Software Development Process and Activities

CS 490MT/5555, Spring 2016, Yongjie Zheng

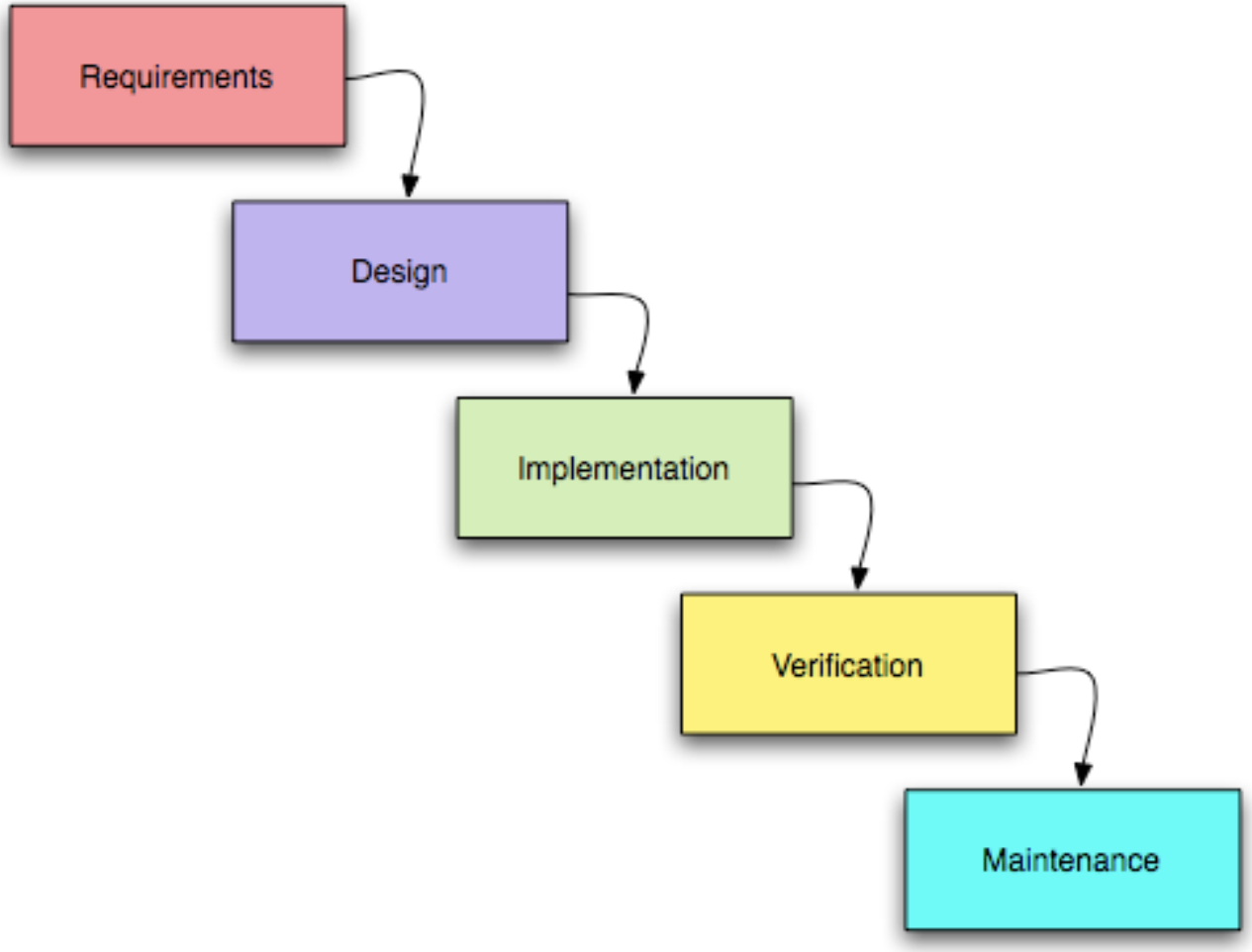
Software Process

- A set of activities that leads to the production of a software product
  - What product we should work on next
  - What criteria that work product must satisfy
- There is no standard or ideal process!
  - For some systems, such as critical systems, a very structured development process is required.
  - For business systems, with rapidly changing requirements, a flexible process is likely to be more effective.
  - For large systems, a mixed process is often preferred.

Software Process Models

- Waterfall Model
- V-Model
- Prototyping
- Iterative Process
  - Incremental Development
  - Spiral Model
  - Agile Methods
- Rational Unified Process (RUP)

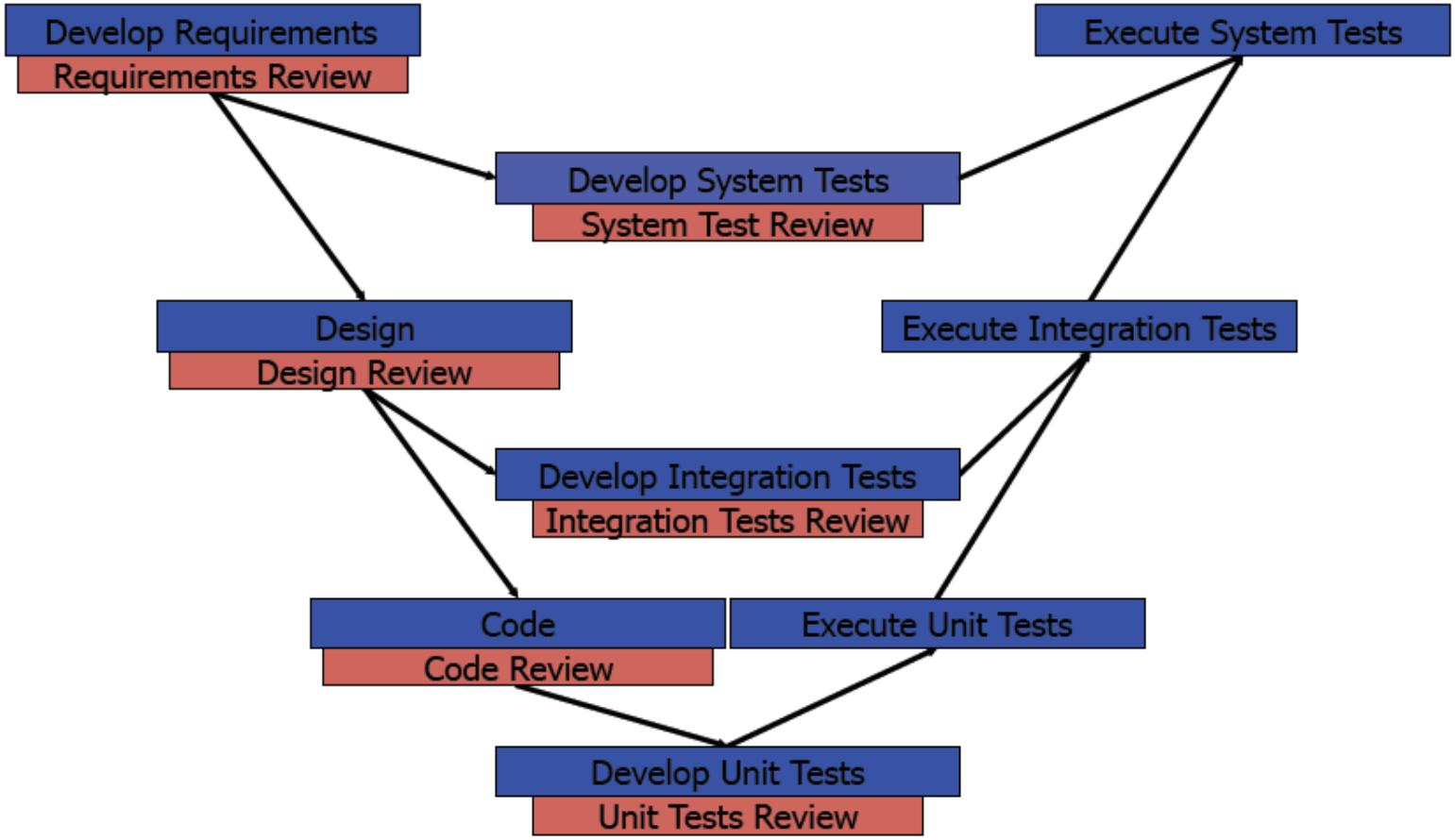
Waterfall Model



Waterfall Model

- Pros
  - A plan-based, document-driven, rigor process
  - Fits with other engineering process model
- Cons
  - Premature freezing of requirements
  - Inflexibility
- Usages
  - Critical systems
  - Large systems (hardware & software)
  - Different development teams in different places
  - Long lifetime

V-Model (the extension of Waterfall)



Annotated V-Model

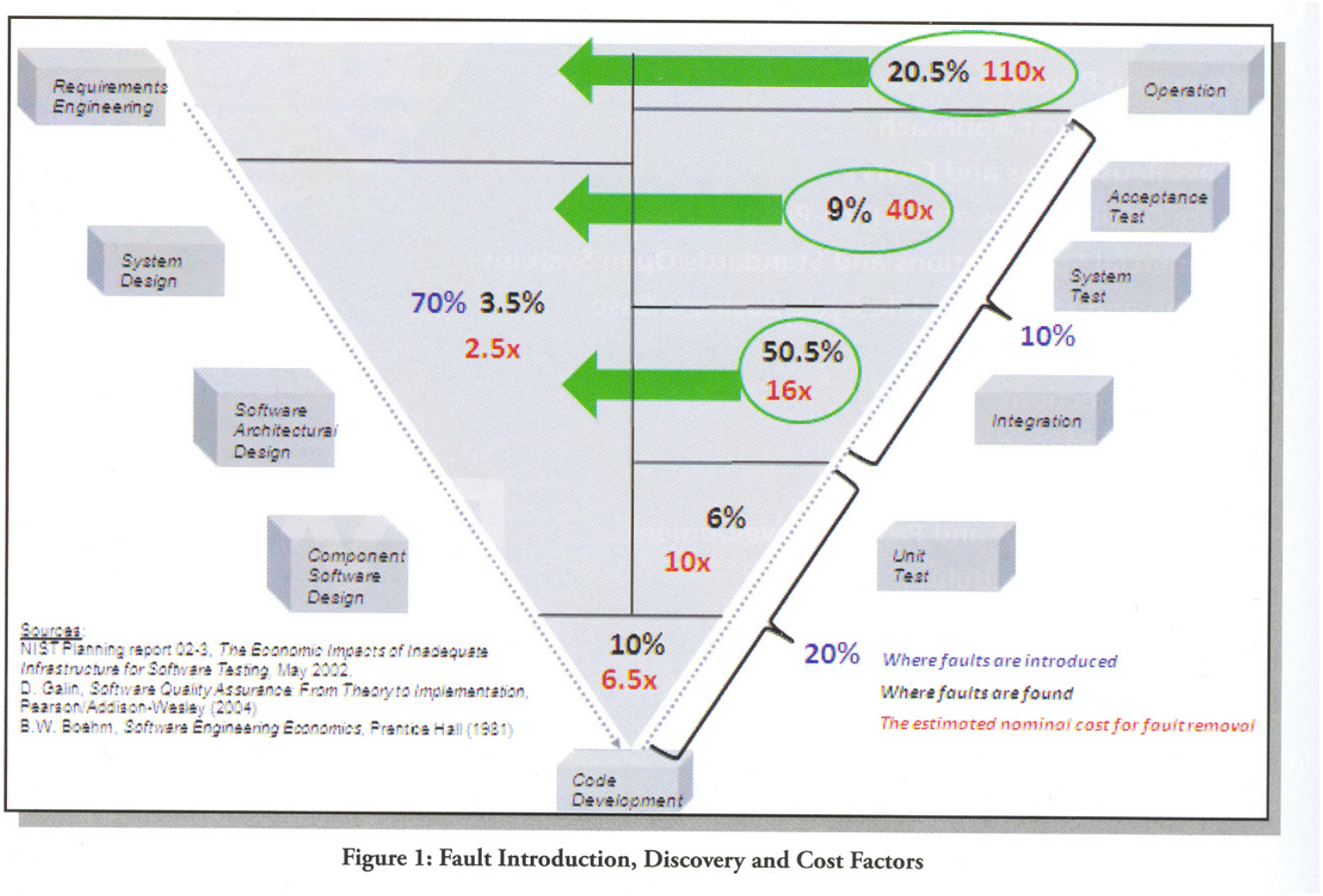
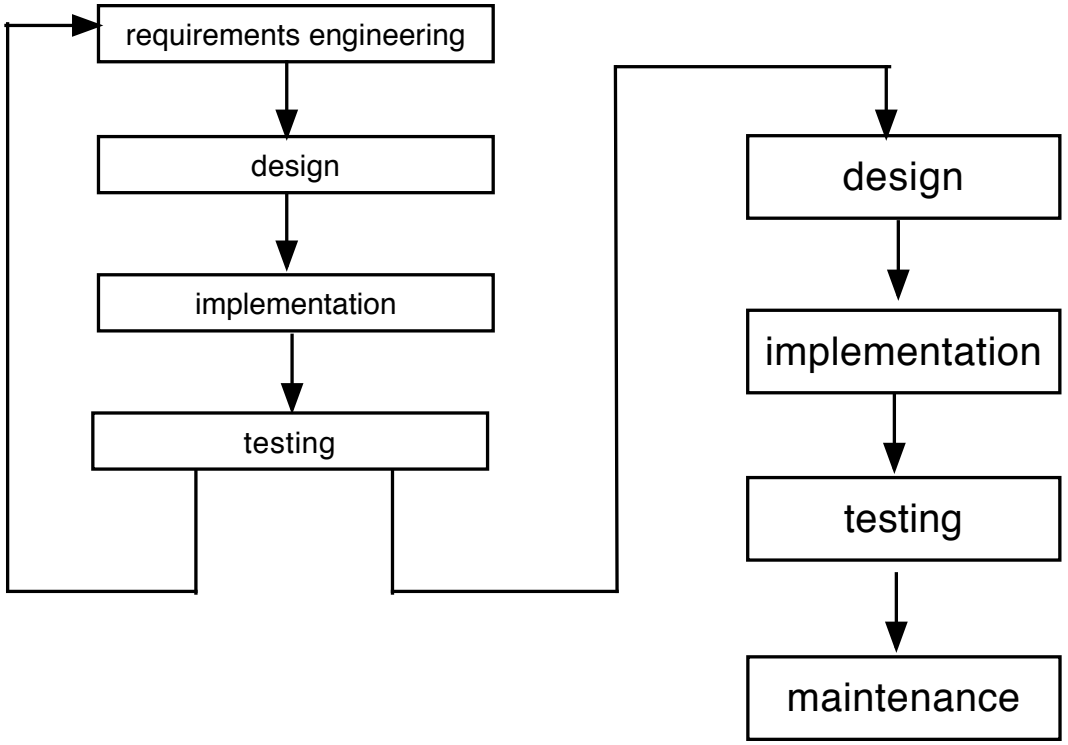


Figure 1: Fault Introduction, Discovery and Cost Factors

Prototyping



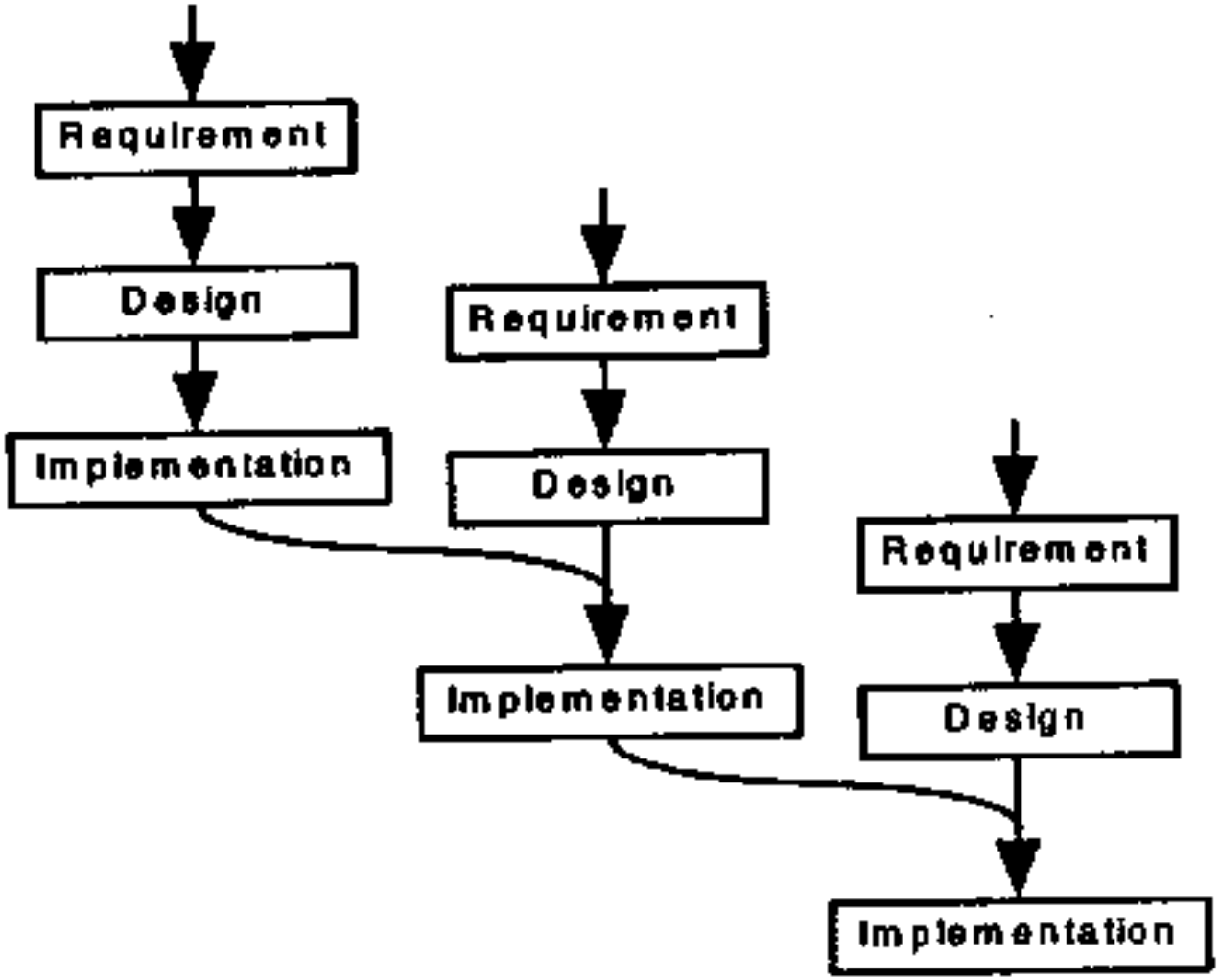
Prototyping

- Prototype
  - Prototype is a working model of a software system, which emphasizes certain aspects.
  - Usually starts with requirements that are not well understood
- Pros
  - A good way to clarify user requirements
  - Useful for user interface design
- Cons
  - The process is not visible
  - The resulting system can be poorly structured

Iterative Processes

- The essence of iterative processes is that the specification is developed in conjunction with the software.
  - Incremental Development
  - Spiral Model
  - Agile Methods

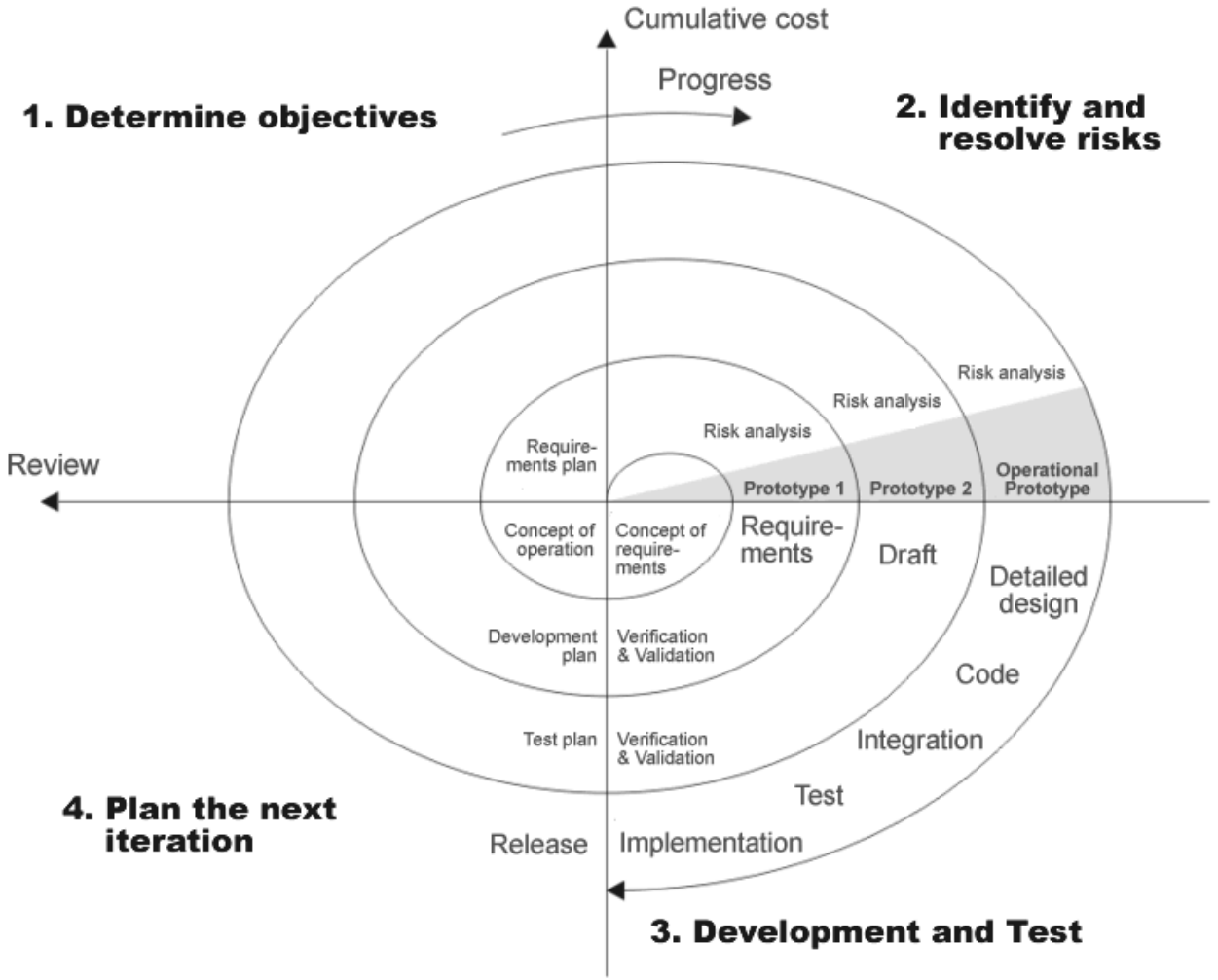
Incremental Development



Incremental Development

- The functionality of the system is produced and delivered to the customer in small increments.
- Starts with the user requirements that are best understood, or essential features.
- Each release is a mini-waterfall.
- Pros
  - Avoids the "Bing Bang" effect.
  - The most important system services receive the most testing.
- Cons
  - Contractual problems
  - Hard to manage

Spiral Model



Spiral Model

- A risk-driven process model
- Each loop in the spiral consists of:
  - Identify objectives and alternatives
  - Evaluate the alternatives, identify and resolve risks
  - Develop the identified portion of the product
  - Plan for succeeding phases
- Pros
  - Subsumes previous moels
- Cons
  - Not clear how to analyze risk

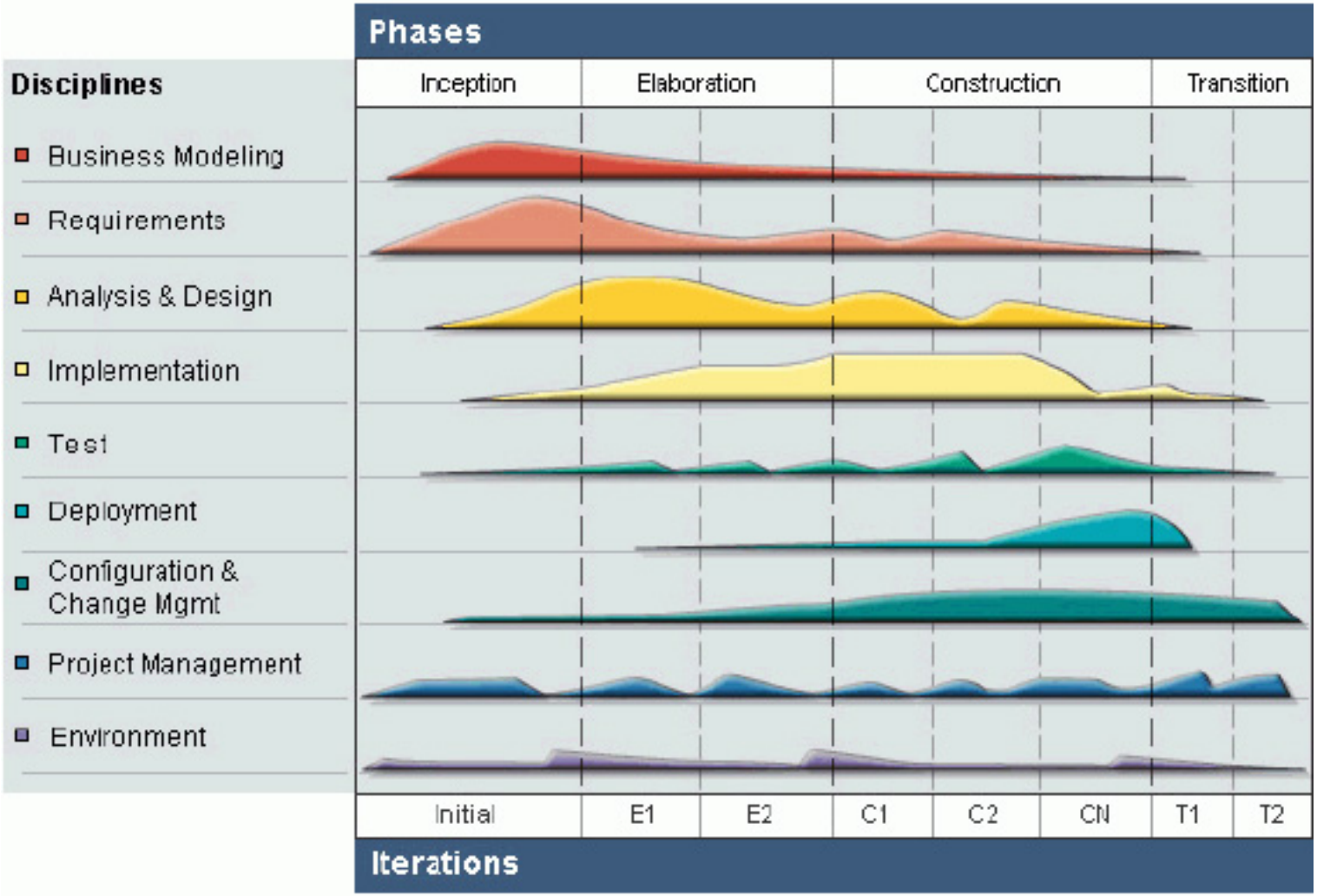
Agile Methods

- Traditional plan-based approaches (waterfall, spiral, incremental) often involve a significant overhead in planning, designing, and documenting the system.
- Agile methods allow the development team to focus on the software itself rather than on its design and documentation

Agile Methods

- Principles
  - Customer involvement
  - Incremental delivery
  - No prescriptive process
  - Embrace change
  - Maintain simplicity
- Examples
  - Extreme Programming (XP)
    - Pair programming
    - Test-driven development

Rational Unified Process (RUP)



Rational Unified Process (RUP)

- Separation of phases and development activities (workflows)
  - Four phases: Inception, Elaboration, Construction, Transition
  - Nine workflows: Business modeling, Requirements, Analysis and design, Implementation, Testing, Deployment, Configuration and change management, Project management, Environment
- In principle, all of RUP workflows may be active at all phases of the process.
- A hybrid of traditional process models
  - A phased model where phases are more related to business
  - Each phase is enacted in an iterative way
  - The whole set of phases may also be enacted incrementally

Software Engineering Activities (covered in this course)

- Requirements Analysis and Specification
  - Requirements Specification (UML Use Case Diagrams)
- Software Architecture and Design
  - Architectural Design (ArchStudio)
  - Detailed Design (UML Class Diagrams, etc.)
- Software Implementation
  - Development Environment (Eclipse)
- Software Testing
  - Unit Testing (JUnit Testing Framework)
- Software Maintenance
  - Version Control (Subversion, GIT)

Other Software Engineering Activities and Topics (not covered)

- Software Deployment
- Software Refactoring
- Software Metrics
- Software Mining
- Software Usability
- Software Product Line
- …

Reminder

- Lab #1 is on next Tuesday.
- Assignment #1 will be assigned after Lab #1.