

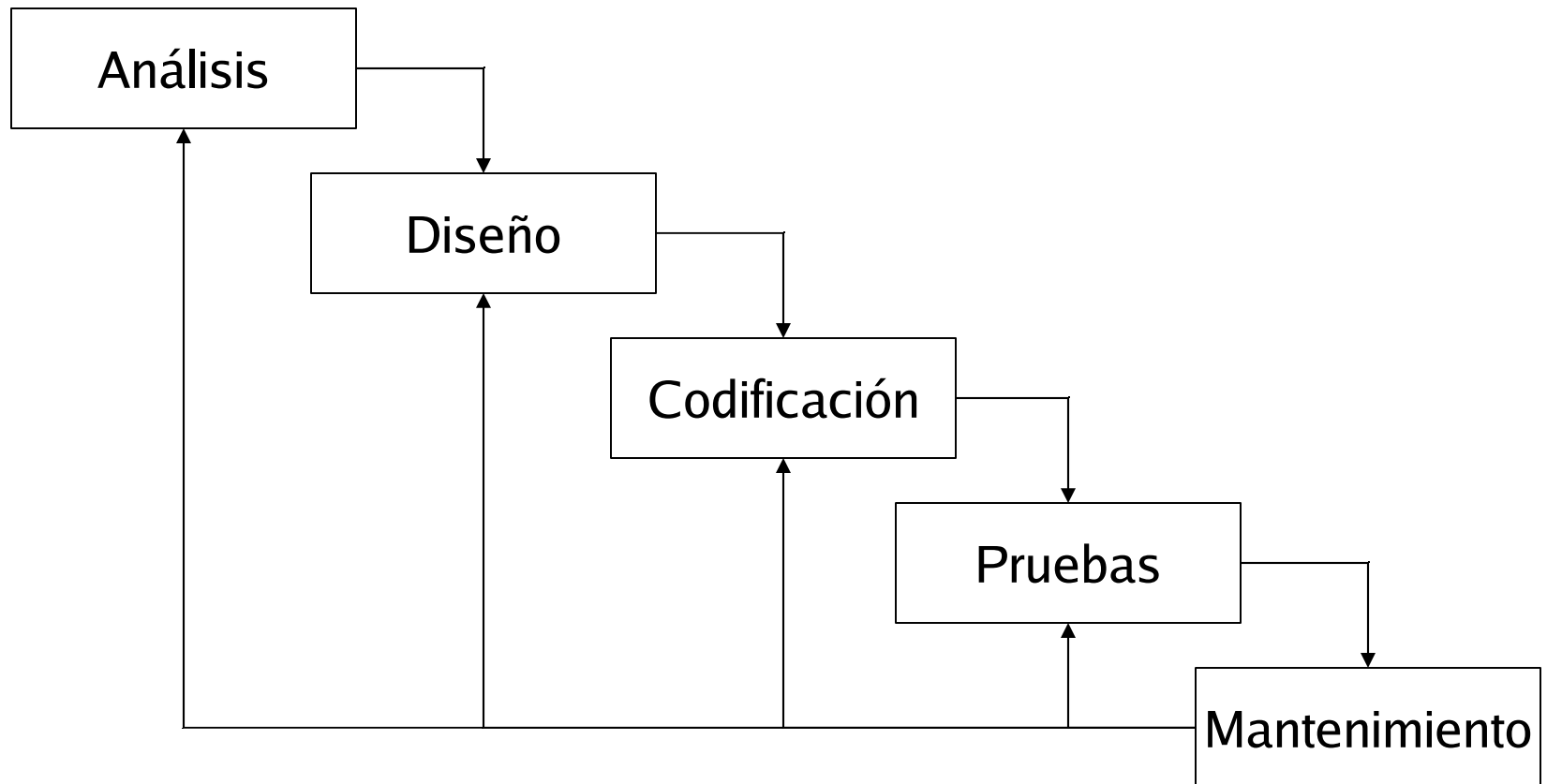
El Proceso de Desarrollo de Software

- Ciclos de vida
- Métodos de desarrollo de software
- El Proceso Unificado de Desarrollo de Software

Fases principales del desarrollo de software

- Captura de requisitos:
 - cuál es el problema?
- análisis:
 - qué debe hacerse? qué sistema debemos construir?
- diseño:
 - cómo podemos solucionar el problema?
- codificación:
 - trasladar el diseño a programas...
- pruebas:
 - ... que funcionen...
- implantación:
 - ... en un entorno productivo ...
- mantenimiento:
 - ... y que pueden estar sujetos a posibles modificaciones o mejoras posteriores!

Ciclo de vida en cascada



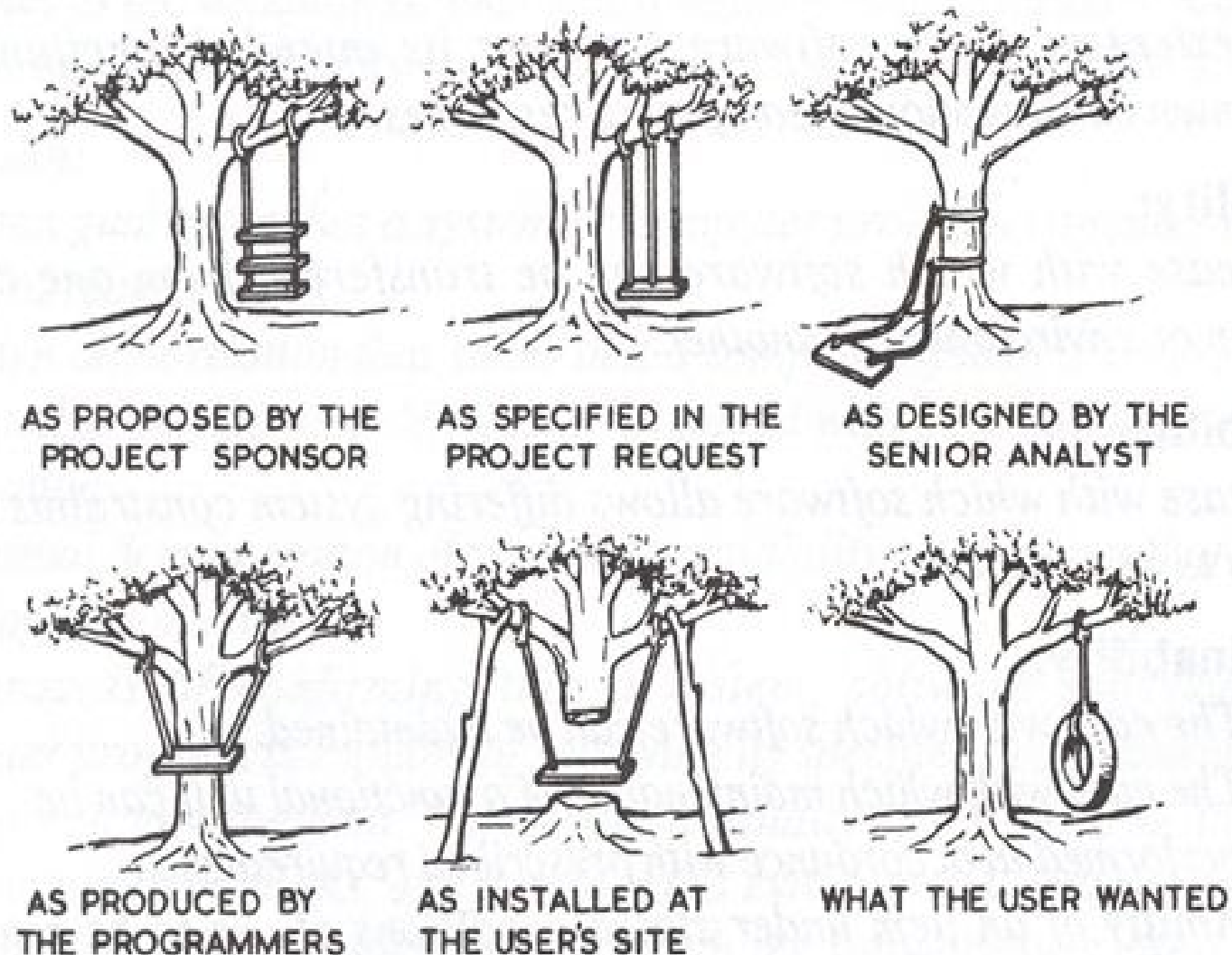


Figure 1.1. The problematics of software development [Brittan 80], © BCS 1980

Ciclo de vida en cascada

- A veces no se dispone de las especificaciones correctas ni completas al principio de la fase de desarrollo
 - Imprecisión del SI a desarrollar
 - Se identifican nuevas necesidades!
 - Falta de (buena) comunicación!
- + Se identifican y separan las distintas tareas
- + No se empieza una fase hasta terminar la anterior
- + Se pospone la implementación

Otros ciclos de vida

- Hay otras formas de organizar el ciclo de vida del desarrollo del software. Cada una tiene sus ventajas e inconvenientes.
- Modelo con Prototipos: se construye un sistema muy elemental, se muestra al cliente, se clarifican los requisitos, se mejora ... “extreme programming”,
- Modelo en Espiral: similar al anterior, pero con un ciclo de vida en cascada completo para cada prototipo.
- Modelo iterativo e incremental: construir un buen sistema pero de funcionalidad reducida, se entrega, se completan las funcionalidades extra.

IS como proceso formal

- Lo importante es formalizar el proceso.
- Un proyecto debe adoptar un patrón de desarrollo. Este patrón define las fases del ciclo de vida y sus entregas o productos.
- Este proceso da estructura al proyecto, ayudando a su gestión y proporcionando una guía a sus desarrolladores.
- También proporciona una forma natural de comparar proyectos (y problemas) y aprender de la experiencia.

Métodos de desarrollo de IS

- Se han propuesto varios métodos de IS.
- Un método debería contemplar:
 - un proceso de desarrollo
 - un lenguaje de notación
 - Una colección de herramientas CASE (Computer-Aided Software Engineering)
- Un método debería ser suficientemente general para permitir su adaptación a un problema particular o una metodología de programación concreta.

IS Orientada a Objetos (SIOO)

- Se han propuesto distintos métodos SIOO.
- Los tres métodos principales han sido:
 - el método Booch
 - El método OMT (Object Modelling Technique) desarrollado por Rumbaugh *et. al.*
 - el método Jacobson (OOSE)
- Cada uno de ellos propone un proceso, una notación y sus herramientas de soporte.

El método OMT

- La fase de análisis elabora tres modelos distintos:
 - El modelo objeto:
 - cuáles son las clases y objetos del sistema y cómo están relacionados?
 - El modelo dinámico:
 - cómo interaccionan los objetos entre ellos?
 - El modelo funcional:
 - cómo fluyen los datos a través del sistema, y qué restricciones debe satisfacer el sistema?

El método Booch

- Se da énfasis a la evolución: el resultado de cada fase es refinada en la siguiente fase.
- Los documentos Booch incluyen:
 - Diagramas de clases y objetos
 - Diagramas de transición entre estados
 - Diagramas de interacción
- Los programas Orientados a Objetos encajan bien con un desarrollo evolutivo.
 - Los objetos del mundo real mencionados en los requisitos evolucionan hasta convertirse en objetos en la implementación.
 - A medida que avanza el desarrollo se detallan (completan) mejor los objetos y las relaciones entre ellos.

El Lenguaje UML (Unified Modelling Language)

El Lenguaje Unificado de Modelado (UML) es un lenguaje para especificar, visualizar, construir y documentar los sistemas software, así como para el modelado del negocio y otros sistemas no software [OMG 01]

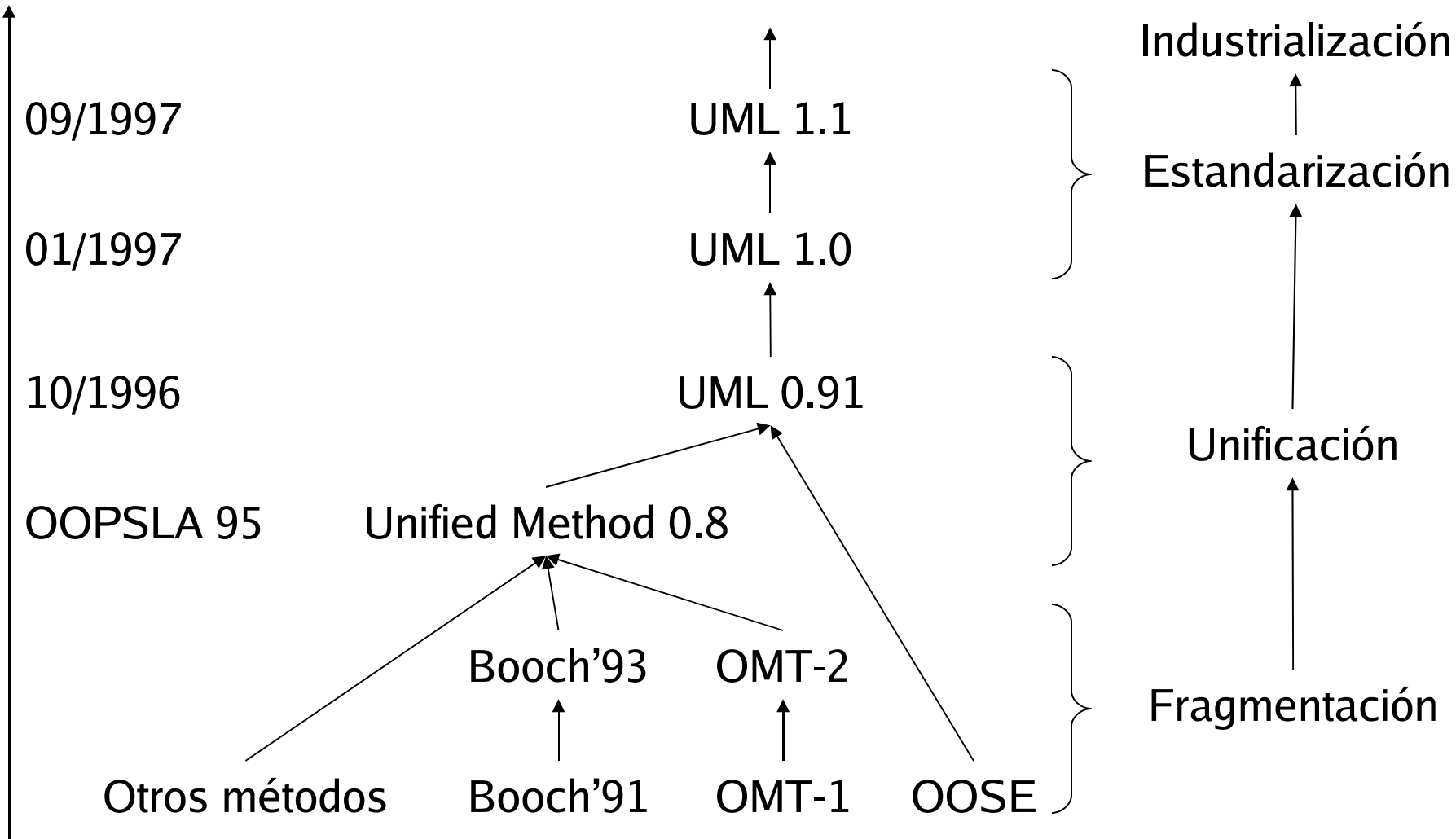
UML no es un método!

UML no es análisis y diseño orientado a objetos!

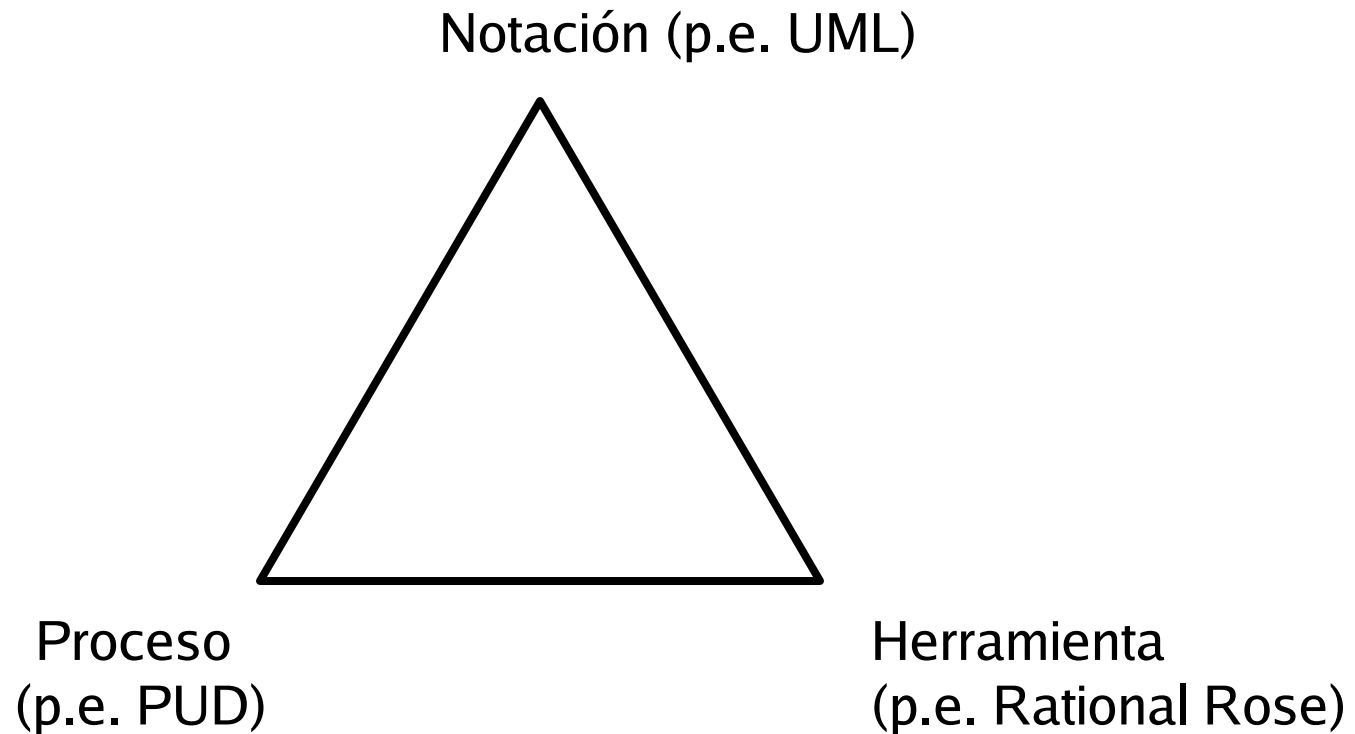
UML es una notación!

UML fue adoptado en 1997 como estándar por el OMG (Object Management Group).

El Lenguaje UML (Unified Modelling Language)



El triángulo del éxito



El Proceso Unificado de Desarrollo

- Proceso de desarrollo que combina un conjunto de “buenas prácticas”:
 - Orientado a *objetos*
 - Guiado por *casos de uso*
 - Desarrollo dirigido por el *riesgo*
 - Centrado en la *arquitectura*
 - Con un ciclo de vida *iterativo e incremental*

El Proceso Unificado de Desarrollo de Software

- Guiado por casos de uso
 - Los sistemas se crean para dar servicio a usuarios
 - Capturan requisitos funcionales:
 - qué debe hacer el sistema para cada usuario?
 - Un Caso de Uso es una funcionalidad del sistema que le proporciona a algún usuario un resultado
 - El Modelo de Casos de Uso (MCU) describe la funcionalidad completa del sistema

El Proceso Unificado de Desarrollo de Software

- Desarrollo dirigido por el riesgo
 - Abordar cuestiones de alto riesgo (complejas, vagas, etc.) en las primeras iteraciones
 - Construir en las primeras iteraciones un núcleo consistente

El Proceso Unificado de Desarrollo de Software

- Centrado en la arquitectura
 - La arquitectura del sistema da una idea de qué forma tiene el sistema completo
 - Conviene pensar en la forma del sistema que se está construyendo: ordenadores necesarios, SOs, SGBDs, comunicaciones, etc.

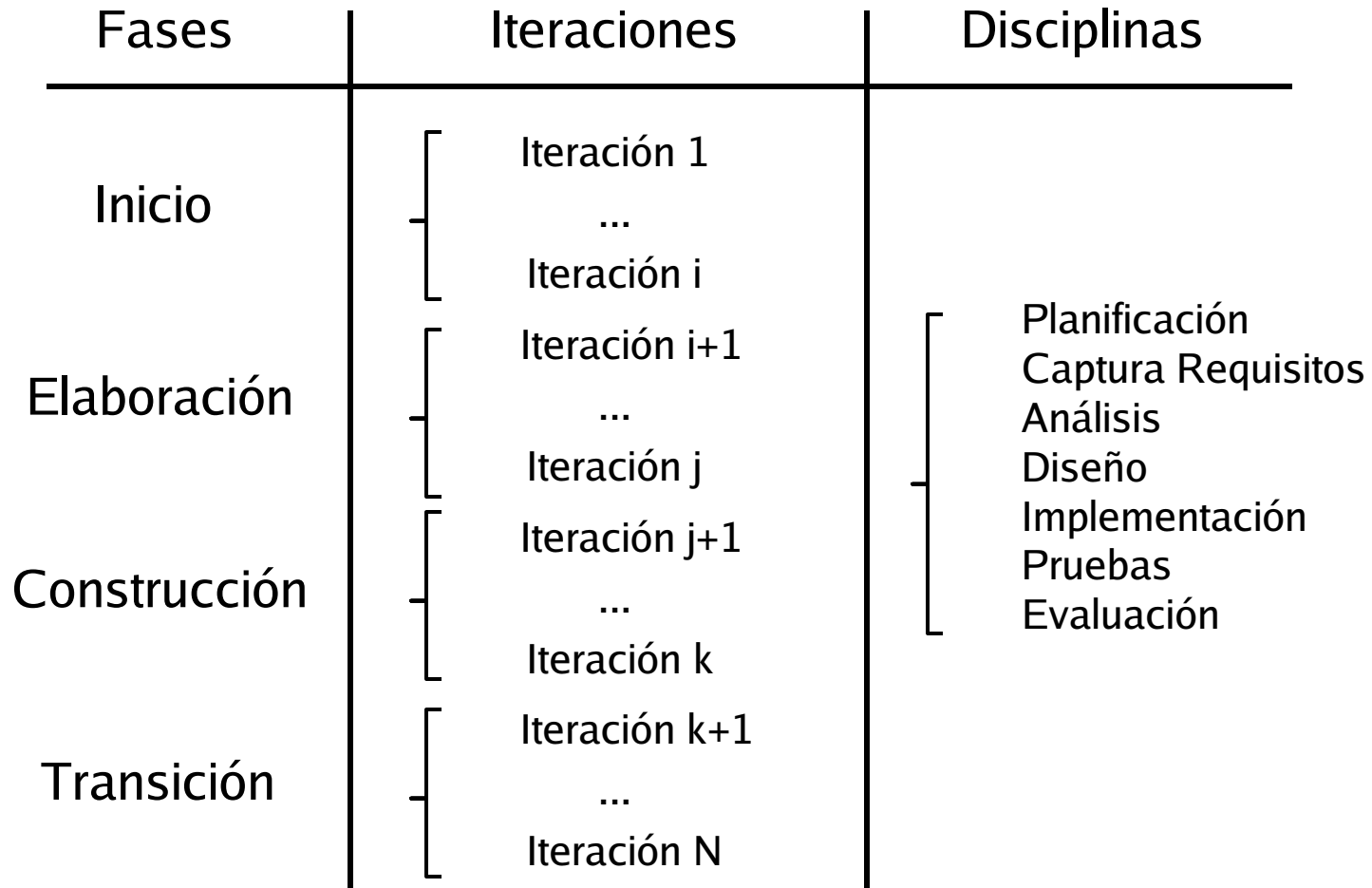
El Proceso Unificado de Desarrollo de Software

- Ciclo de vida iterativo e incremental
 - El desarrollo de un proyecto se divide en iteraciones
 - Cada iteración es como un mini proyecto
 - Tiene una duración fija (p.e. de 2 a 6 semanas)
 - Incluye sus fases de captura de requisitos, análisis, diseño, implementación, pruebas ...
 - Cada iteración produce un resultado mejorado:
 - Más detallado
 - Añade alguna nueva funcionalidad
 - Puede ser probado, integrado, ejecutado, etc.
 - Es incompleto

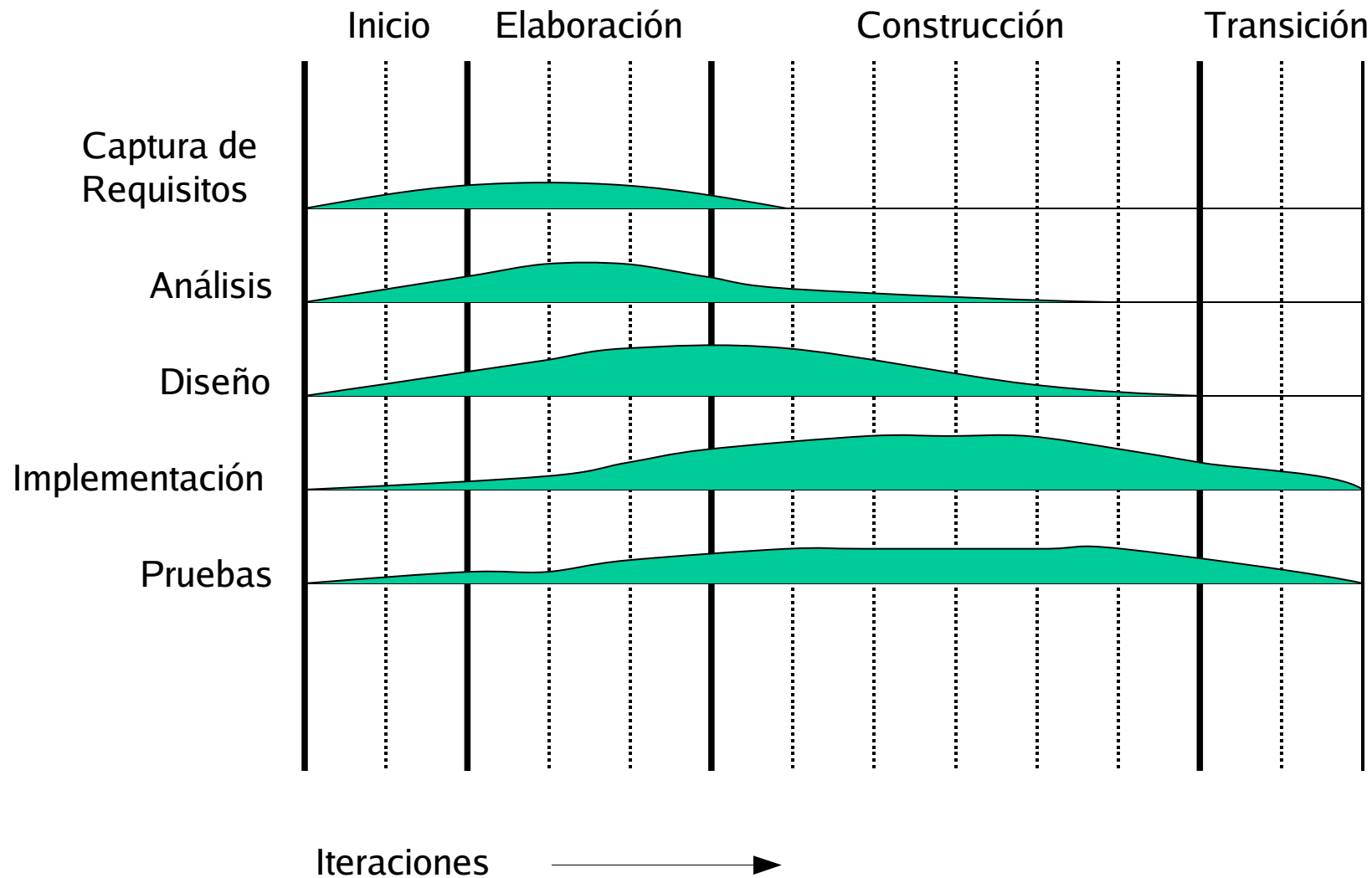
Fases del Proceso Unificado de Desarrollo

- El PUD agrupa las iteraciones en:
 - Inicio: análisis del negocio, alcance, visión aproximada
 - Elaboración: implementación iterativa del núcleo central del sistema, resolución de riesgos altos, visión refinada, identificación de más requisitos, nuevo alcance
 - Construcción: implementación iterativa del resto de requisitos de menor riesgo, preparación para la implantación
 - Transición: pruebas beta, implantación

Ciclo de vida del Proceso Unificado de Desarrollo



Ciclo de vida del Proceso Unificado de Desarrollo



Iteración y entregas en PUD

- Planificación de la iteración i-ésima
 - Captura de requisitos:
 - Modelo de casos de uso, Modelo de Dominio, ...
 - Análisis:
 - Diagrama de secuencia del sistema, Contratos, Modelo Conceptual...
 - Diseño:
 - Diagramas de interacción, Diagrama de Clases
 - Implementación:
 - Código fuente (Clases y métodos)
 - Pruebas:
 - verificación de la implementación
- Evaluación de la iteración i-ésima