

# ***El Proceso de Desarrollo de Software***



# La Ingeniería del Software

## ■ Ingeniería...

- “La **profesión** en la que el **conocimiento** de las ciencias naturales y matemáticas, ganado con estudio, experiencia y práctica, es **aplicado** con buen juicio para **desarrollar** formas de utilizar, económicamente, los materiales y las fuerzas de la naturaleza para el beneficio del género humano” (Accreditation Board for Engineering and Technology, 1996).

## ■ Peculiaridades de la Ingeniería del Software

- El “producto” software.
- Mucho desarrollo, poca disciplina ingenieril.
- Necesidad de describir y documentar lo que se va a producir.
- Cambios frecuentes en el producto.

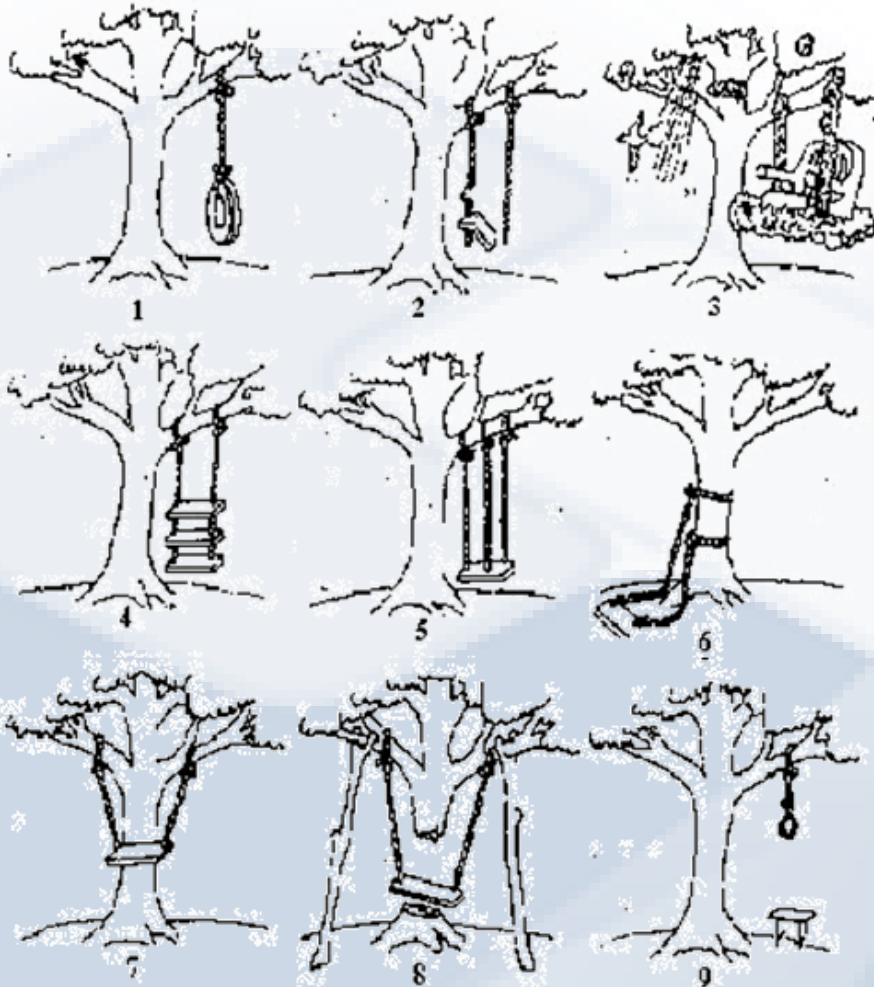
# *La Ingeniería del Software (y II)*

- Ingeniería de Sistemas vs. Ingeniería del Software
  - La Ingeniería de Sistemas está relacionada con todos los aspectos asociados al desarrollo de sistemas complejos: hardware, software, etc.
  - Los sistemas intensivos software son sistemas constituidos principalmente por software.
- Informática (Computer Science) vs. Ingeniería del Software
  - La Informática está relacionada con las teorías y métodos subyacentes a los ordenadores y a los sistemas software.

# Problemas en el Desarrollo de Software

Problemas	Soluciones
El programa no hace lo que se supone que debe hacer	Definir bien las especificaciones
El programa se cuelga	Definir bien el diseño y las pruebas
Hace falta un 200% del tiempo planificado para concluir el programa	Planificar bien
Si el programador se va de la empresa... ¡Vaya! Hay que hacer un nuevo programa	Definir bien la documentación
...	...

# El Típico Problema de Incomunicación

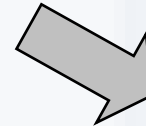
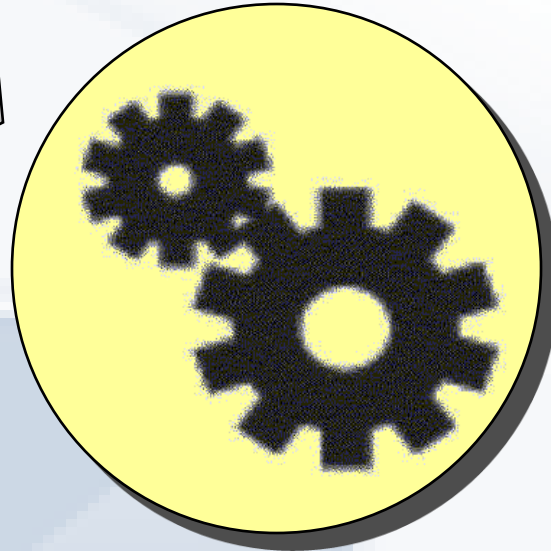


1. **Necesidad:** lo que el cliente realmente quería.
2. **Cliente:** lo que fue capaz de describir como una clara necesidad.
3. **Proceso de ventas:** lo que el fabricante de software prometió al cliente.
4. **Requisitos:** los requisitos descritos por el cliente, tal como finalmente fueron entendidos.
5. **Análisis:** la especificación formal de los requisitos realizada por los analistas.
6. **Diseño:** la especificación del funcionamiento del sistema para satisfacer los requisitos analizados.
7. **Codificación:** lo que escribió el programador.
8. **Instalación:** lo que realmente fue instalado al cliente.
9. **Pruebas:** lo que los responsables vieron en el sistema.

# El Proceso de Desarrollo Software



Requisitos, necesidades



¿Cómo se hace un proyecto software?

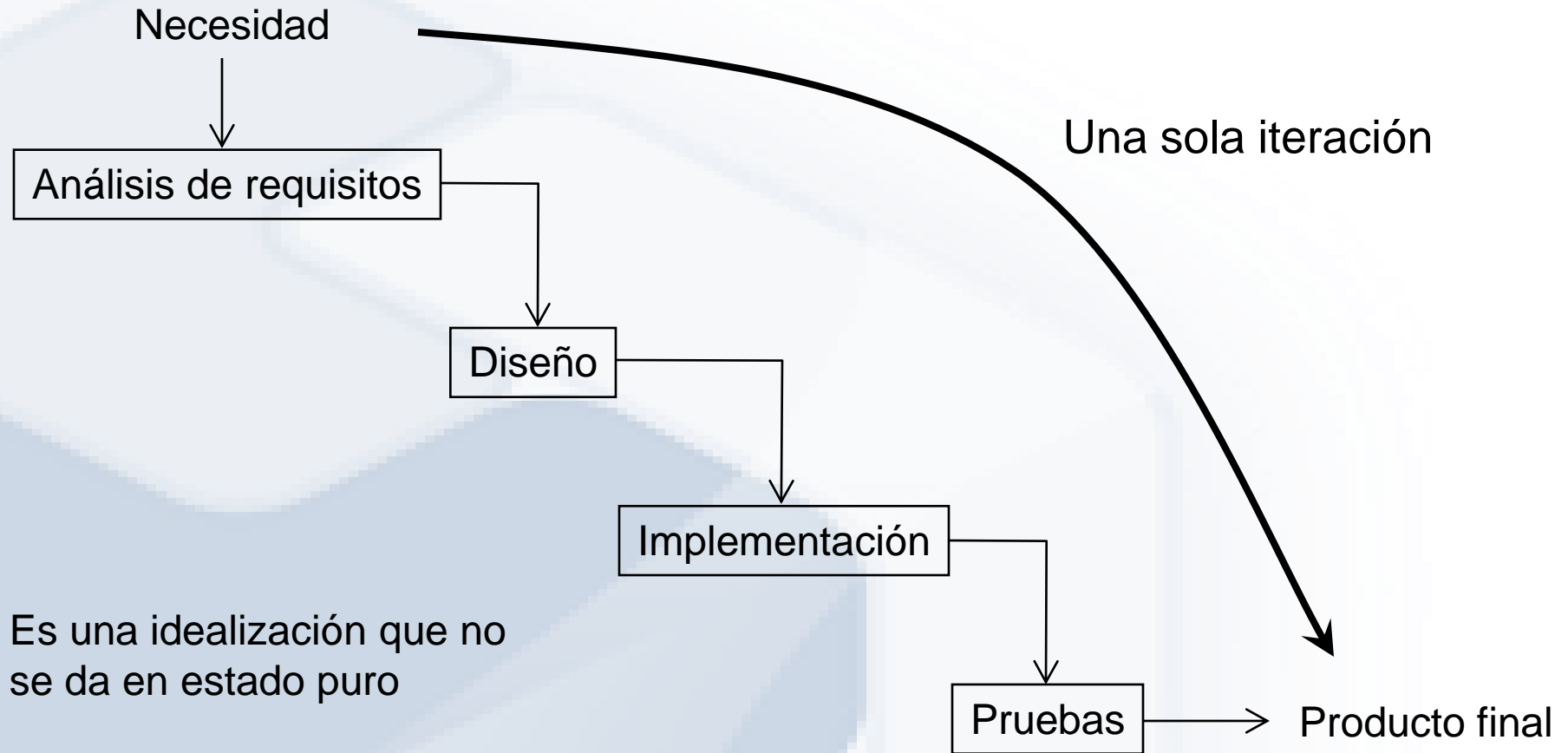
**Sistema software**

Diseño de Software Avanzado  
Departamento de Informática

# El Proceso de Desarrollo Software (II)

- El **SDP** define el **qué, quién, cuándo y cómo** del desarrollo de software.
- Cuatro actividades fundamentales que son comunes para todos los procesos de desarrollo de software :
  - **Especificación del software**
  - **Desarrollo del software**
  - **Validación del software**
  - **Evolución del software**
- Modelo de proceso:
  - Descripción simplificada (**abstracción**) de un proceso de desarrollo de software real.

# Ejemplo: Proceso en Cascada

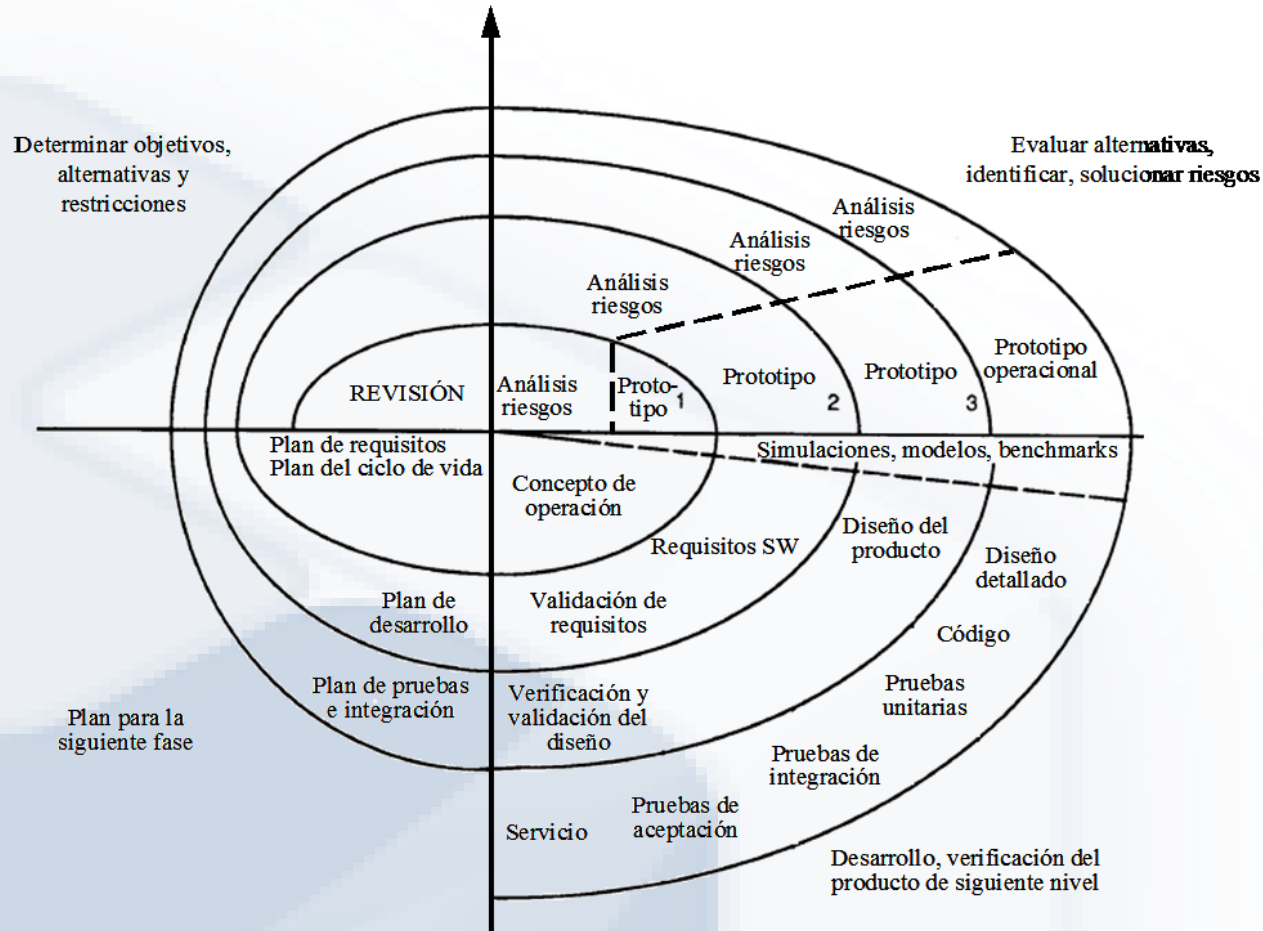




# El Proceso de Desarrollo Software (y III)

- Es un problema complejo para proyectos medios o grandes:
  - Dividir un proyecto en **mini-proyectos**, más fáciles de manejar y completar.
  - Cada mini-proyecto es una **iteración**.
  - Cada iteración contiene **todos los elementos** de un proyecto normal:
    - planificación
    - análisis y diseño
    - construcción
    - integración y pruebas
    - versión del producto (interna o externa)
  - Cada iteración genera una **línea base** (baseline) que comprende una versión parcialmente completa del sistema final, y toda la documentación asociada.
  - Las sucesivas iteraciones se construyen unas sobre otras hasta que se alcanza el sistema final terminado.
  - La diferencia entre dos líneas base se conoce como **incremento**.

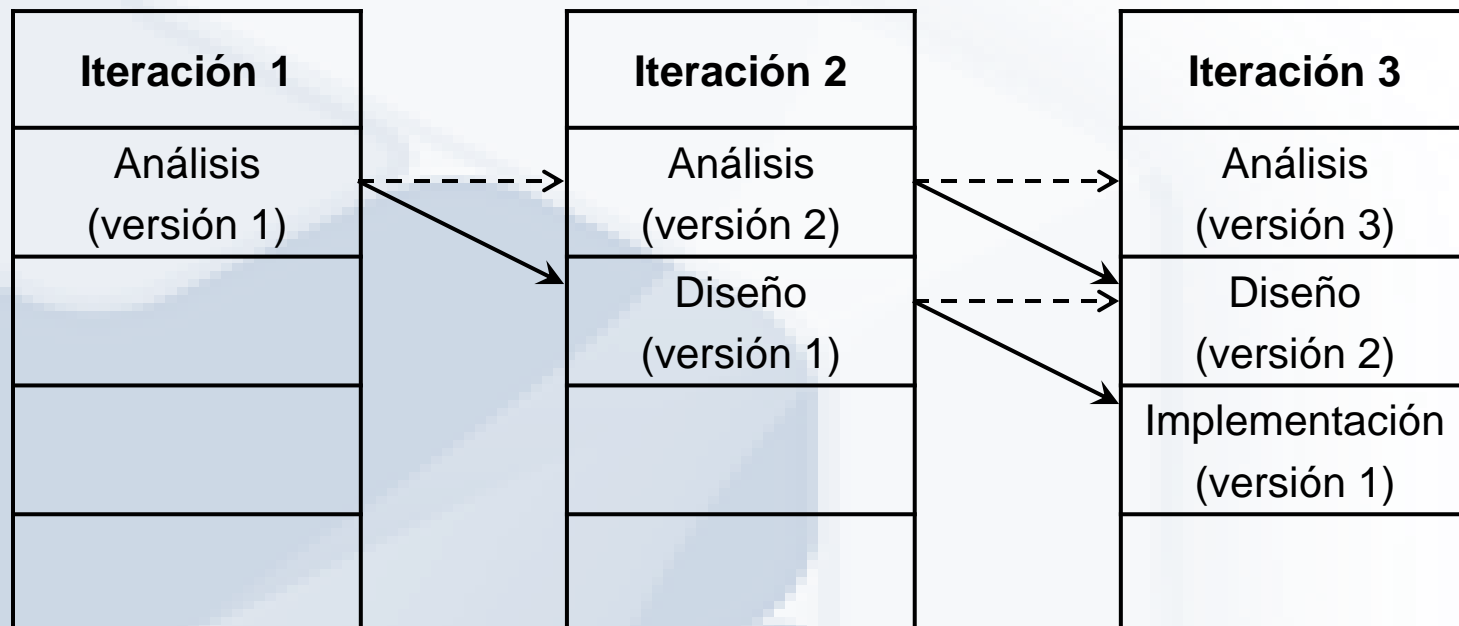
# Ejemplo: Proceso en Espiral



Espiral de Boehm (©IEEE, 1988)

# Ejemplo: Proceso Iterativo e Incremental

- Posibilita la **evolución en paralelo** de los distintos flujos de trabajo, y por tanto el trabajo en paralelo de distintos equipos de personas.
- Las distintas versiones de los documentos producidas en cada iteración no necesariamente son compatibles entre sí: **organizar bien la documentación.**



# El Proceso Unificado de Desarrollo de Software

- Ligado en su origen histórico a los trabajos de **Ivar Jacobson** en Ericsson (1967), Objectory (1987) y Rational (1997).
- Principios del USDP/UP.
  - Dirigido por **casos de uso y riesgos**
    - Capturar requisitos.
    - Analizar riesgos y enfrentarse a ellos.
  - Centrado en la **arquitectura**
    - La arquitectura de un sistema describe la estrategia de división del sistema en componentes, cómo estos componentes interaccionan, y cómo son desplegados en el hardware.
    - Una arquitectura de calidad garantiza un sistema de calidad, no una mera colección de piezas mal conectadas.
  - **Iterativo e incremental**
    - Proceso dividido en sucesivas iteraciones.
    - La diferencia entre dos iteraciones es un incremento.

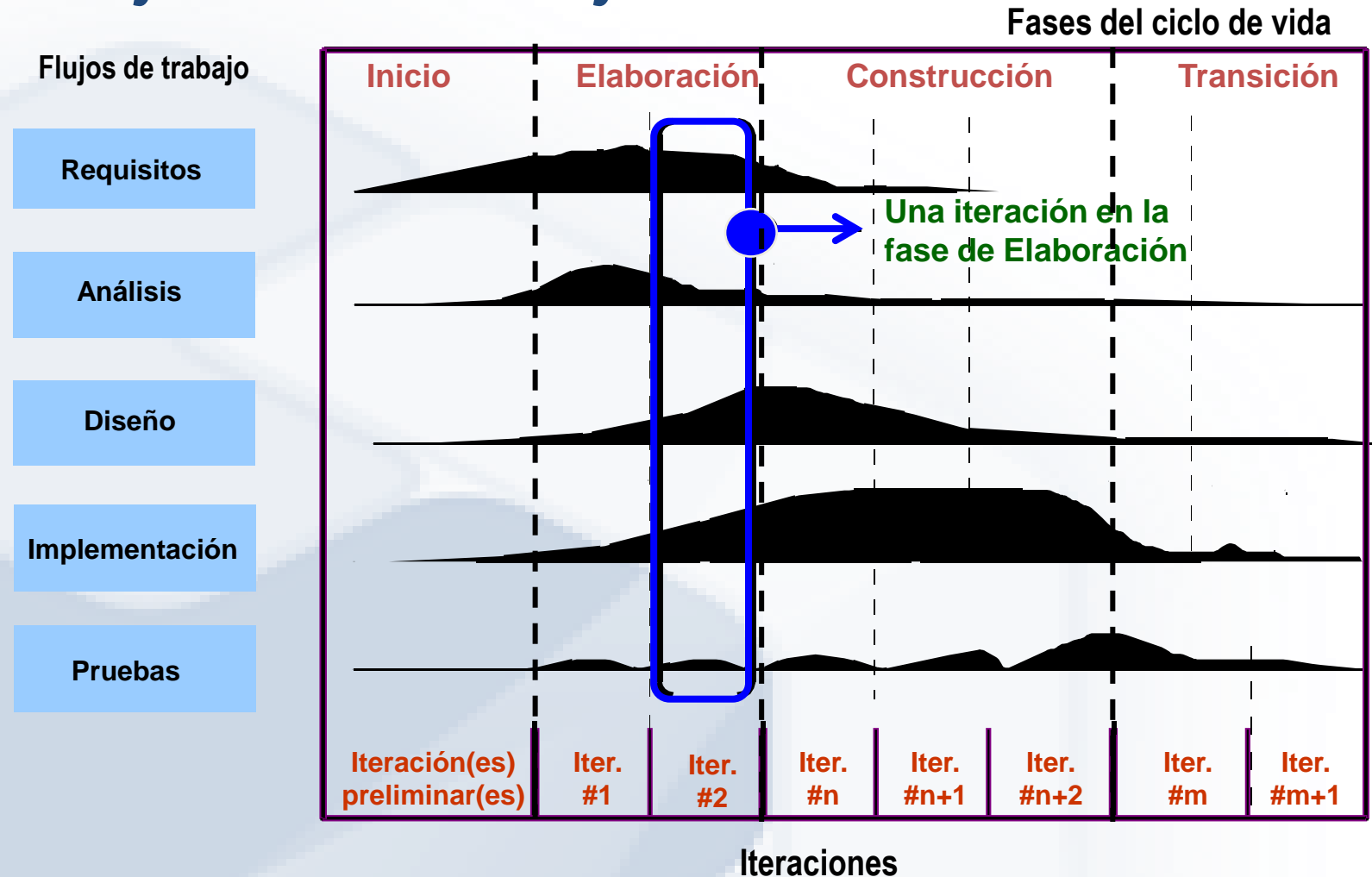
# USDP: Flujos de Trabajo

- En cada iteración existen **cinco flujos de trabajo** (workflows):
  - **Requisitos**: capturar lo que el sistema debe hacer.
  - **Análisis**: refinar y estructurar los requisitos.
  - **Diseño**: realizar los requisitos en la arquitectura del sistema.
  - **Implementación**: construir el software.
  - **Pruebas**: verificar que la implementación funciona como se desea.
- Pueden existir otros (no cubiertos por USDP):
  - Planificación
  - Evaluación
  - Específicos del proyecto
- Distinto **peso relativo** en cada iteración.

# USDP: Fases

- El ciclo de vida del proyecto se divide en **cuatro fases**, cada una de las cuales termina con un hito (milestone):
  - **Inicio**: objetivos del proyecto.
  - **Elaboración**: arquitectura del sistema.
  - **Construcción**: capacidad operativa inicial.
  - **Transición**: entrega del producto.
- En cada fase puede haber una o más **iteraciones**.
- En cada iteración se ejecutan los cinco **flujos de trabajo** principales y los otros que sean necesarios.

# USDP: Fases, Iteraciones y Flujos de Trabajo



# Otros Enfoques y Nomenclaturas

Terminología USDP	Terminología clásica
Requisitos	Análisis de requisitos
Análisis	
Diseño	Diseño
Implementación	Implementación
	Integración
Pruebas	Pruebas

Eric Braude, Software Engineering. An Object-Oriented Perspective, John Wiley & Sons, 2001, p. 30.