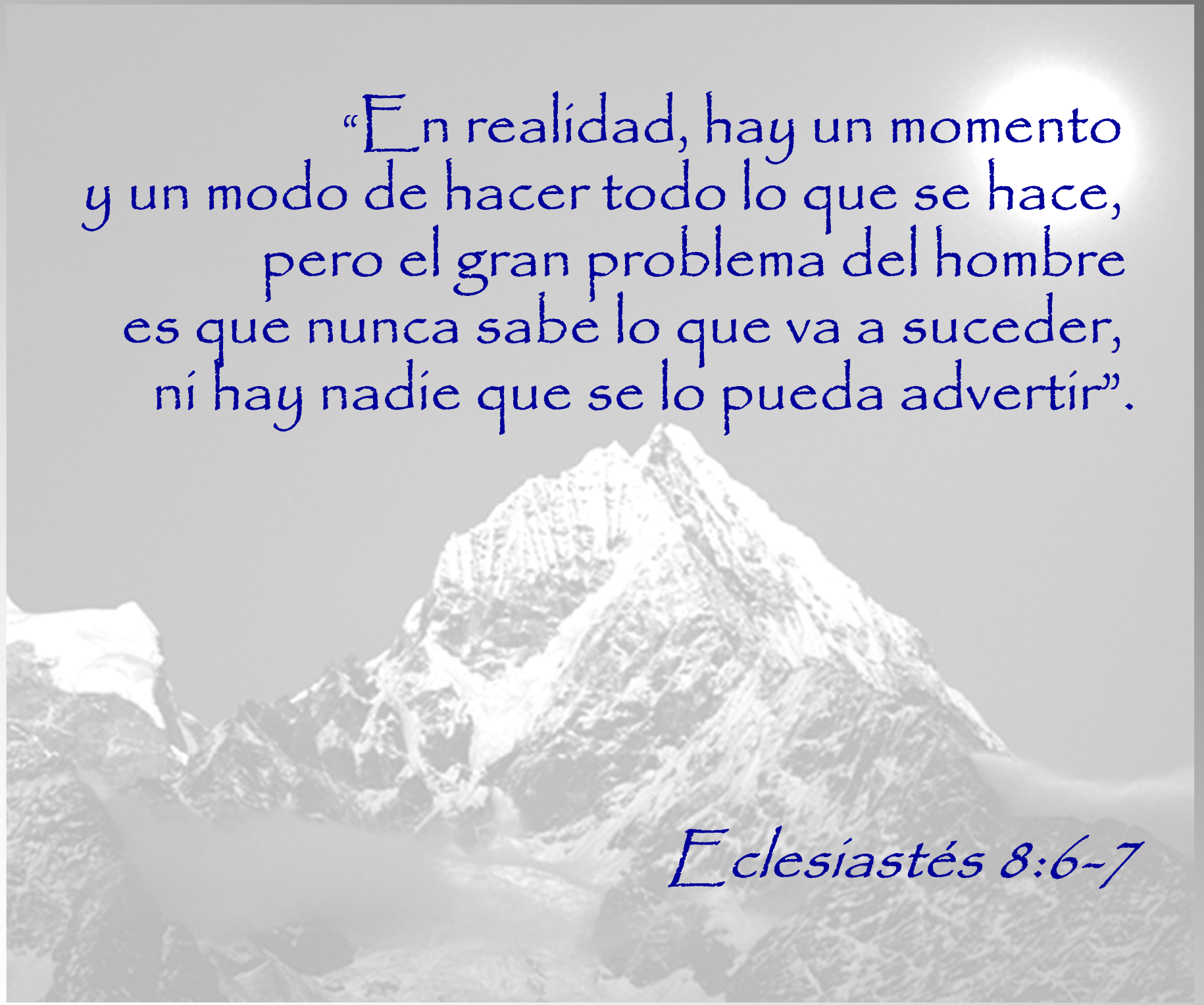
Ciclos de Vida del Desarrollo de Software

Mcc. Fernando Rojas Morales

Ingeniería de Software 1

Mcc. Fernando A. Rojas Morales – UIS - EISI



Ciclos de Vida del Desarrollo de Software

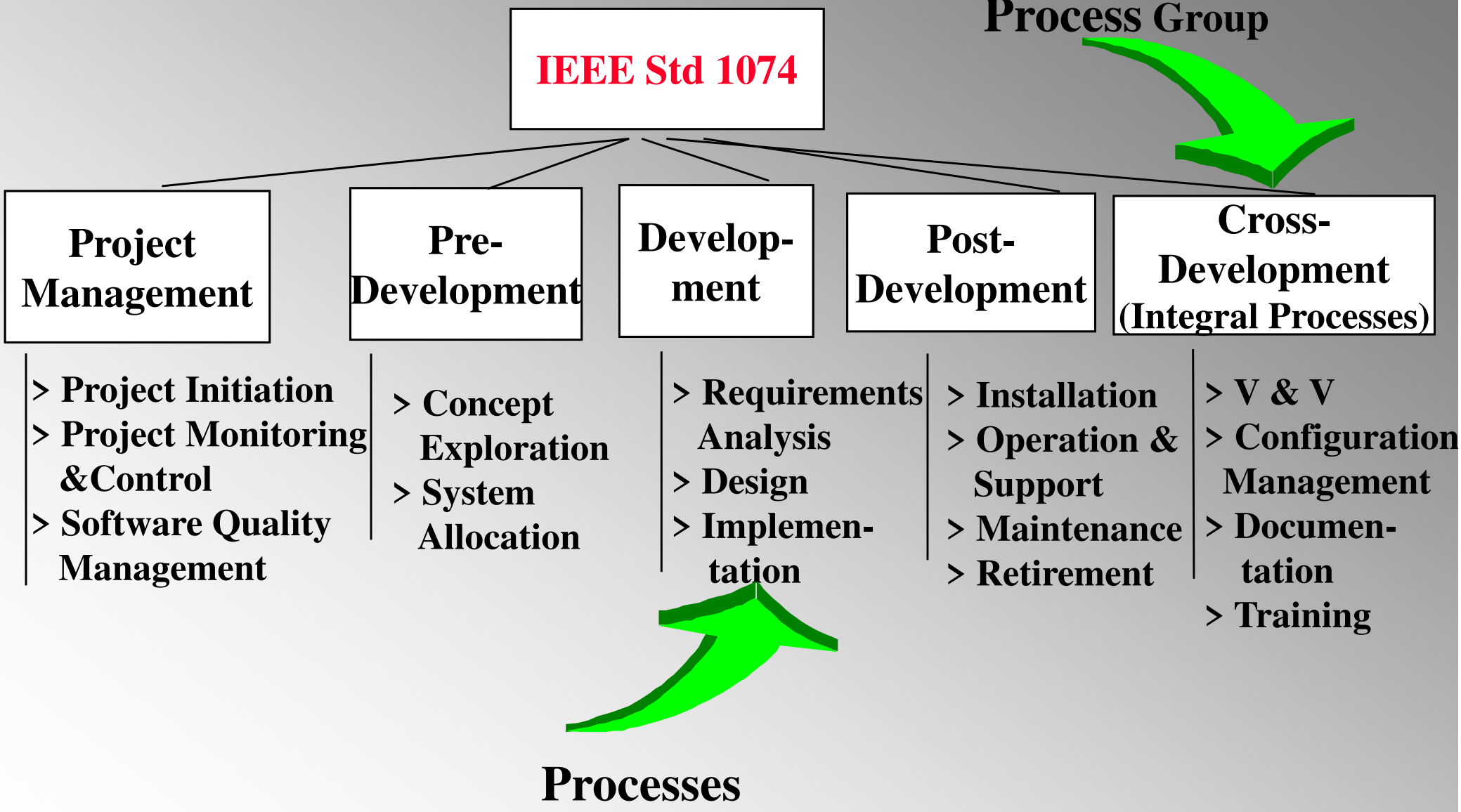
Definiciones

- El modelado del ciclo de vida: Intenta tratar con la complejidad y el cambio

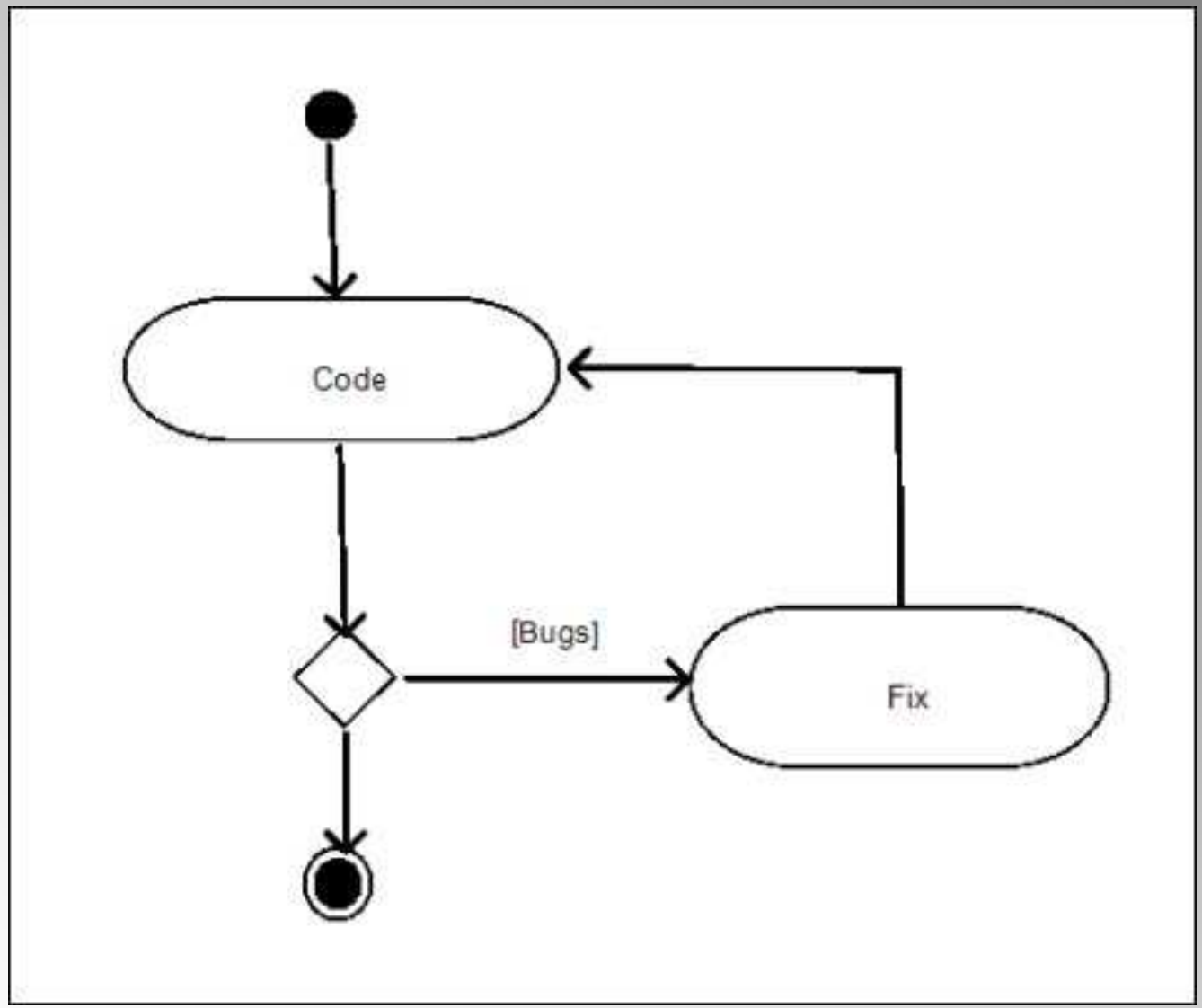
- El ciclo de vida del software:
  - **Conjunto de actividades y sus relaciones que dan soporte al desarrollo de un sistema de software**

- Metodología de desarrollo de software:
  - **Colección de técnicas para la construcción de modelos - aplicadas a través del ciclo de desarrollo de software**
  - **Estructuradas y Orientadas a Objetos**

IEEE Std 1074 - 1995: Standard for Software Lifecycle

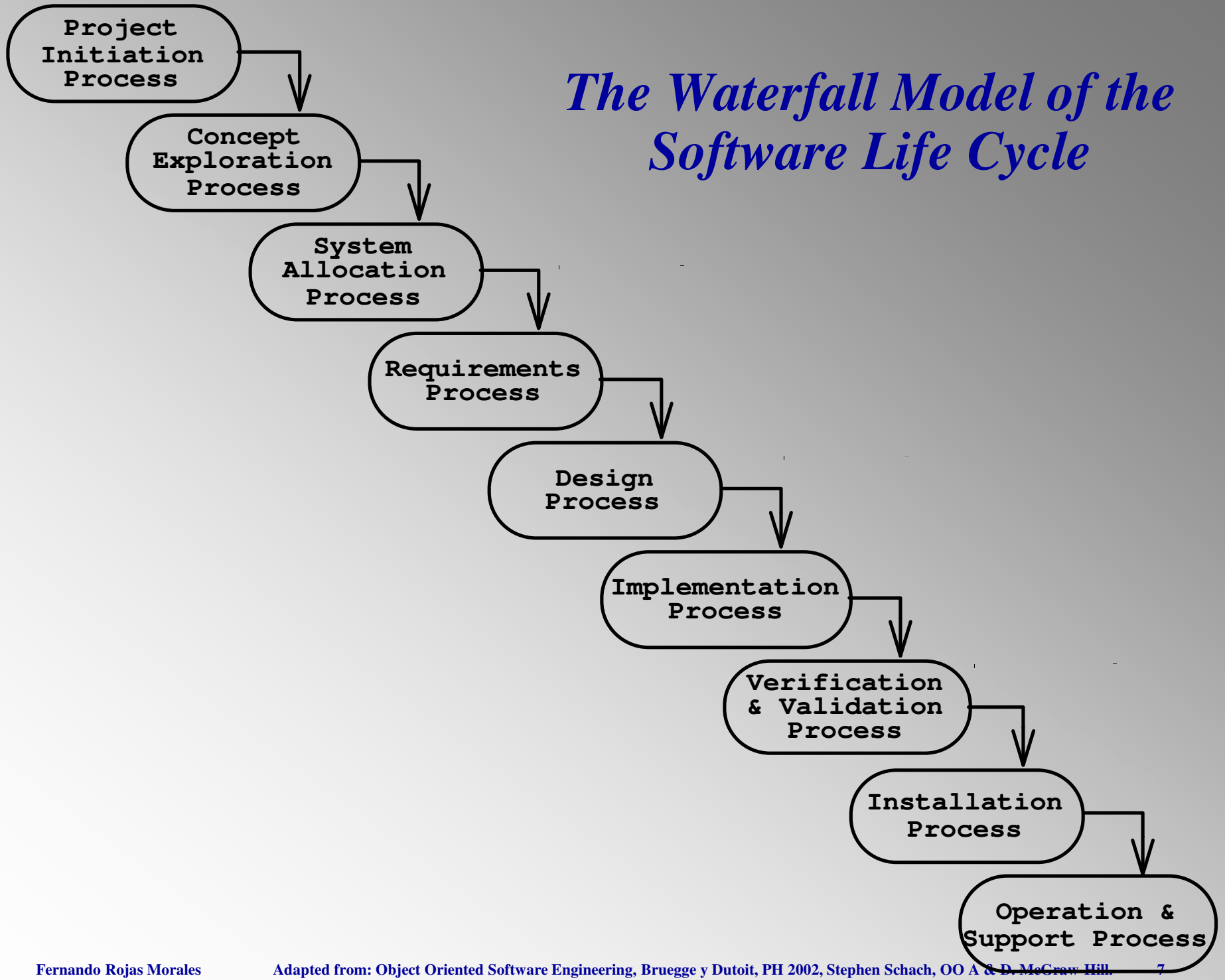


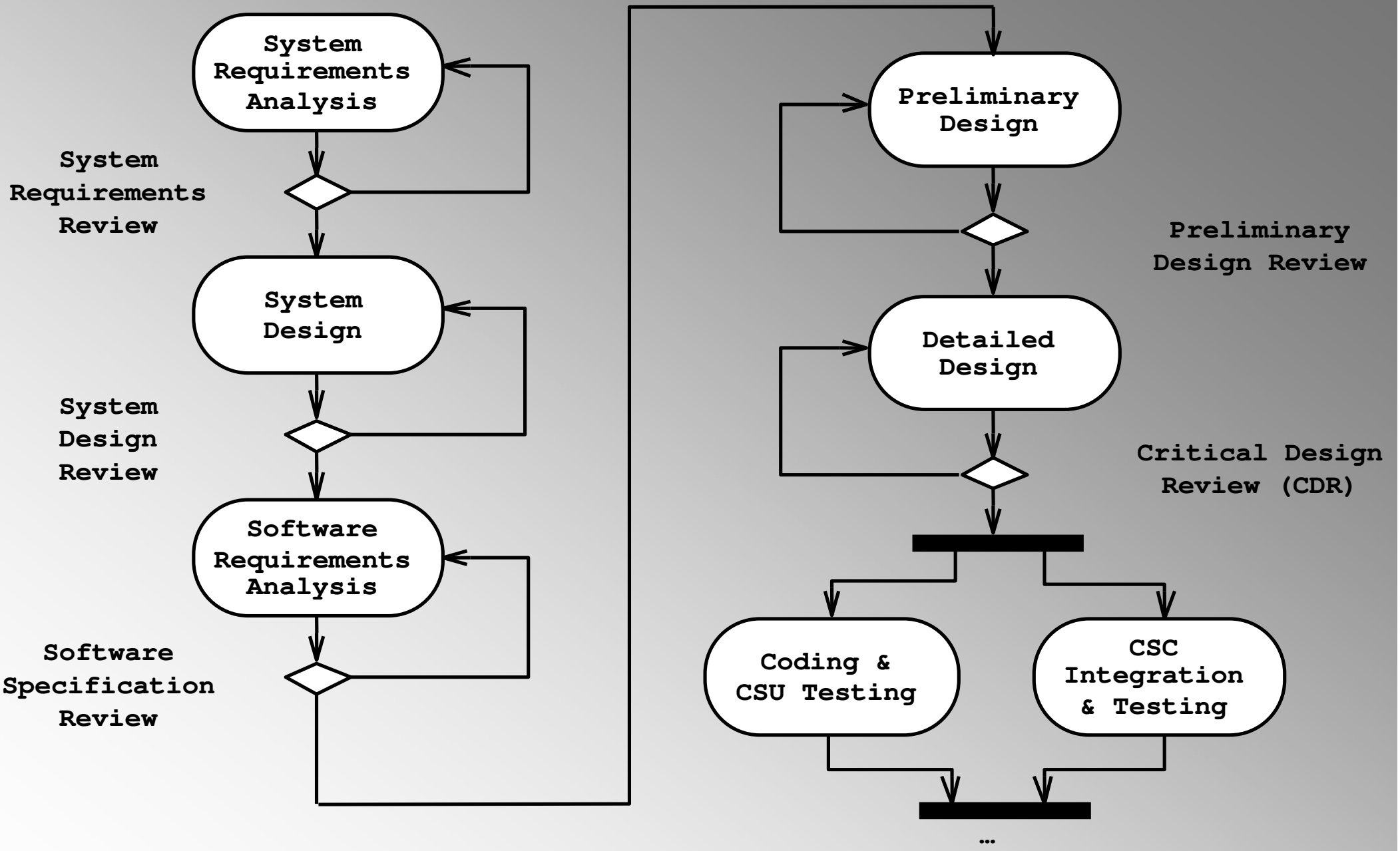
El primer Ciclo de Vida del Desarrollo de Software



- El Intuitivo “Code and Fix”

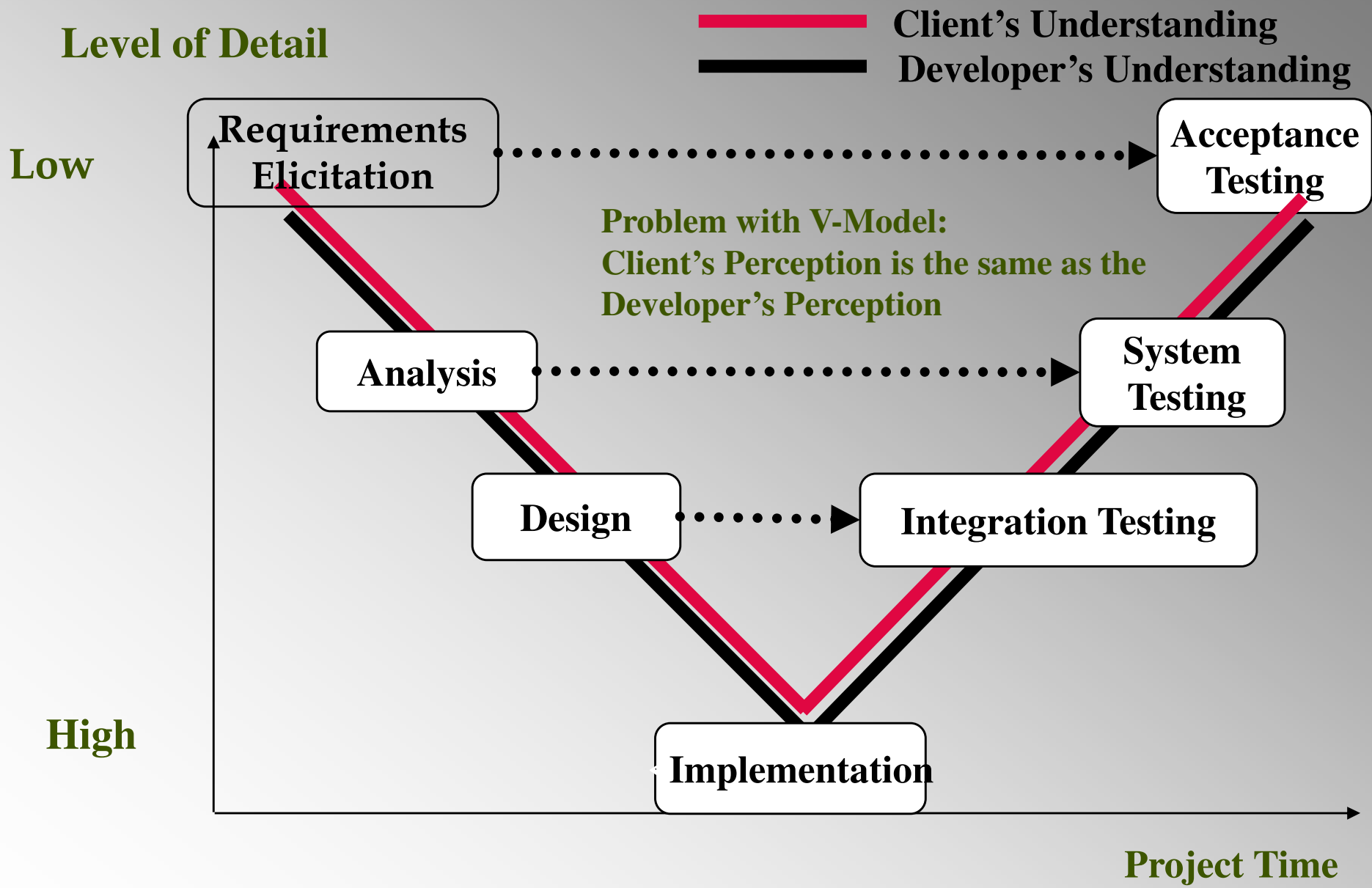
The Waterfall Model of the Software Life Cycle



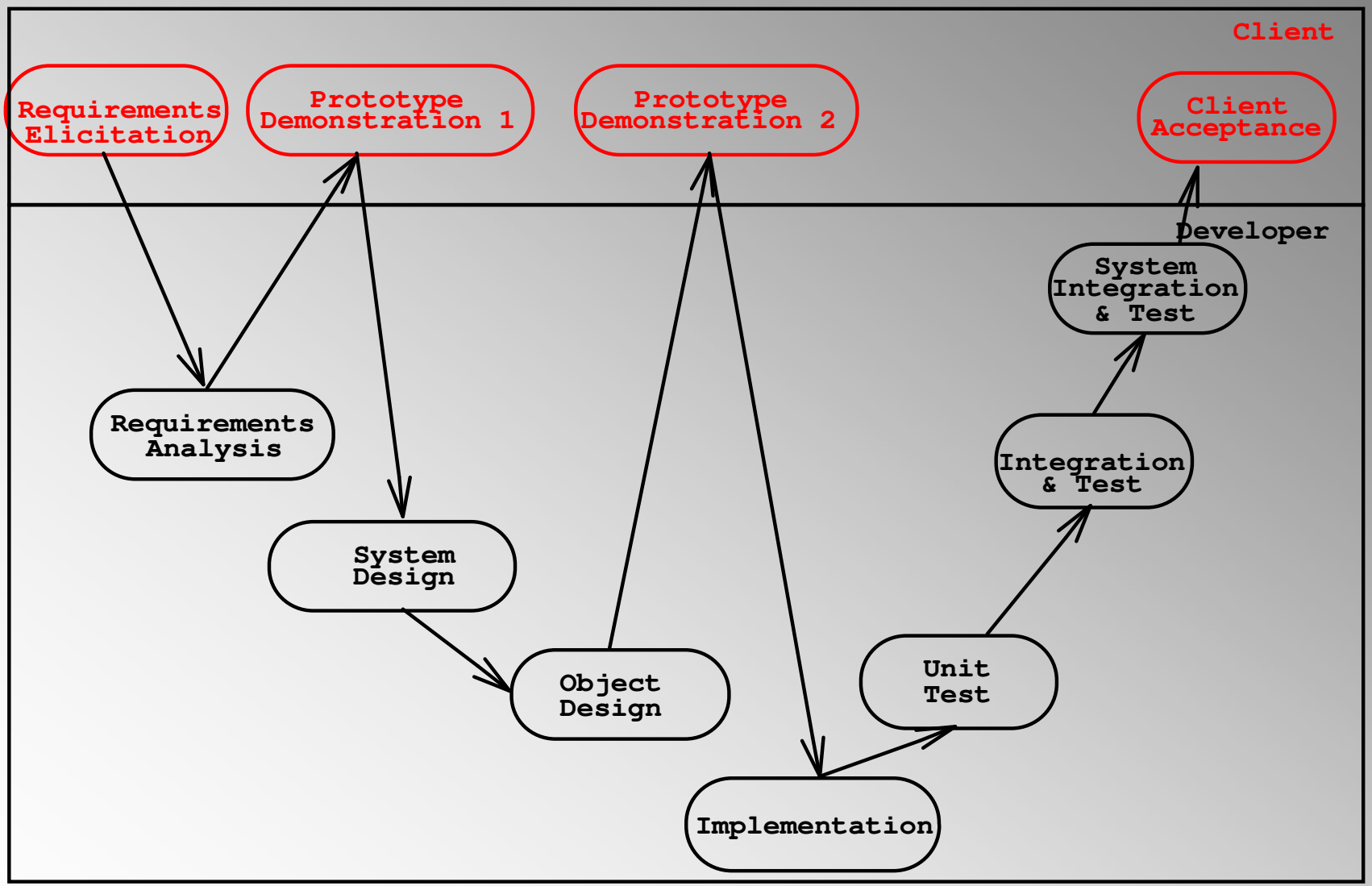


Waterfall model for the DOD Standard 2167A (UML activity diagram). Note activities specific to the DOD are used instead of IEEE 1074 activities. Decision points denote reviews: The subsequent activity is initiated only if the review is successful.

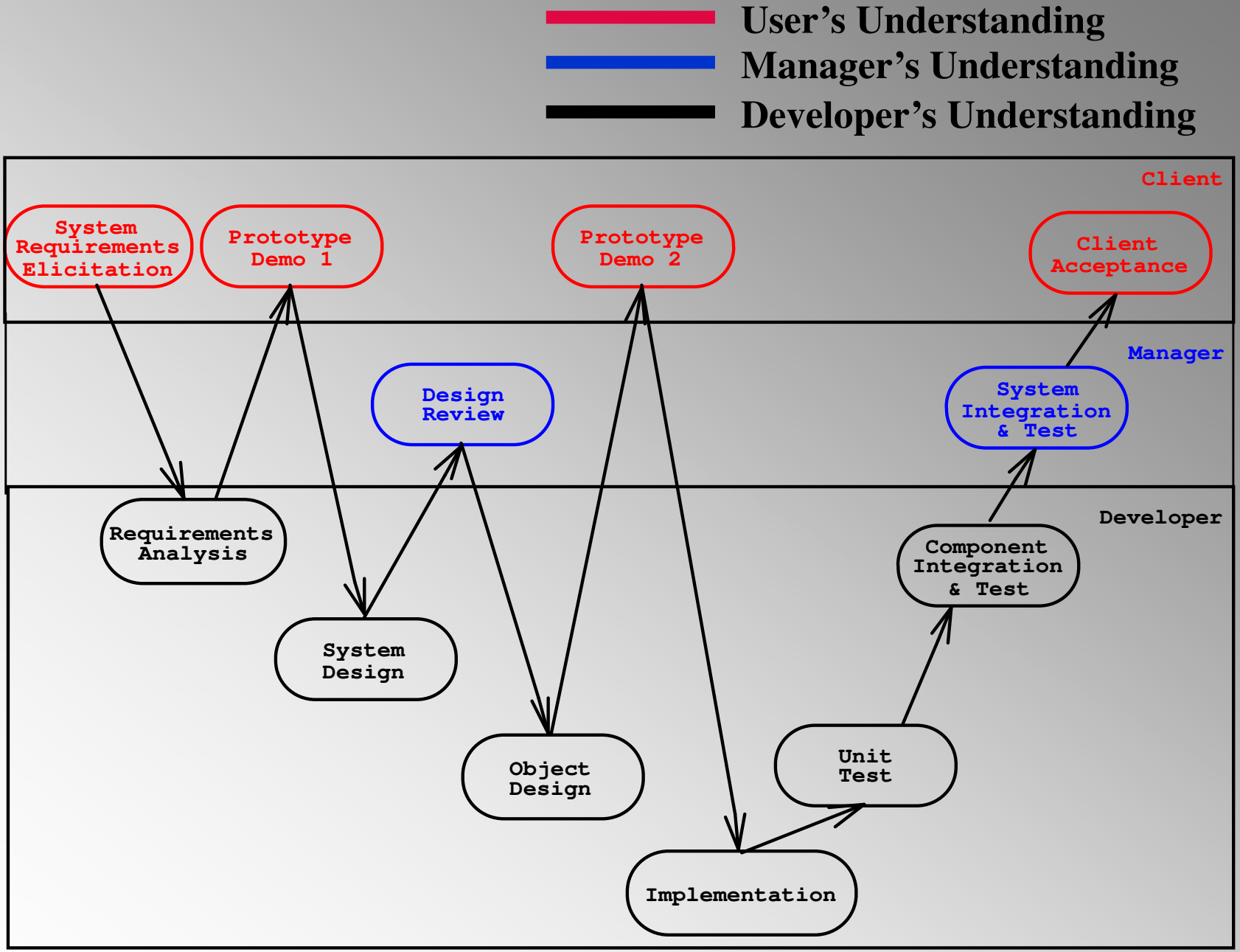
V Model: Distinguishes between Development and Verification Activities



Sawtooth Model



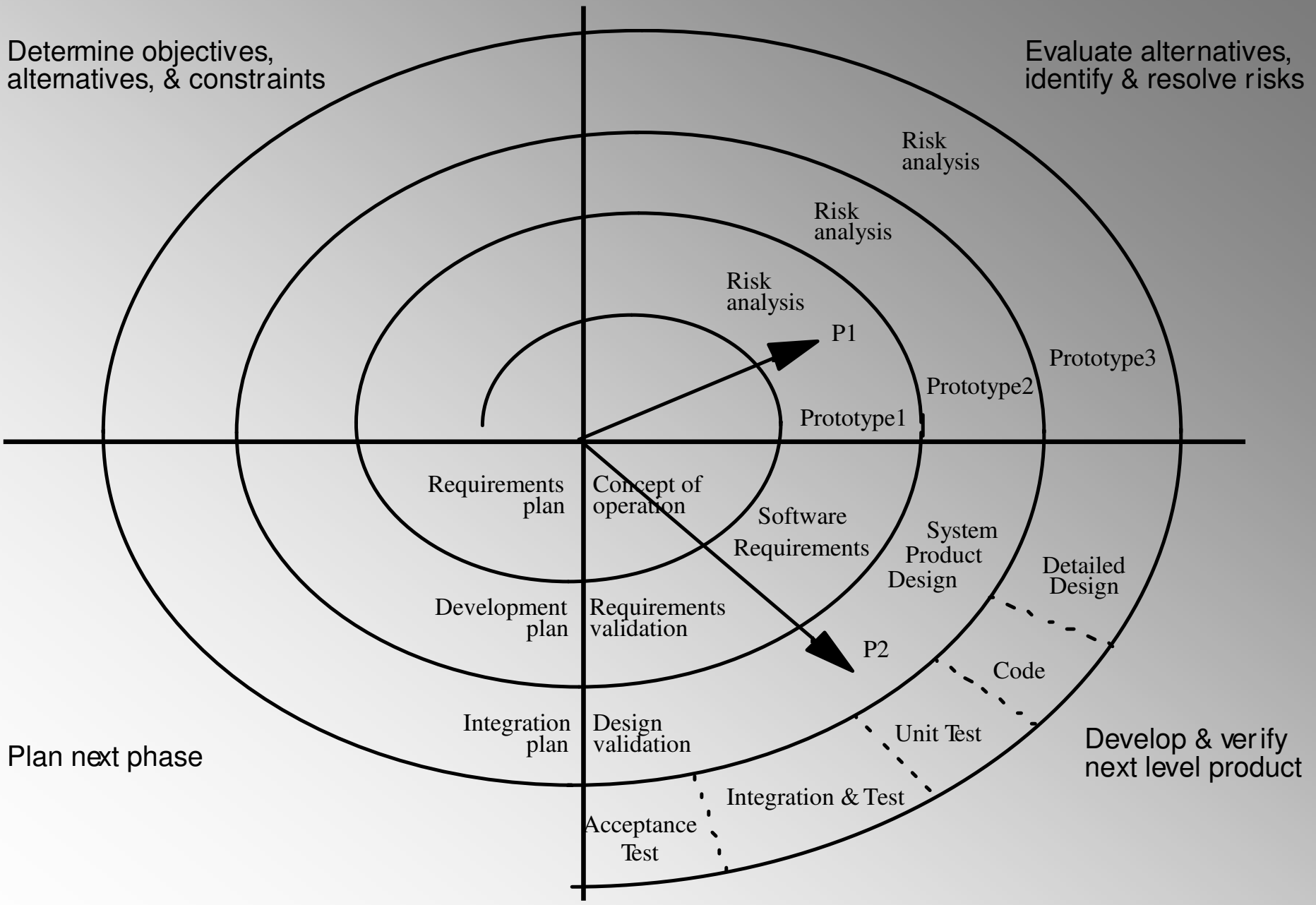
Sharktooth Model



Problems with Waterfall Model

- Managers love waterfall models:
  - **Nice milestones**
  - **No need to look back (linear system), one activity at a time**
  - **Easy to check progress : 90% coded, 20% tested**
- Different stakeholders need different abstractions
  - **=> V-Model**
- Software development is iterative
  - **During design problems with requirements are identified**
  - **During coding, design and requirement problems are found**
  - **During testing, coding, design & requirement errors are found**
  - **=> Spiral Model**
- System development is a nonlinear activity
  - **=> Issue-Based Model**

Spiral Model



Spiral Model (Boehm) Deals with Iteration

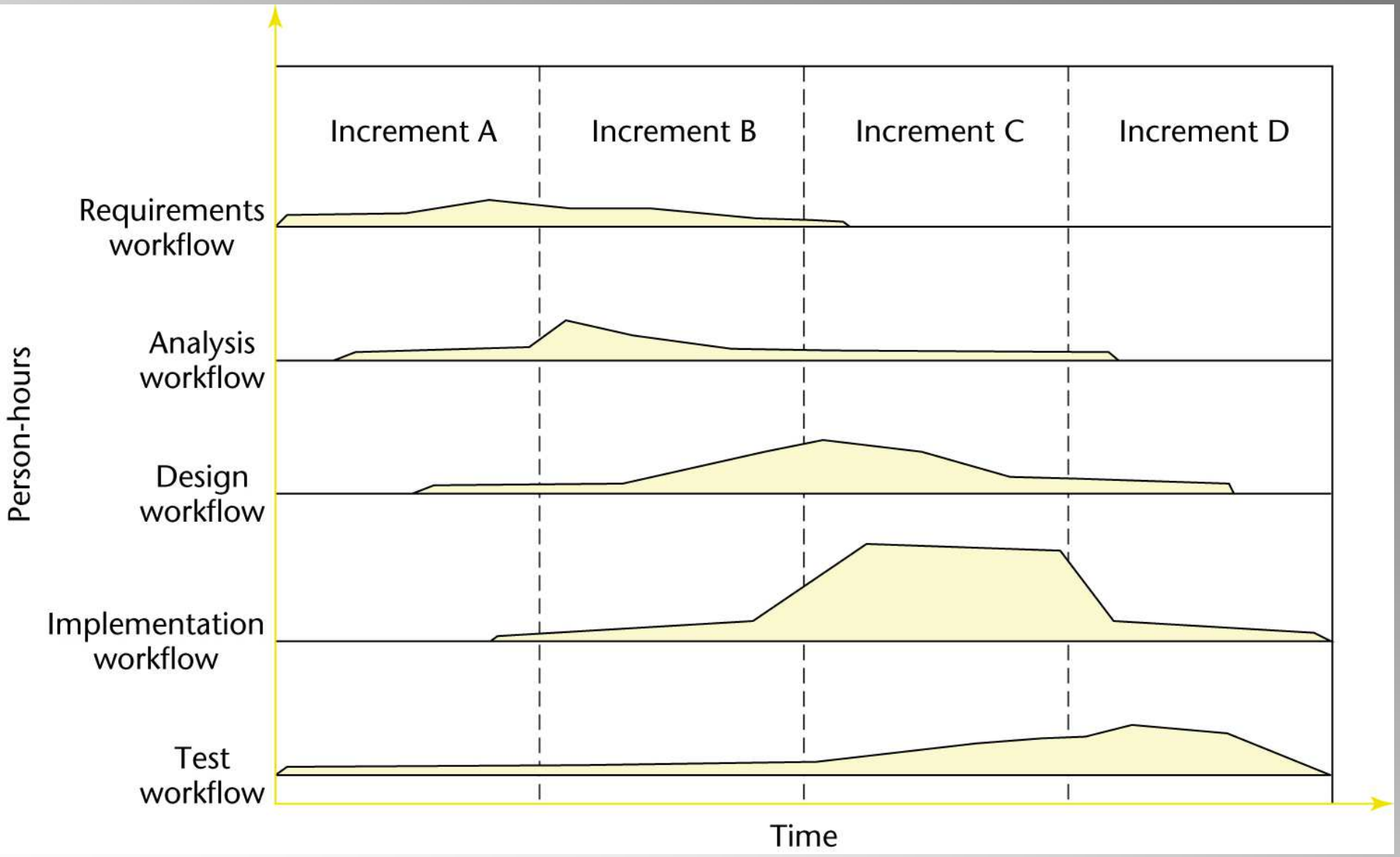
- Identify risks
- Assign priorities to risks
- Develop a series of prototypes for the identified risks starting with the highest risk.
- Use a waterfall model for each prototype development (“cycle”)
- If a risk has successfully been resolved, evaluate the results of the “cycle” and plan the next round
- If a certain risk cannot be resolved, terminate the project immediately

Activities ("Rounds") in Boehm's Spiral Model

- Concept of Operations
- Software Requirements
- Software Product Design
- Detailed Design
- Code
- Unit Test
- Integration and Test
- Acceptance Test
- Implementation

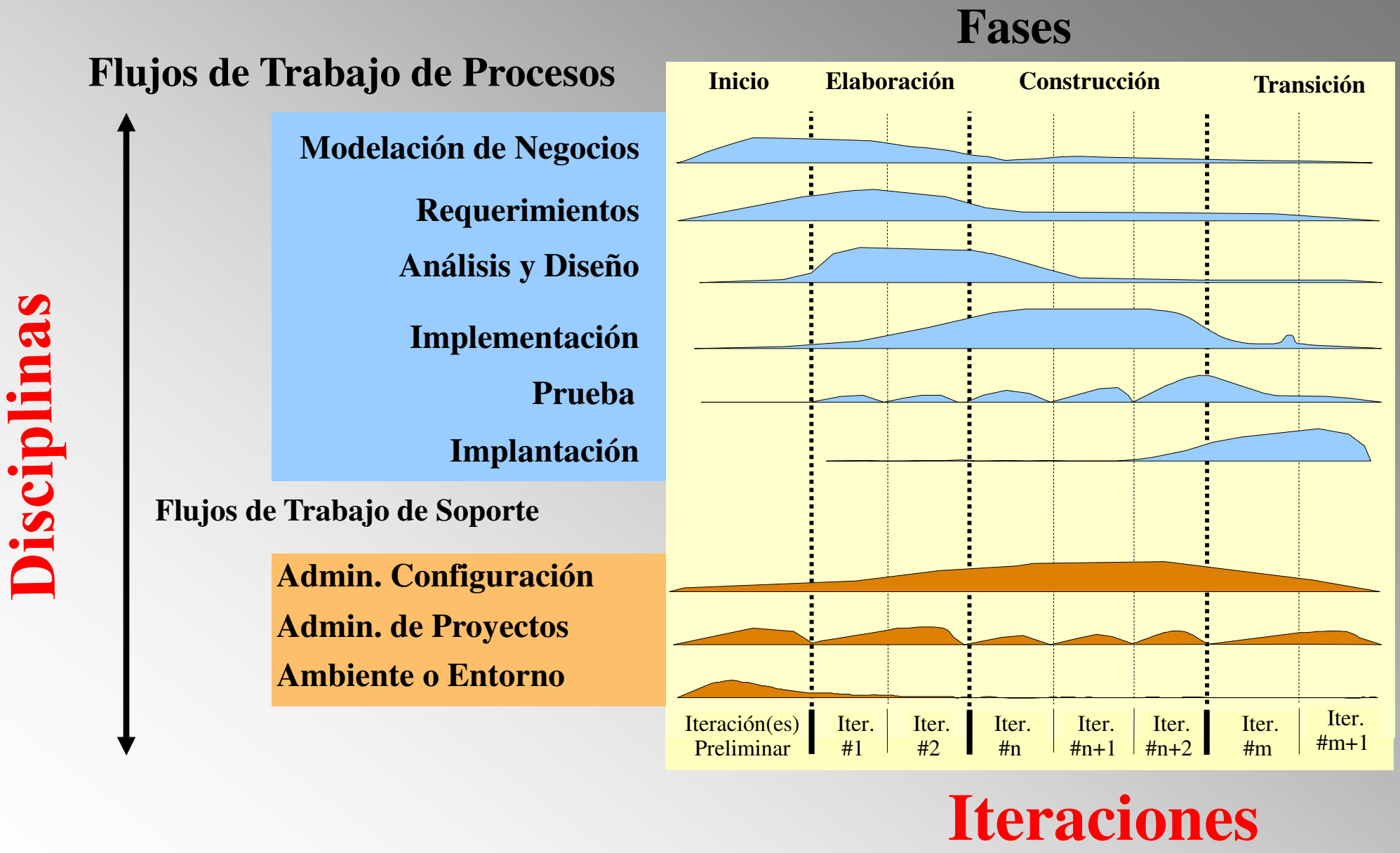
- For each cycle go through these steps
  - **Define objectives, alternatives, constraints**
  - **Evaluate alternative, identify and resolve risks**
  - **Develop, verify prototype**
  - **Plan next "cycle"**

Iterative and Incremental Life-Cycle Model



- Sample life cycle of an information system

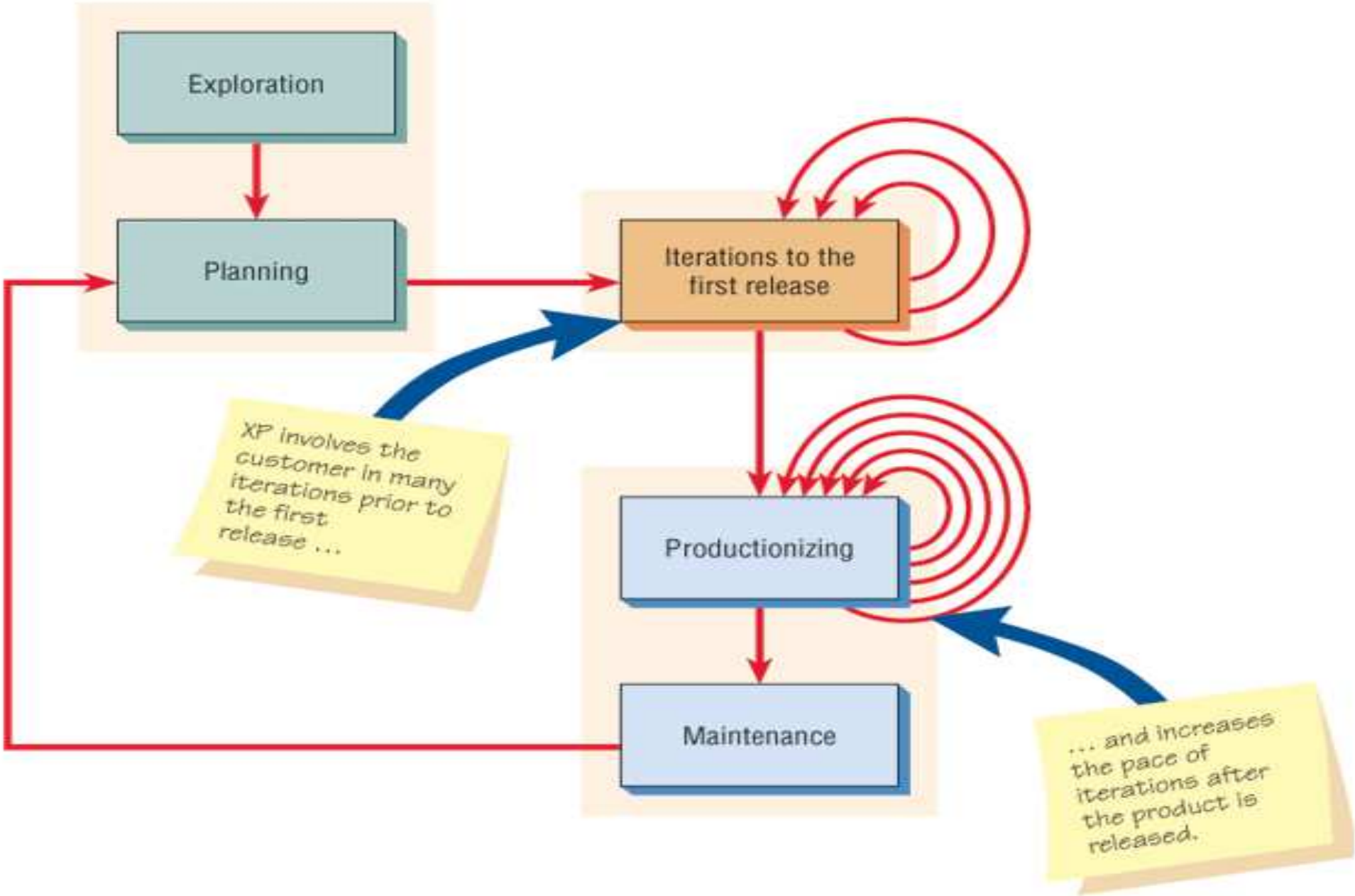
Unified Process (UP)



Agile Methods

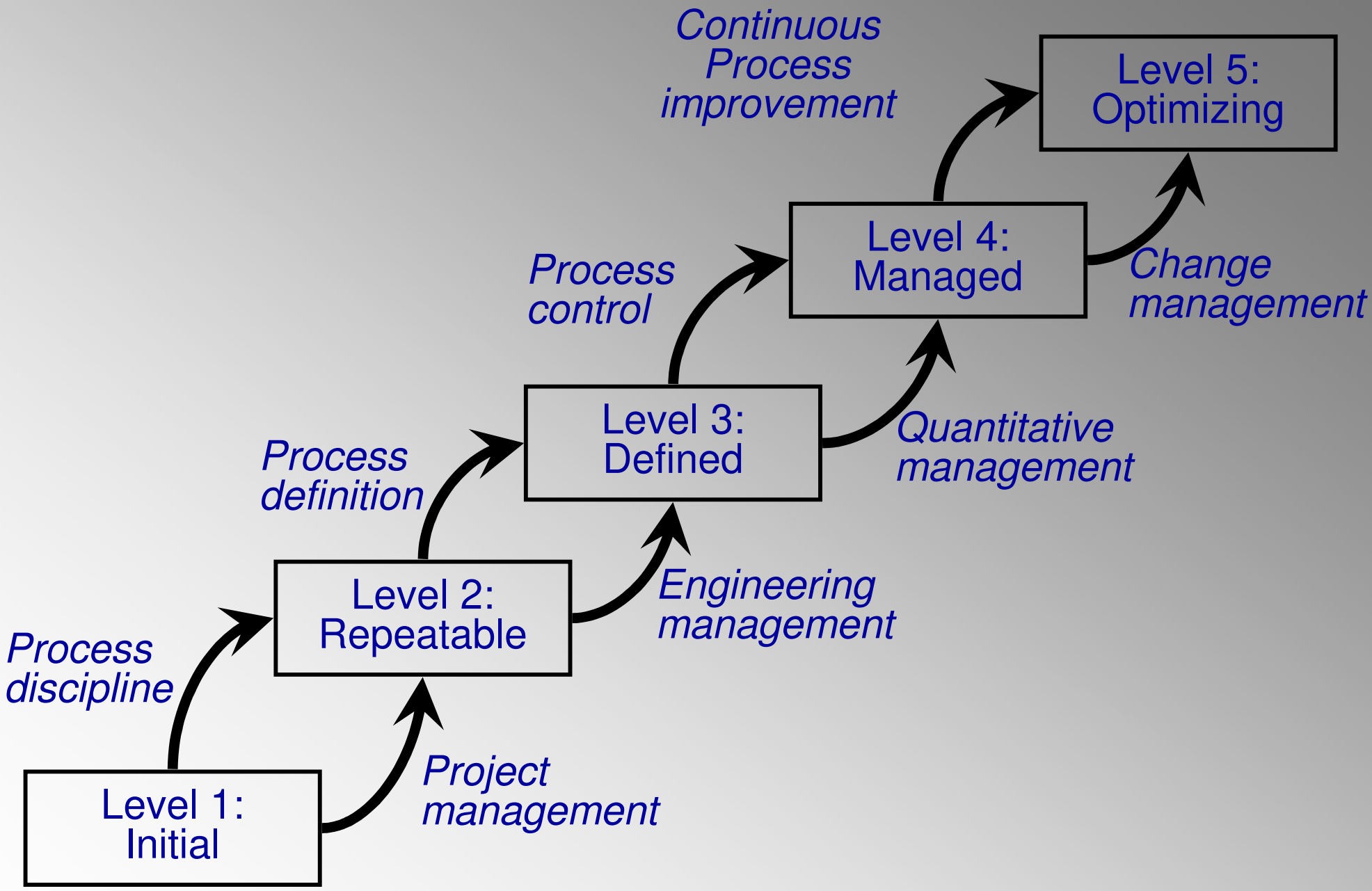
- RAD [Martin]
- Agile Manifesto [17 gurues]
- XP [Beck, Cunningham]
- FDD
- User Centered
- Cristal [Cockburn]
- SCRUM

The five stages of the XP development process shows that frequent iterations are essential to successful system development.



Process *Helpers*

- CMM
- ITIL
- PSP/TSP



The Software Engineering Institute's levels of maturity.

Summary by FRoM

A modern life cycle

- Use iteration among processes
- Achieve software incrementally
- Include risk analysis
- Work in priority order
- Use any kind of prototypes

The user participation is a very sensible issue

- Very intensive (eXtreme Programming)
- Very missed (the reality)

¿What can we do or What can we use in order to fix this issue?