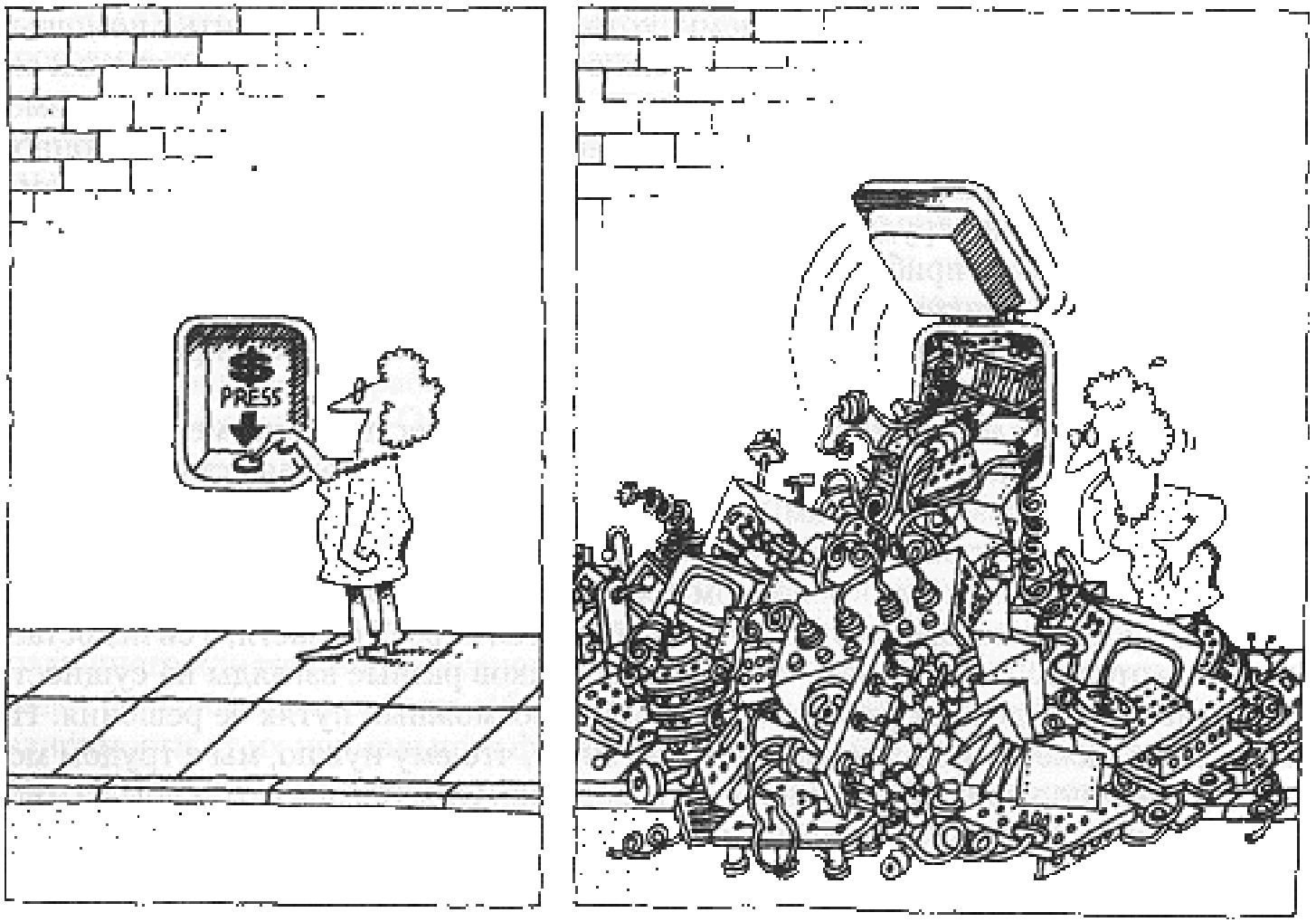
Ingegneria del Software

Corso di Laurea in Informatica per il Management

Software process model

Davide Rossi
Dipartimento di Informatica
Università di Bologna



“The task of the software development team is to engineer the illusion of simplicity”
[G. Booch]

Software process model

- Process: set of coordinated activities leading to a goal
- Software process: the goal is software production / deployment / evolution / maintenance

The software process's objectives

Planning / organizing / running a software project within given constraints, such as

- quality
- time
- cost (mostly resources, e.g. people)

optimizing (and assessing) progress, risks (profit, customer satisfaction, ...)

Typical software process activities (a.k.a. The software lifecycle)

- Specification
- Design
- Implementation
- Validation
- Evolution

Deliverables

Software is intangible, we may have lack of visibility, so we produce additional artifacts:

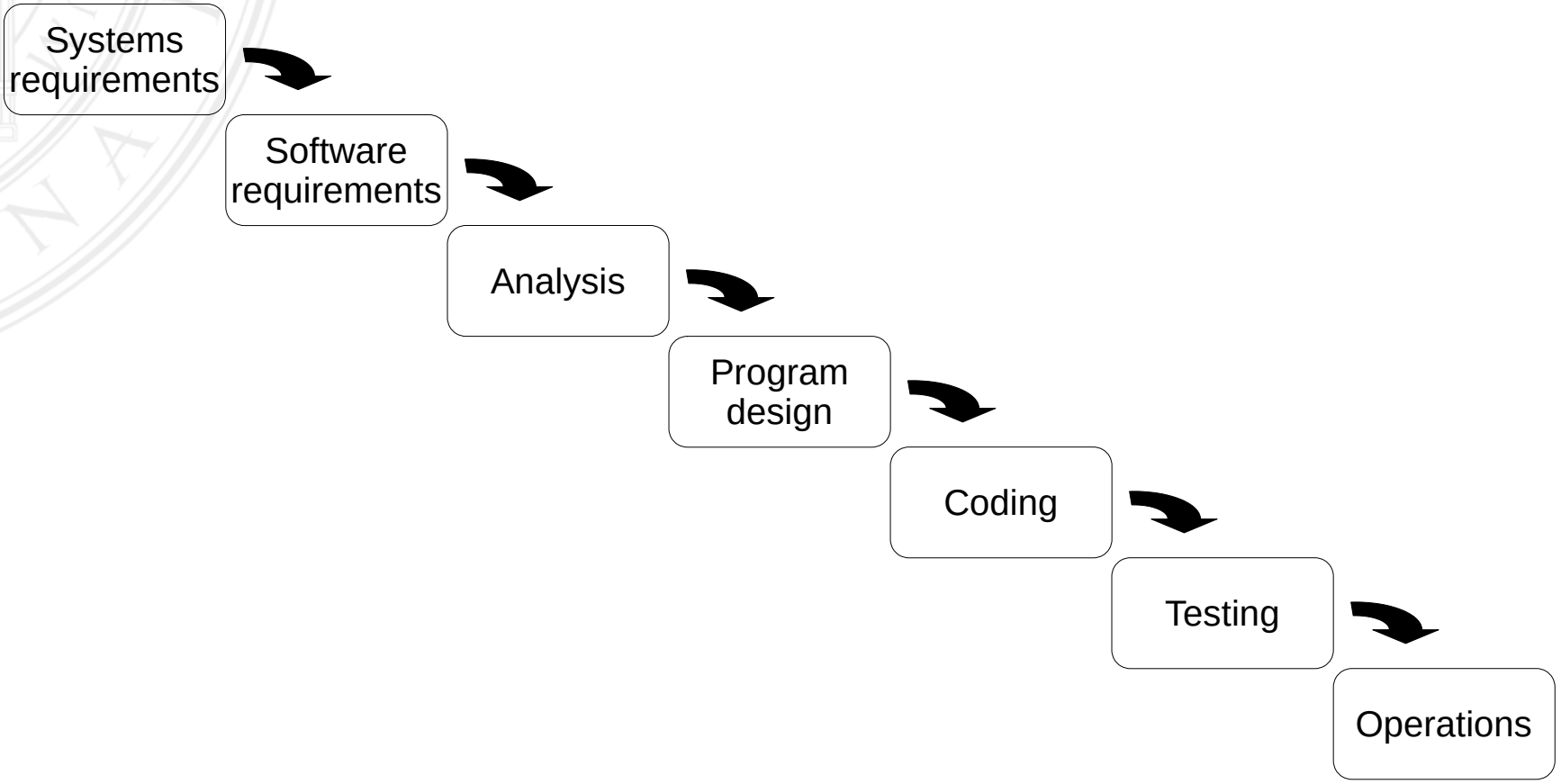
- Design documents/prototypes
- Reports
- Project/status meetings
- Client surveys (e.g. satisfaction level)

The waterfall model

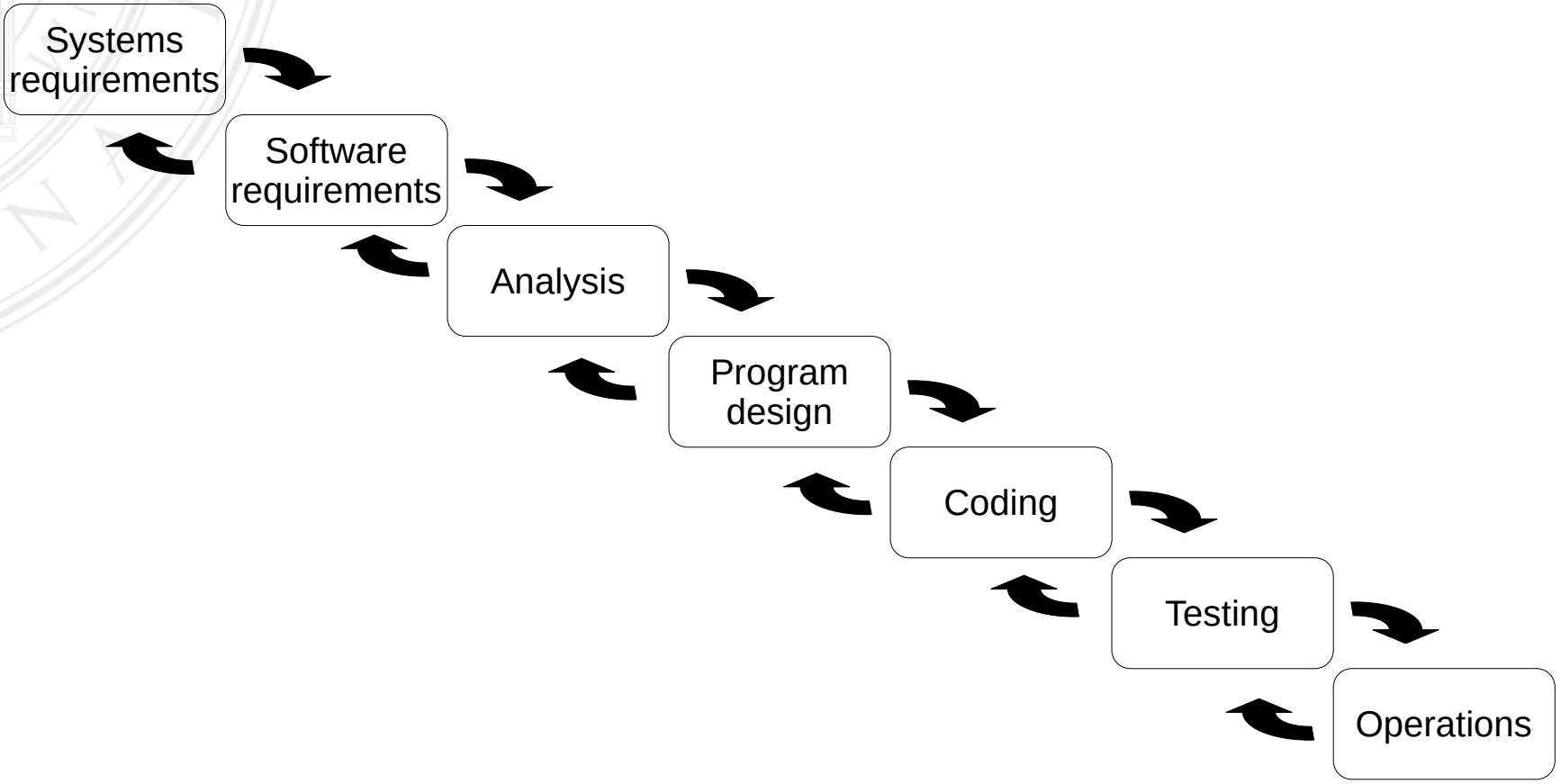
The waterfall model applies concepts from other engineering domains (manufacturing, construction, ...) to software production.

It is essentially composed by rigidly sequential activities.

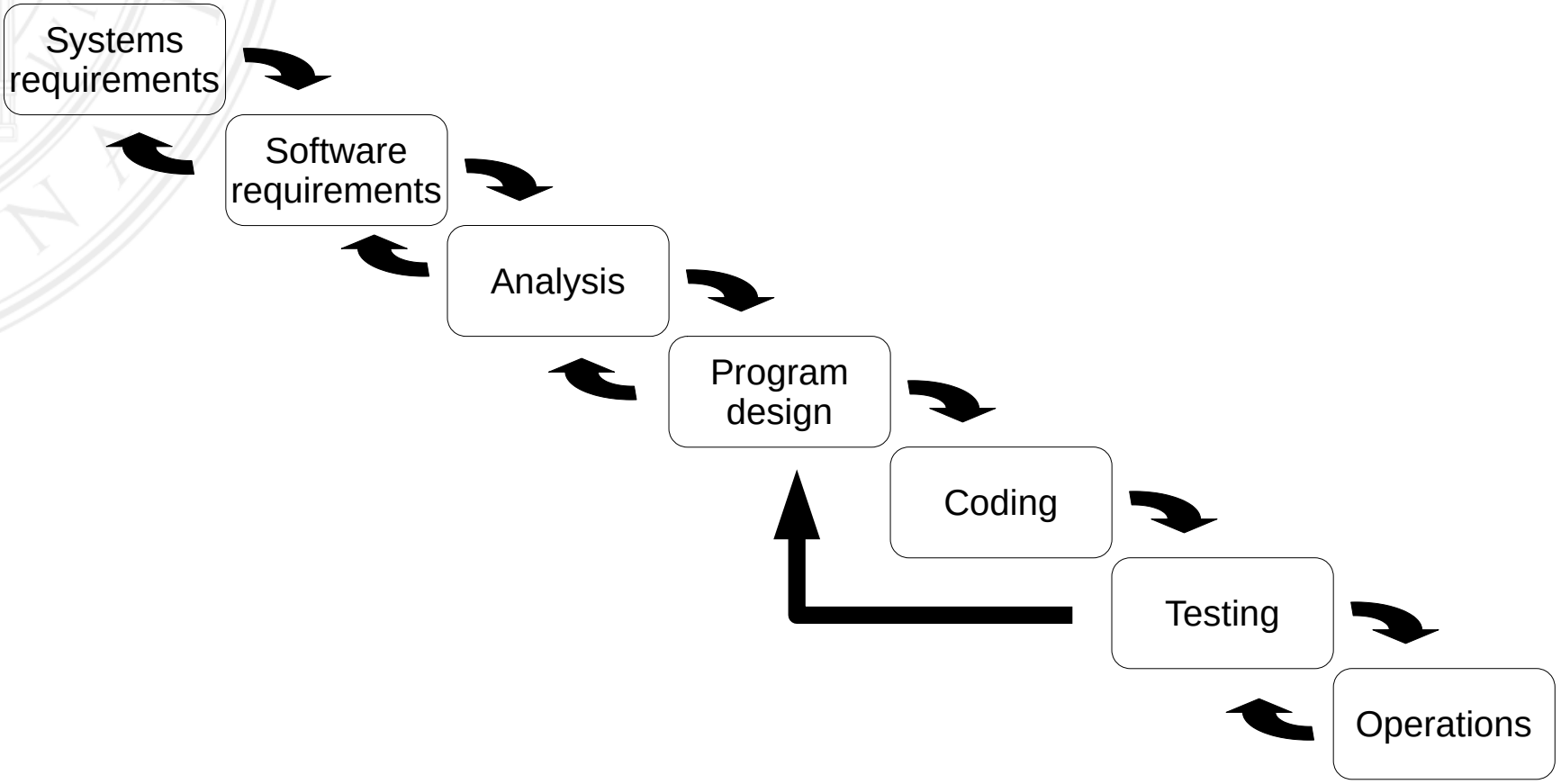
The waterfall model



The waterfall model



The waterfall model



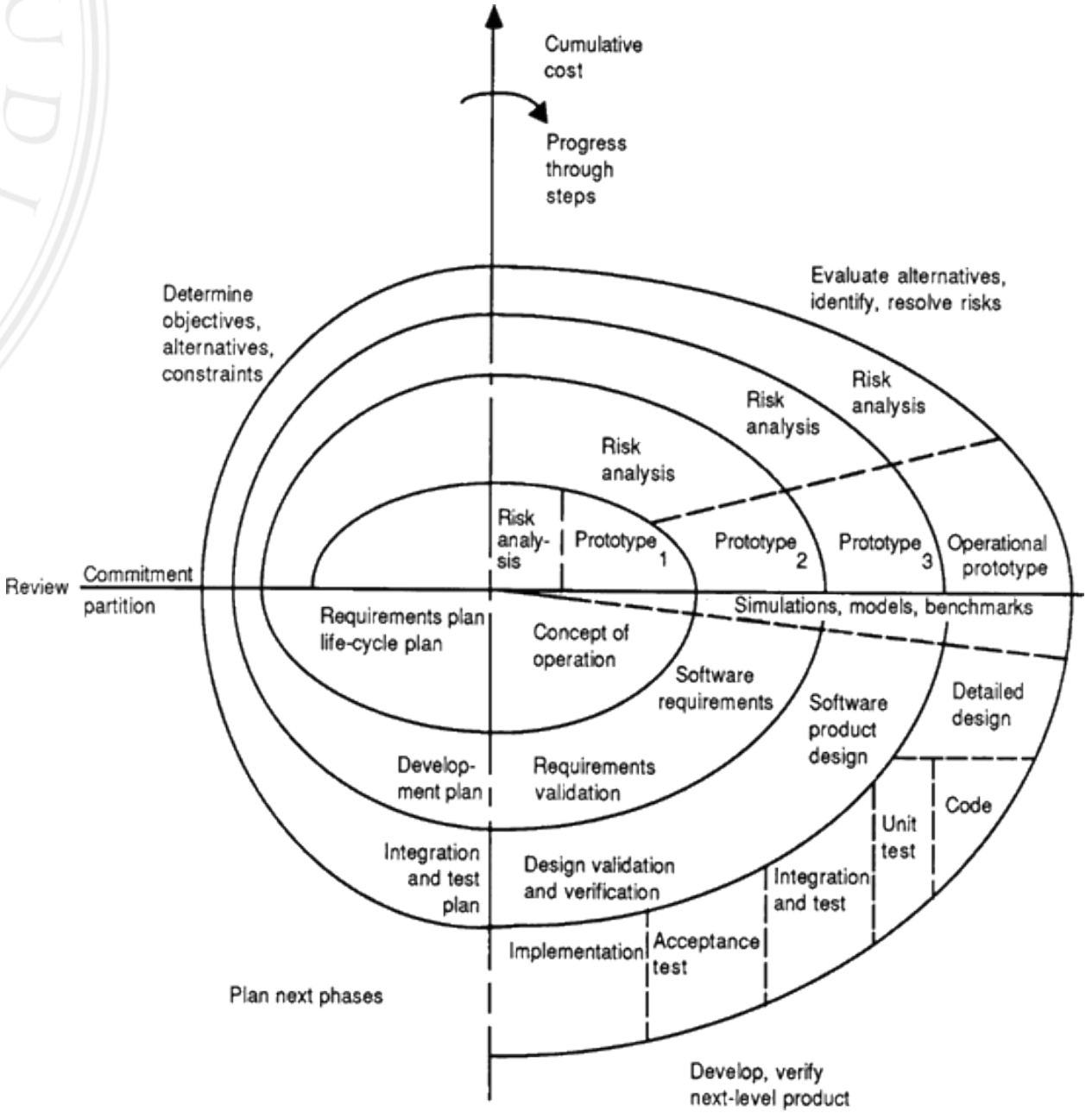
Waterfall: pros and cons

- Pros
  - Easy to understand
  - Enforces good practices
  - Identifiable deliverables and milestones
  - Comprehensive documentation
- Cons
  - Unrealistic
  - Late delivery
  - Ineffective risk management
  - Hard to cope with changes
  - High overhead

Spiral model

- *Family* of processes
- Risk-driven process model generator (phased reduction of risk)
  - *The major distinguishing feature of the Spiral Model is that it creates a risk-driven approach to the software process.* [Boehm]
- Iterative model (*cyclic approach*)
- Formalized in 1986 but used internally before that [Boehm 86]

Spiral model



Spiral model

- Each cycle starts with
  - Objectives
  - Alternatives
  - Constraints
- The next step is determined on the basis of the risks remaining
- Each cycle ends with a review from the stakeholders (mutual commitment)

Spiral: pros and cons

- Pros
  - Reflects the iterative nature of software development
  - Good visibility
  - Risk assessment
- Cons
  - Risk analysis is far from trivial
  - Complicated model, risks priorities could lead to late delivery
  - High overhead

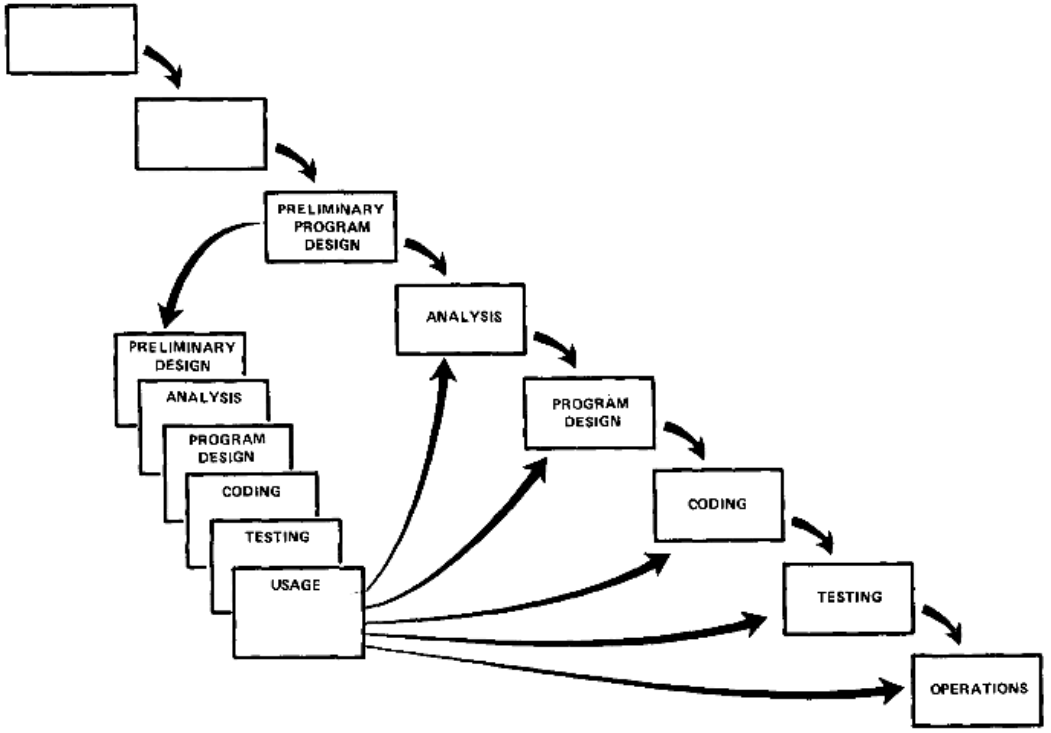
Iterative incremental development

Involves early programming and testing of a partial system, in repeating cycles. It also normally assumes development starts before all the requirements are defined in detail; feedback is used to clarify and improve the evolving specifications.

Empiric studies demonstrate that iterative methods are associated with higher success and productivity rates, and lower defect levels.

The waterfall model

Has it ever existed? The usual reference for the waterfall model [Royce 70] describes it as "fundamentally sound" but suggests to extend it to embrace iterations!



The waterfall model

Yet it became a military standard (DOD-STD-2167A)

Unified Process

The Unified Process (UP) is an iterative and incremental software development process framework.

The UP combines commonly accepted best practices, such as an iterative lifecycle and risk-driven development, into a cohesive and well-documented process description.

Unified Process

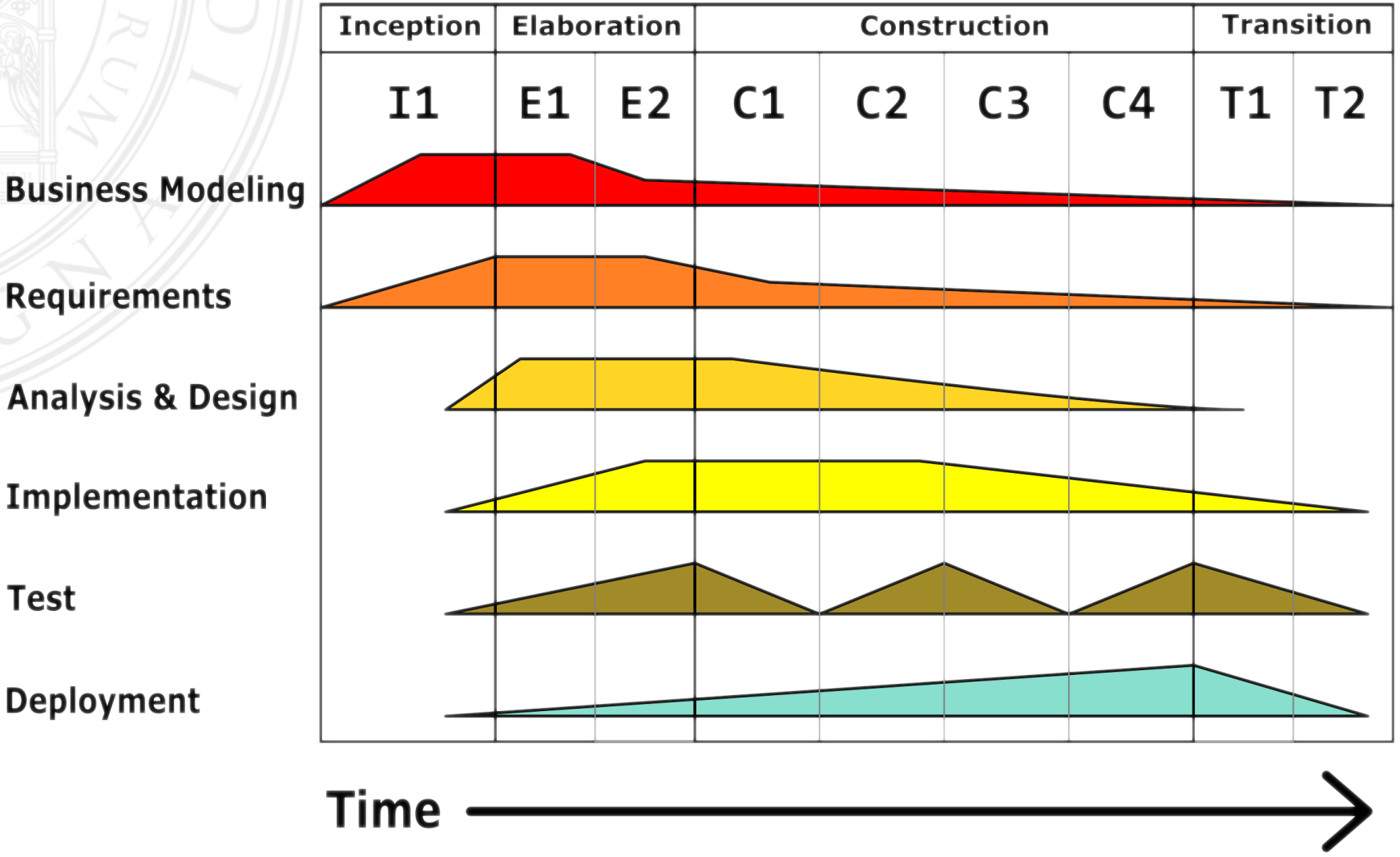
- Iterative and incremental
- Use case-driven
- Architecture-centric
- Risk focused

Unified Process

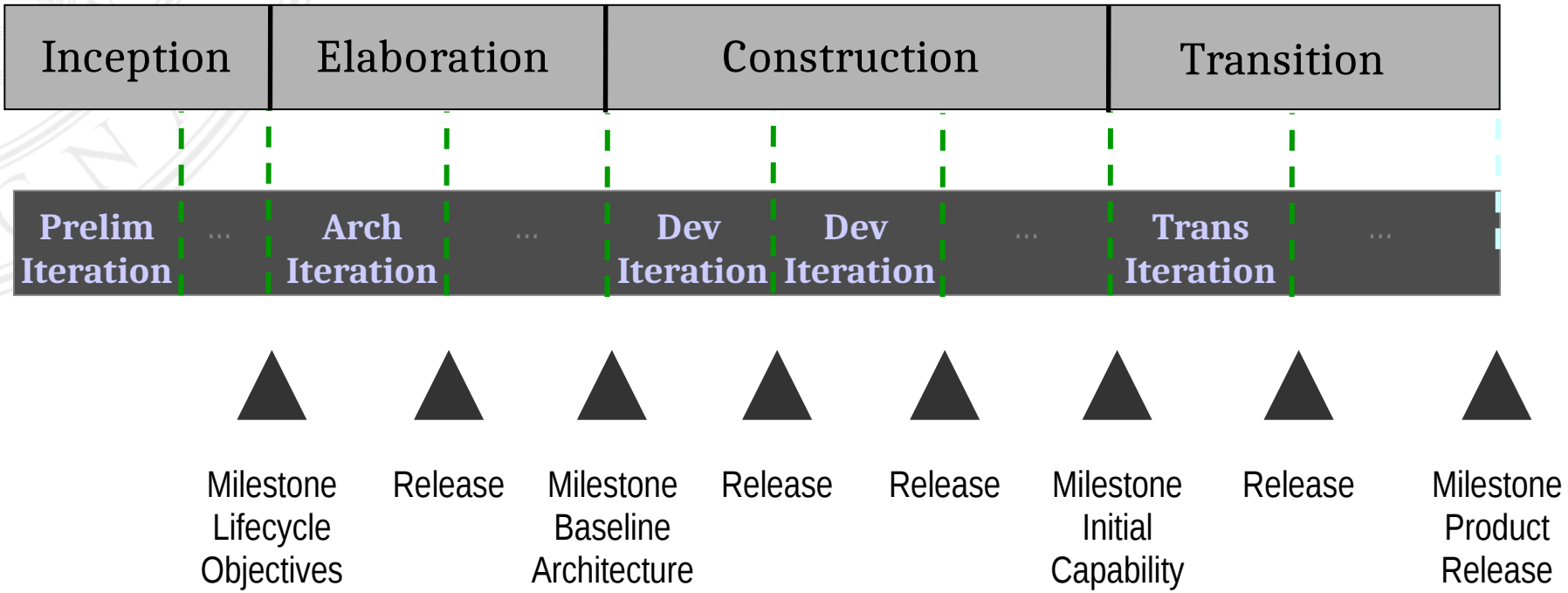
The Unified Process divides the project into four phases

- Inception
- Elaboration
- Construction
- Transition

Unified Process



Phases and iterations



UP: inception

- Goals
  - Business case/scope
  - Use cases
  - Candidate architectures
  - Risk identification
- Ends with *Lifecycle Objective Milestone*

UP: elaboration

- Goals
  - Address risks
  - Validate architecture
- En *executable architecture baseline* is implemented
- Ends with a plan for the construction phase (includes costs and times)

UP: construction

- Goal
  - Implement system features
- Uses timeboxed iterations producing a release
- Incremental refinement

UP: transition

- Goals
  - Deploy the system
  - User engagement
  - Collect feedback
- Also includes training

UP: most known implementations

- Rational Unified Process (RUP)
- Agile Unified Process
- Open Unified Process (OpenUP)
- Oracle Unified Method