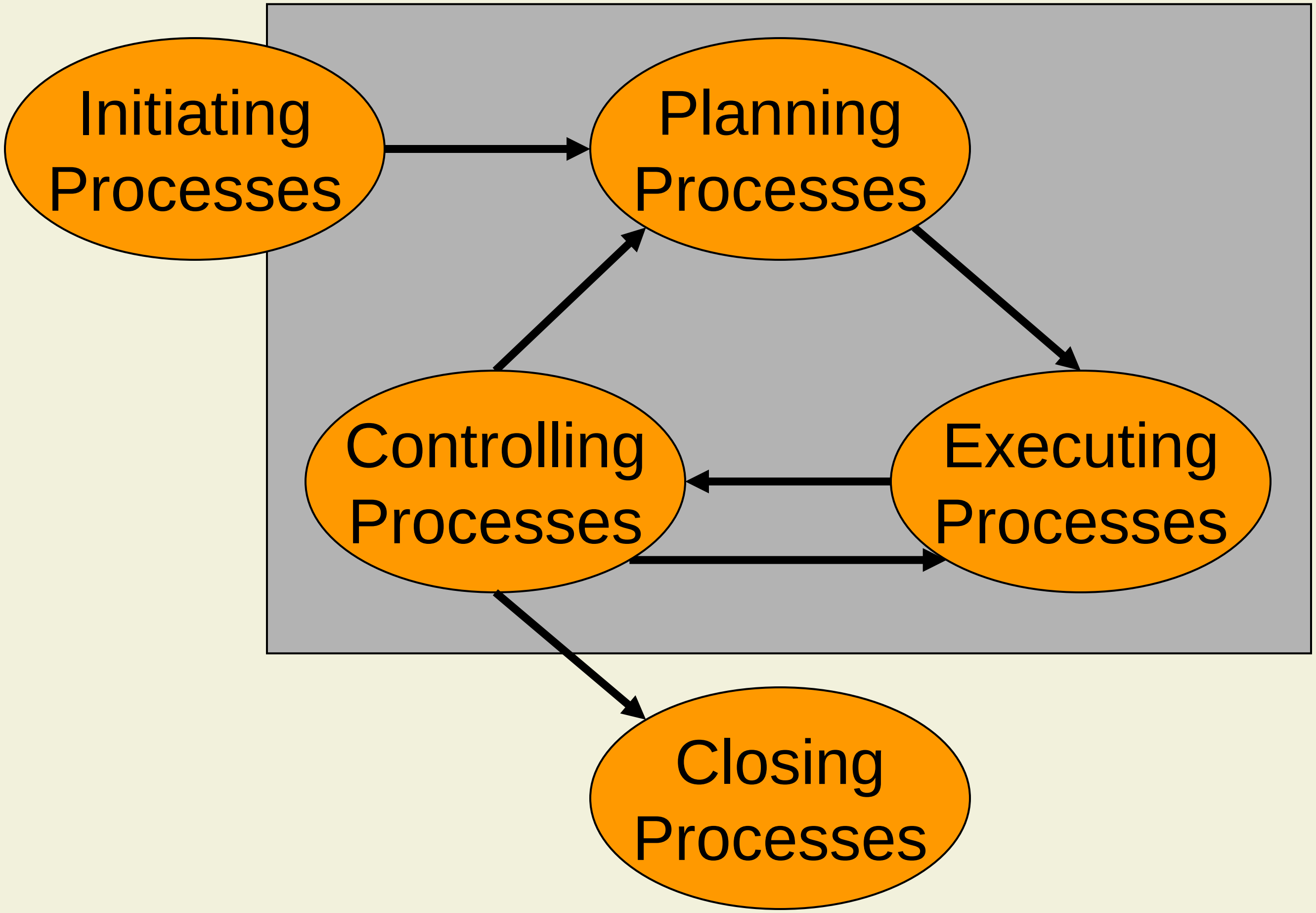
Grouped into phases

Core Processes

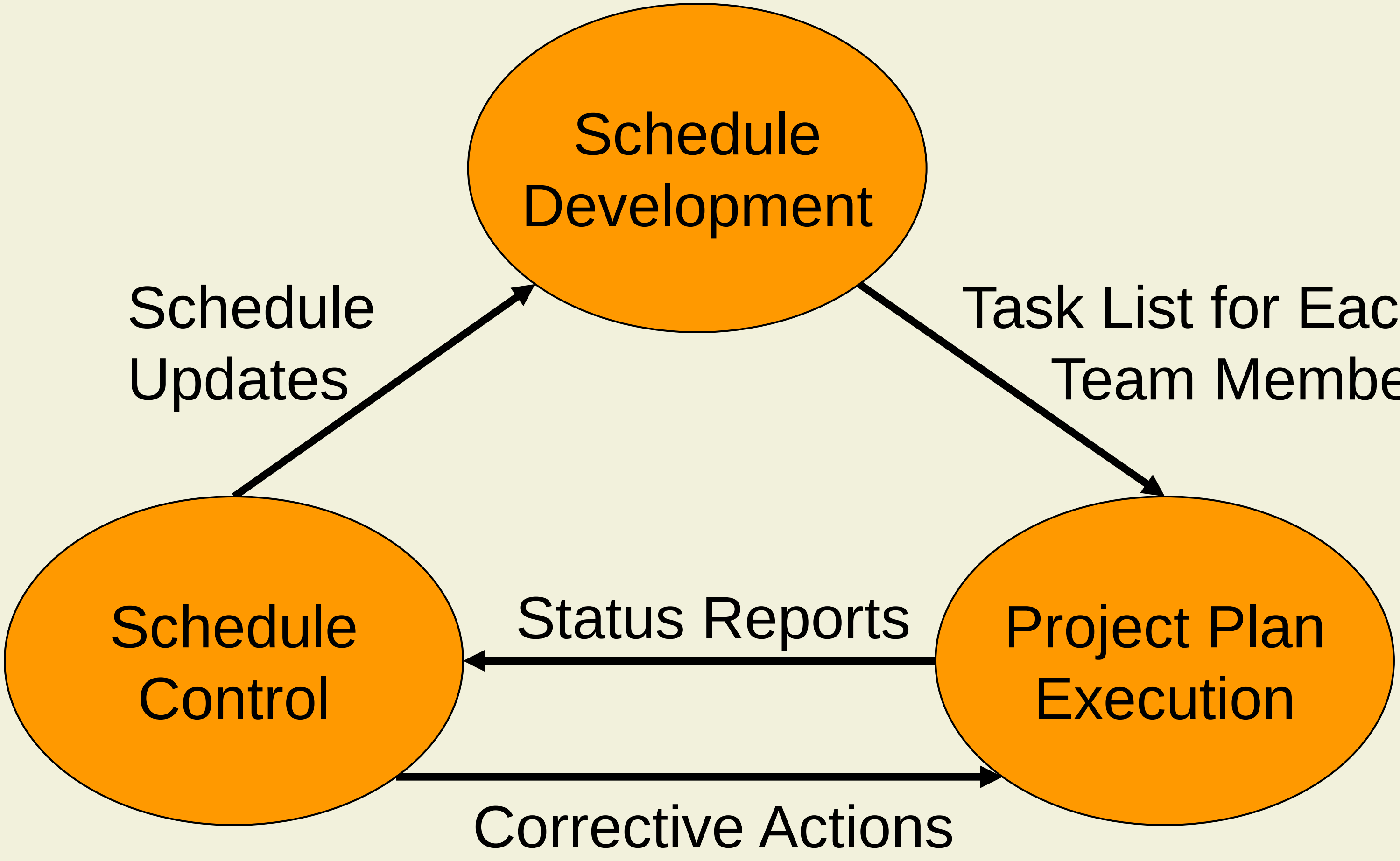
Project Management Processes

Grouped into process groups

Project Management Life Cycle



Example: Time Management



Schedule Development

Task List for Each Team Member

Project Plan Execution

Status Reports

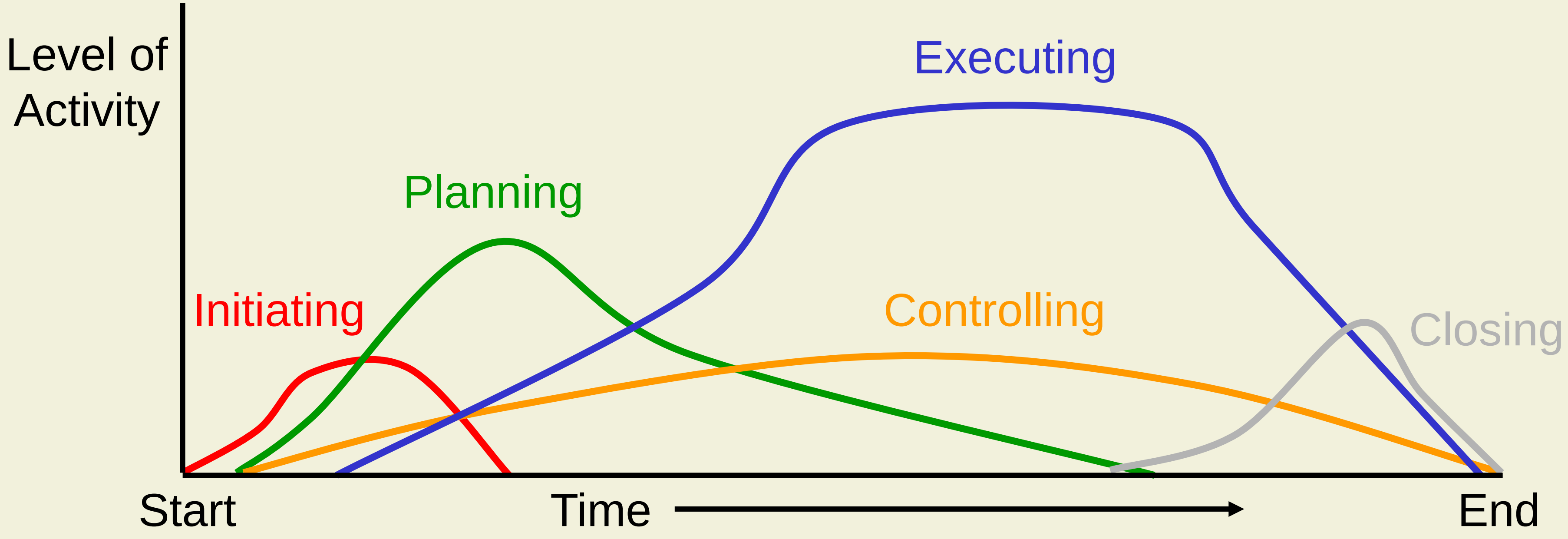
Schedule Control

Corrective Actions

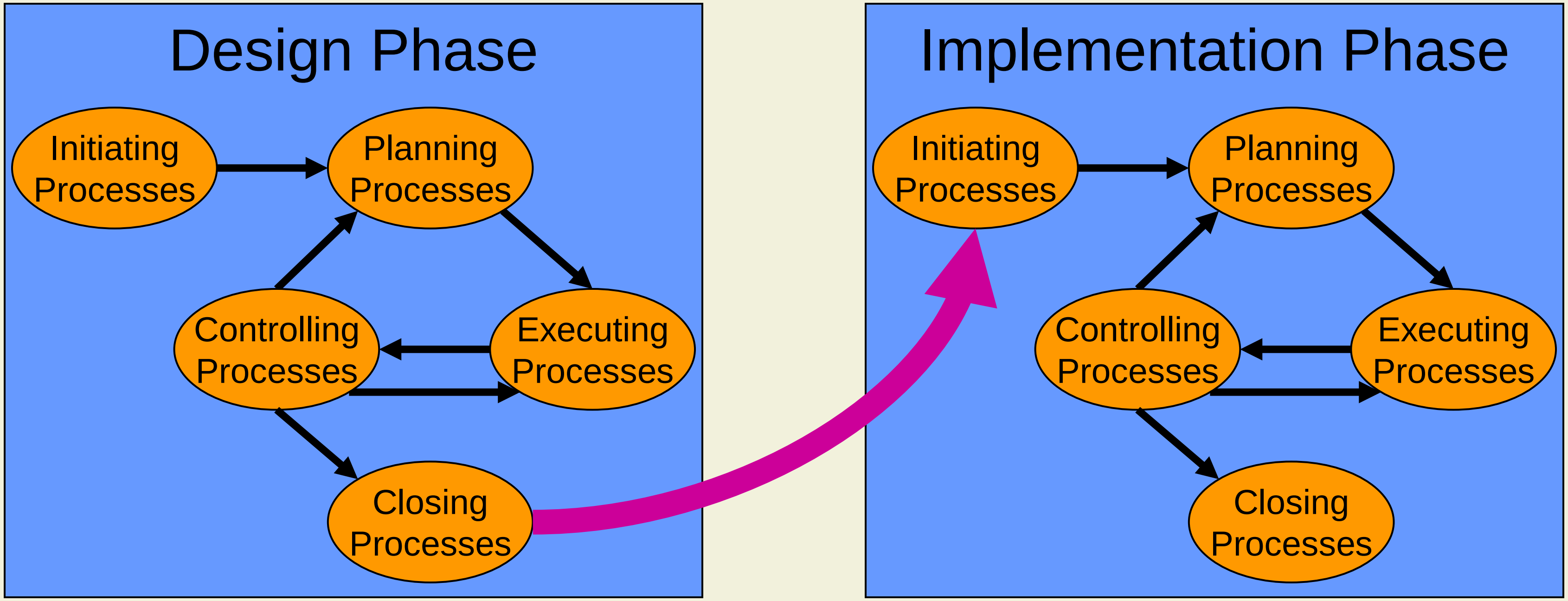
Schedule Updates

Process Groups

- Project groups are not discrete one-time events
- They overlap and occur at varying levels of intensity within each phase of the project

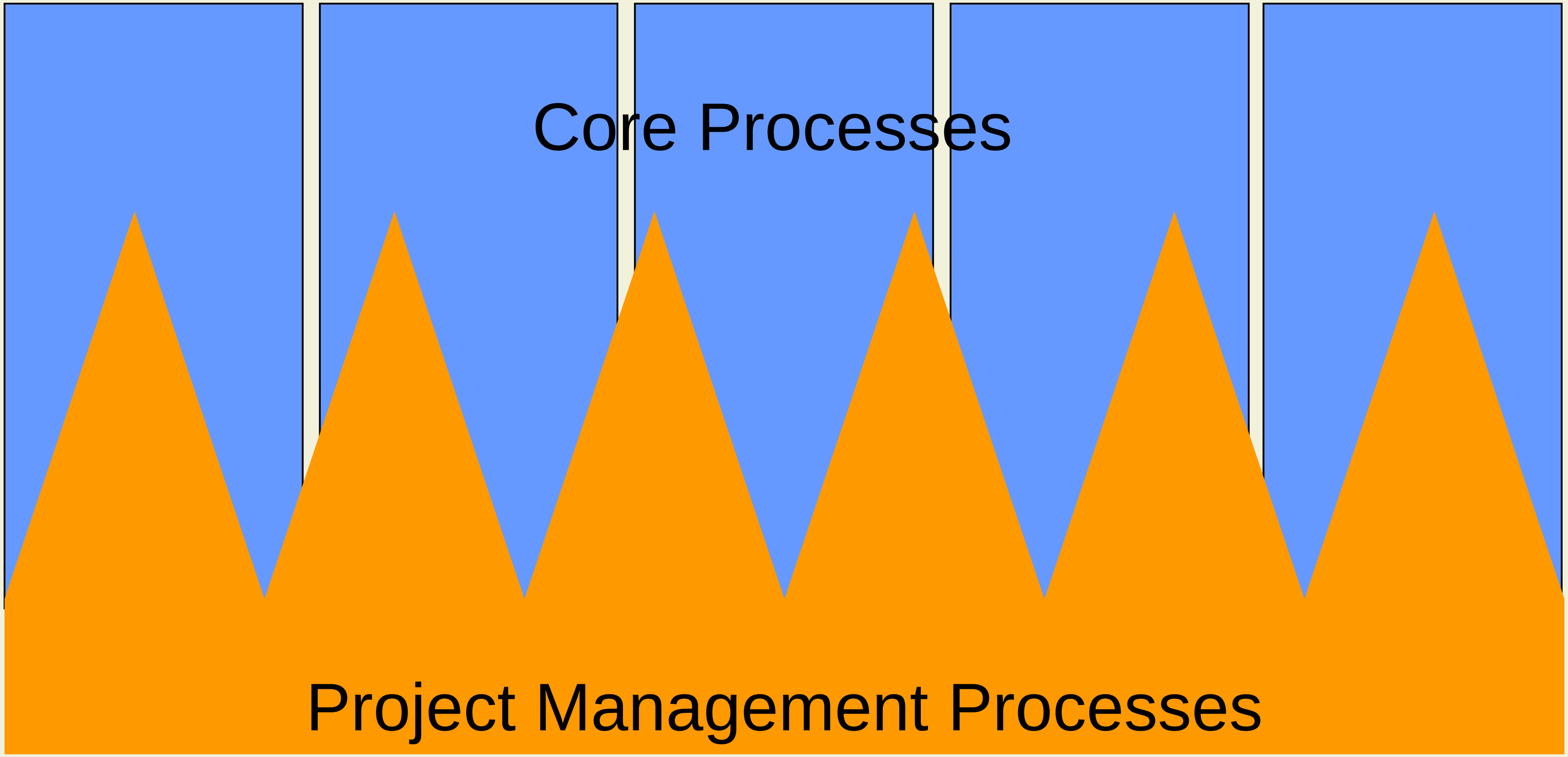


Interaction between Phases



- Input and output of the processes depend on the phase in which they are carried out
- <u>But</u> processes are not limited to one phase (overlaps)

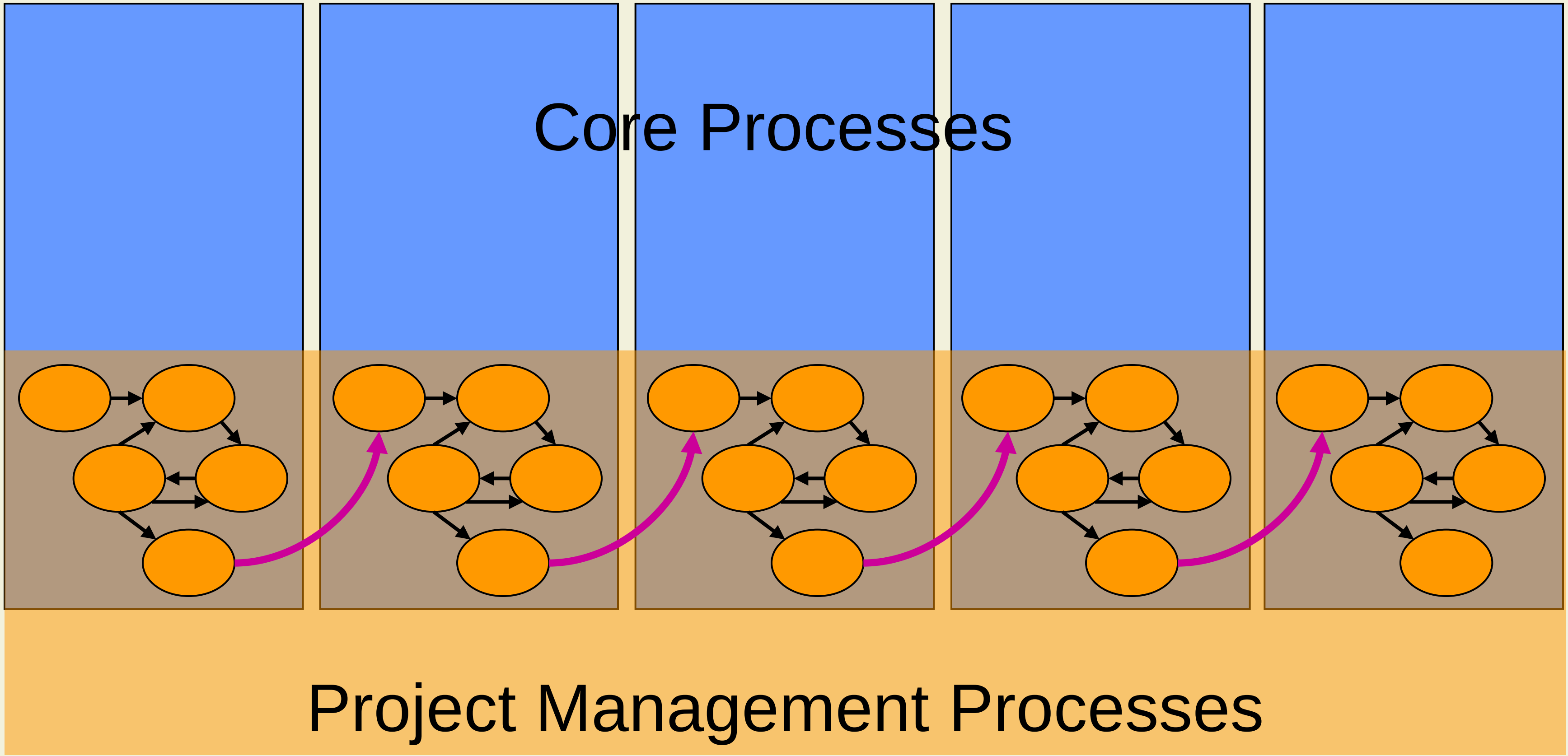
Core and Project Management Processes



Core Processes

Project Management Processes

Core and Project Management Processes



Core Processes

Project Management Processes

Systematics of Processes

	Initiating	Planning	Executing	Controlling	Closing
Integration		Project Plan Development	Project Plan Execution	Integrated Change Control	
Scope					
Time					
Cost					
Quality					
HR					
Comm.					
Risk					
Procurement					

2. Project Life Cycle and Project Management Life Cycle

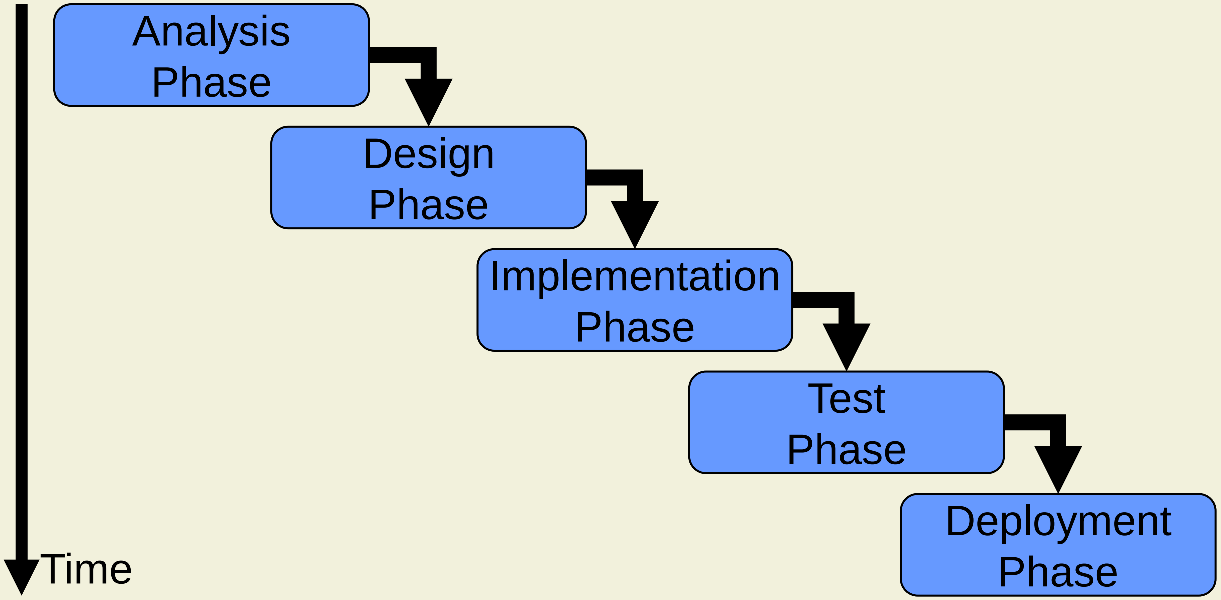
2.1 Project Life Cycle

2.2 Project Management Life Cycle

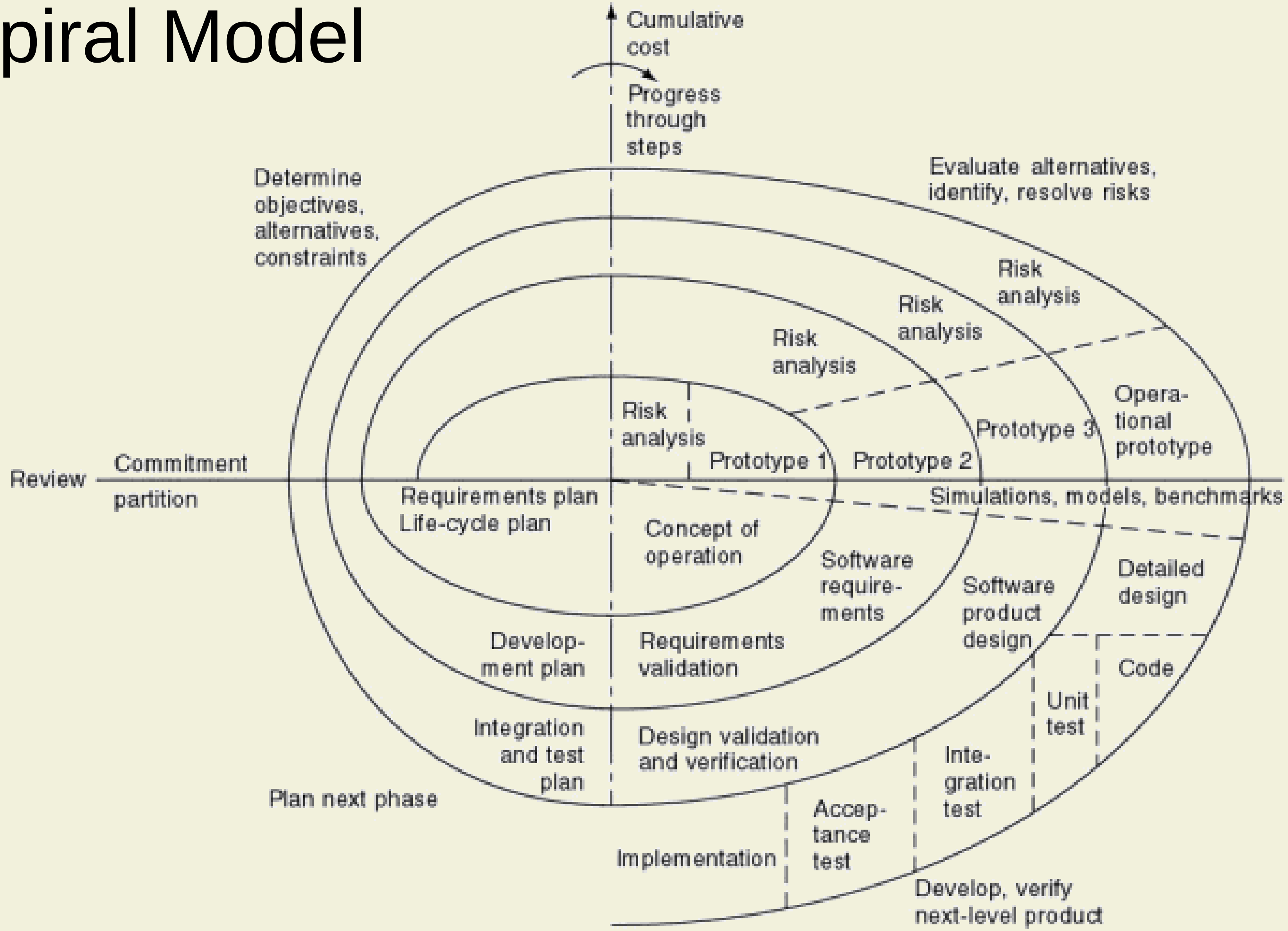
2.3 Development Models

Shortcomings of the Waterfall Model

- Division of labor hampers total quality management
- Lack of support for requirement changes
- Late appearance of actual code
- Lack of support for the maintenance activity



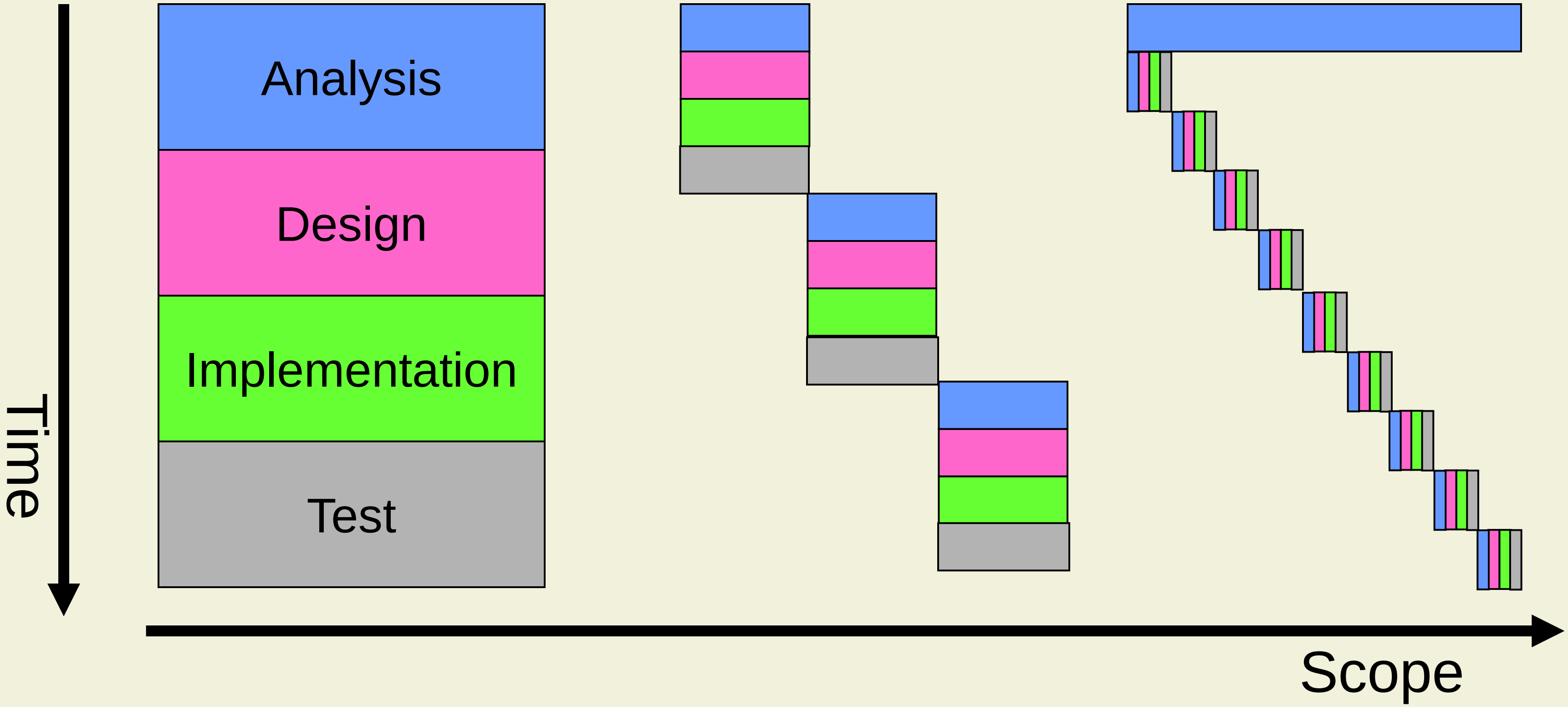
Spiral Model



Spiral Model

- Combines elements of both design and prototyping-in-stages
- Each phase starts with a design goal and ends with the client reviewing the progress thus far
- Advantages
    - Estimates get more realistic as work progresses
    - Supports changes
    - Good support for risk management
- Disadvantages
    - Estimates are harder at the outset

Extreme Programming



- Suggested reading: Kent Beck: Embracing Change with Extreme Programming, 1999

XP Practices

1. Planning game
    - Customers decide the scope and timing of releases based on estimates made by programmers
    - Programmers implement only functionality demanded by stories in this iteration
2. Small releases
    - Working system early
    - Releases anywhere from daily to monthly
3. Metaphor
    - System shape defined by a metaphor shared by the customer and programmers

XP Practices (cont'd)

4. Simple design
   - Design defines all the tests
   - Communicates everything the programmers want to communicate
   - Contains no duplicate code
   - Has the fewest possible classes and methods
   - Say everything once and only once
5. Tests
   - Programmers write unit tests
   - Customers write functional tests

XP Practices (cont’d)

6. Refactoring
   - System evolves through transformations of existing designs
   - Keep all tests running
7. Pair programming
   - All code written by two people at one screen, keyboard, mouse
8. Continuous integration
   - No more than one day between code integration
9. On-site customer
   - A customer sits with the team full-time

XP Practices (cont’d)

10. Collective ownership
    - Every programmer improves any code anywhere in the system at any time if he sees the opportunity
11. 40-hour weeks
    - No one can work a consecutive week of overtime
    - Even isolated overtime is a sign of deeper problems
12. Fair rules
    - Sign up to follow team rules
    - Team can change rules at any time as long as team agrees to the change

BACKUP