

# TEMA 2

## Modelos de proceso del software

María N. Moreno García  
Departamento de Informática y Automática  
Universidad de Salamanca

# Contenidos

1. Conceptos básicos
2. Procesos del ciclo de vida
3. Modelos de proceso
  - Modelo clásico
  - Modelos iterativos basados en prototipos
  - Modelos en espiral
  - Desarrollo rápido de aplicaciones
  - Modelos orientados a la reutilización
  - Modelos para sistemas orientados a objetos
  - Procesos ágiles
  - Modelos de proceso de la Ingeniería Web

## Conceptos básicos

- **Proceso del software:** conjunto de actividades y resultados asociados que conducen a la creación de un producto software [Sommerville, 2002]
- **Ciclo de vida del software:** *Aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software* (norma IEEE 1074) [IEEE, 1999]  
El ciclo de vida incluye
  - Ciclo de desarrollo del sistema
  - Tiempo de vida del sistema
- **Modelo de ciclo de vida:** *Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso* (norma ISO 12207-1) [ISO/IEC, 1995]
  - Actividad: conjunto de tareas
  - Tarea: acción que transforma entradas en salidas

# Procesos del ciclo de vida (I)

## Norma ISO 12207-1 [ISO/IEC, 1995]

### ■ Procesos principales

- Adquisición
- Suministro
- Desarrollo
- Explotación
- Mantenimiento

### ■ Procesos de soporte

- Documentación
- Gestión de la configuración
- Aseguramiento de la calidad
- Verificación
- Validación
- Revisión conjunta
- Auditoría
- Resolución de problemas

### ■ Procesos de la organización (generales)

- Gestión
- Mejora
- Infraestructura
- Formación

### Nuevos procesos (*amendments* 2002-2005)

- Usabilidad
- Evaluación de productos
- Recursos Humanos
- Gestión de “assets”
- Gestión de petición de cambios
- Gestión del programa de reutilización
- Ingeniería del dominio

## Procesos del ciclo de vida (II)

### Norma ISO/IEC 12207:2008 [ISO/IEC, 2008]

- **Procesos del ciclo de vida del sistema**
  - **Procesos de acuerdo (*agreement*):** Adquisición, suministro
  - **Procesos de organización del proyecto:** Gestión del modelo de ciclo de vida, gestión de la infraestructura, gestión del porfolio del proyecto, gestión de recursos humanos, gestión de la calidad
  - **Procesos del proyecto:** Planificación del proyecto, control y evaluación del proyecto, gestión de las decisiones, gestión de riesgos, gestión de la configuración, gestión de la información, medición
  - **Procesos técnicos:** Definición de requisitos de los *stakeholders*, análisis de requisitos del sistema, diseño arquitectónico del sistema
- **Procesos específicos del software**
  - **Procesos de implementación del software:** análisis de requisitos, diseño arquitectónico, diseño detallado, construcción, integración, prueba
  - **Procesos de soporte del software:** Gestión de la documentación, gestión de la configuración, aseguramiento de la calidad, verificación, validación, revisión conjunta, auditoría, resolución de problemas
  - **Procesos de reutilización del software:** Ingeniería del dominio, gestión de *assets*, gestión del programa de reutilización

## Modelos de proceso

### ■ Modelos tradicionales

Formados por un conjunto de fases o actividades en las que no tienen en cuenta la naturaleza evolutiva del software

- Clásico, lineal o en cascada
- Estructurado
- Basado en prototipos
- Desarrollo rápido de aplicaciones (RAD)

### ■ Modelos evolutivos

Son modelos que se adaptan a la evolución que sufren los requisitos del sistema en función del tiempo

- En espiral
- Evolutivo
- Incremental
- Modelo de desarrollo concurrente

### ■ Modelos orientados a la reutilización

- Basado en componentes
- Proceso Unificado

### ■ Modelos para sistemas orientados a objetos

Modelos con un alto grado de iteratividad y solapamiento entre fases

- De agrupamiento
- Fuente
- Basado en componentes
- Proceso Unificado

### ■ Procesos ágiles

- Programación extrema (XP)
- Desarrollo de software adaptativo
- Scrum, Crystal ...

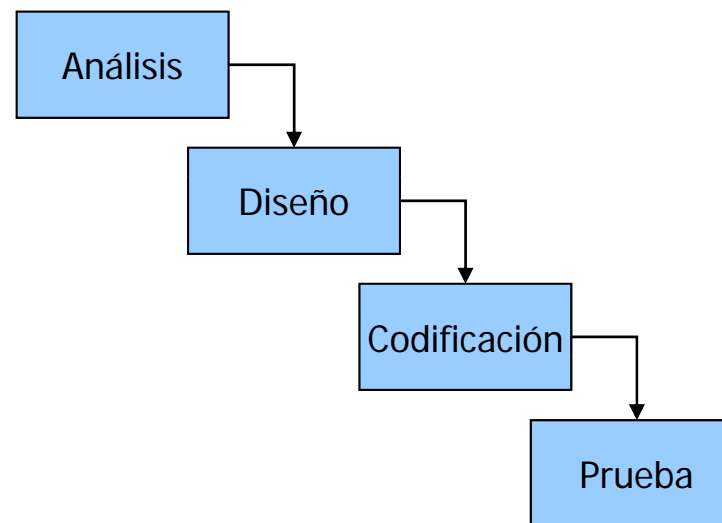
### ■ Modelos para sistemas web

- *UML-based Web Engineering*

# Modelos de proceso

## Modelo clásico (I)

- Conocido también como modelo lineal o “en cascada”
- Versión original se debe a W. W. Royce [Royce, 1970], apareciendo después numerosos refinamientos
  - Características
    - Está compuesto por una serie de fases que se ejecutan secuencialmente
    - Obtención de documentos como criterio de finalización de fase
  - Problemas de la progresión secuencial
    - Desconocimiento de las necesidades por parte del cliente
    - Inestabilidad de los requisitos
    - No se ven resultados hasta muy avanzado el proyecto
    - Efecto *big bang* próximo a la entrega

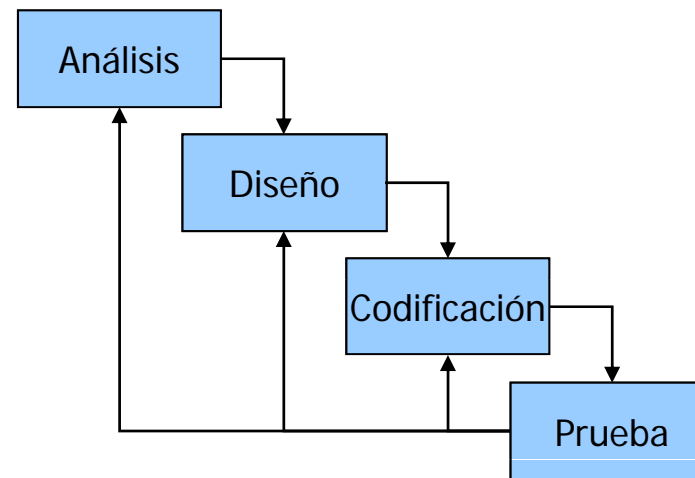


Ciclo de vida clásico

# Modelos de proceso

## Modelo clásico (II)

- Modelo satisfactorio sólo en desarrollos conocidos y estables
- El desconocimiento y el riesgo suele ser alto en el desarrollo del software
- La linealidad no se corresponde con la realidad
  - Los retornos de información entre las fases se hacen necesarios para incorporar correcciones hacia arriba, en función de los descubrimientos realizados hacia abajo
  - Estos retornos entre fases perturban la visión lineal dada por el ciclo de vida en cascada
  - Los retornos están limitados a fases adyacentes



Ciclo de vida clásico con realimentación



# Modelos de proceso

## Modelos iterativos basados en prototipos (I)

- Un prototipo es un modelo experimental de un sistema o de un componente de un sistema que tiene los suficientes elementos que permiten su uso
  - **Objetivos:**
    - Son un medio eficaz para aclarar los requisitos de los usuarios e identificar las características de un sistema que deben cambiarse o añadirse
    - Mediante el prototipo se puede verificar la viabilidad del diseño de un sistema
  - **Características:**
    - Es una aplicación que funciona
    - Su finalidad es probar varias suposiciones con respecto a las características requeridas por el sistema
    - Se crean con rapidez
    - Evolucionan a través de un proceso iterativo
    - Tienen un costo bajo de desarrollo

# Modelos de proceso

## Modelos iterativos basados en prototipos (II)

### ■ Tipos de prototipos

- **Prototipos desechables:** El prototipo es una versión rudimentaria del sistema que posteriormente es desechada
- **Prototipos evolutivos:** El prototipo debe convertirse, eventualmente, en el sistema final usado (alternativa al ciclo de vida)
- **Combinación de prototipos evolutivos y desechables** (prototipado operativo):
  - Se aplican técnicas convencionales para los requisitos bien conocidos y se crea una "línea base"
  - Combinación de prototipos desechables y evolutivos para los requisitos poco conocidos

	DESECHABLE	EVOLUTIVO
<b>Enfoque de desarrollo</b>	Rápido y sin rigor	Riguroso
<b>Qué construir</b>	Sólo las partes problemáticas	Primero las partes bien entendidas. Sobre una base sólida.
<b>Directrices del diseño</b>	Optimizar el tiempo de desarrollo	Optimizar la modificabilidad
<b>Objetivo último</b>	Desecharlo	Incluirlo en el sistema

Diferencias entre los prototipos desechables y evolutivos

# Modelos de proceso

## Modelos iterativos basados en prototipos (III)

### ■ **Prototipos desechables**

#### □ **Características:**

- Se desarrolla código para explorar factores críticos para el éxito del sistema
- La implementación usa lenguajes y/o métodos de desarrollo más rápidos que los definitivos
- Se usa como herramienta auxiliar de la especificación de requisitos y el diseño:
  - Determinar la viabilidad de los requisitos
  - validar la funcionalidad del sistema
  - Encontrar requisitos ocultos.
  - Determinar la viabilidad de la interfaz de usuario.
  - Examinar alternativas de diseño.
  - Validar una arquitectura de diseño particular

#### □ **Aplicaciones:**

- Interfaz de usuario
- Formatos de informes
- Formatos de gráficos
- Organización de bases de datos
- Rendimiento de bases de datos
- Precisión e implementación de cálculos complejos
- Partes con respuesta crítica en el tiempo en sistemas de tiempo real
- Rendimiento de sistemas interactivos
- Viabilidad de partes del sistema en las que no se tiene experiencia

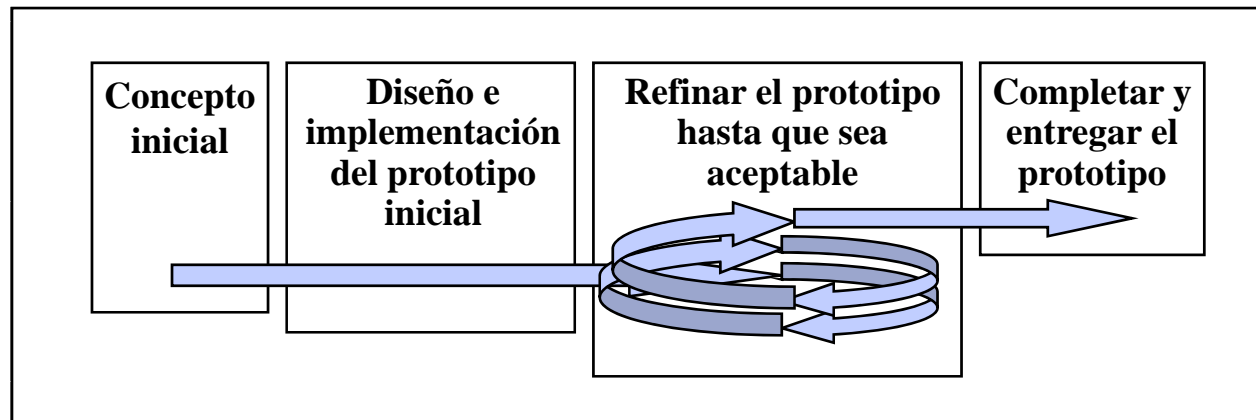
# Modelos de proceso

## Modelos iterativos basados en prototipos (IV)

### ■ **Prototipado evolutivo (ciclo de vida iterativo)**

#### □ **Características:**

- Enfoque de desarrollo que se utiliza cuando no se conoce con seguridad lo que se quiere construir
- Se comienza diseñando e implementando las partes más destacadas del sistema
- La evaluación del prototipo proporciona la realimentación necesaria para aumentar y refinar el prototipo
- El prototipo evoluciona y se transforma en el sistema final



Modelo de prototipado evolutivo

# Modelos de proceso

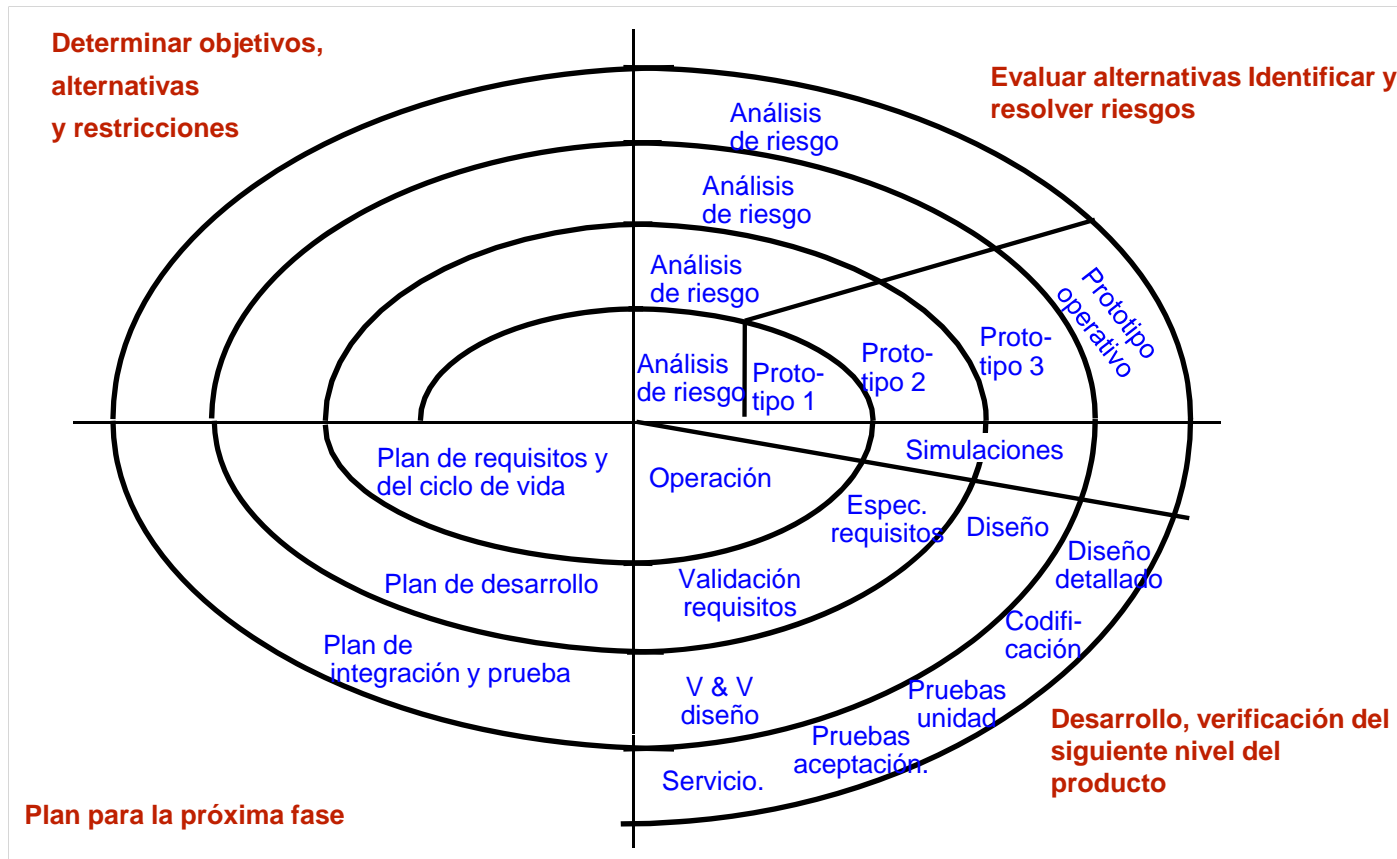
## Modelos en espiral (I)

### ■ Ciclo de vida en espiral

- Fue propuesto inicialmente por **B. Boehm** [Boehm, 1986, 1988]
- Es un modelo de proceso de software evolutivo, que proporciona el potencial para el desarrollo rápido de versiones incrementales del software
- Características
  - Puede considerarse como un metamodelo de proceso
    - Principalmente, reúne características del modelo clásico y de prototipos
  - Aparece el análisis de riesgo
  - Se divide en un número de actividades estructurales, también denominadas regiones de tareas. En el modelo original de **Boehm** aparecen cuatro regiones de tareas
    - Planificación, Análisis de riesgos, Ingeniería, Evaluación del cliente
  - El avance se realiza desde el centro de la espiral hacia el exterior

# Modelos de proceso

## Modelos en espiral (II)

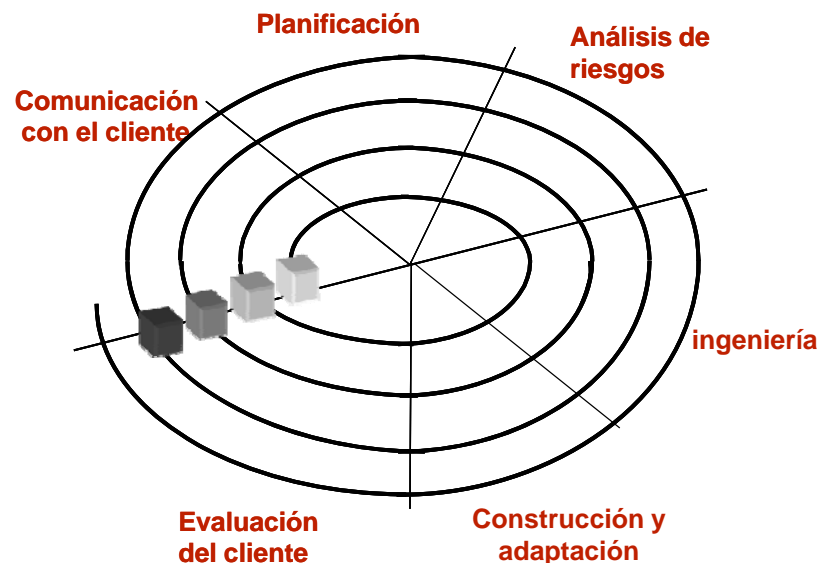


Ciclo de vida en espiral [Boehm, 1988]

# Modelos de proceso

## Modelos en espiral (III)

- **Modelo en espiral de Pressman** [Pressman, 2002]
  - Variante del modelo de Boehm con 6 regiones de tareas
  - Se define un eje con diferentes puntos de entrada para diferentes tipos de proyectos



### Puntos de entrada al proyecto

- Proyecto de mantenimiento de productos
- Proyecto de mejora de productos
- Proyecto de desarrollo de productos nuevos
- Proyecto de desarrollo de conceptos

Modelo en espiral de Pressman

## Modelos de proceso

### Modelos en espiral (IV)

- **Modelo *win-win*** [Boehm et al., 1998]
  - Extiende el modelo en espiral haciendo énfasis en las condiciones de éxito (ganancia) de todas las partes involucradas en el proyecto
  - Consta de cuatro **ciclos**:
    - **Ciclo 0. Grupos de aplicación**: Determinación de la viabilidad de un grupo
    - **Ciclo 1. Objetivos del ciclo de vida de la aplicación**: objetivos, prototipos, planes, especificaciones de cada aplicación y arquitectura viable
    - **Ciclo 2. Arquitectura del ciclo de vida de la aplicación**: establecimiento de una arquitectura detallada y verificación de su viabilidad
    - **Ciclo 3. Capacidad de operación inicial**: consecución de la capacidad para cada etapa crítica del proyecto





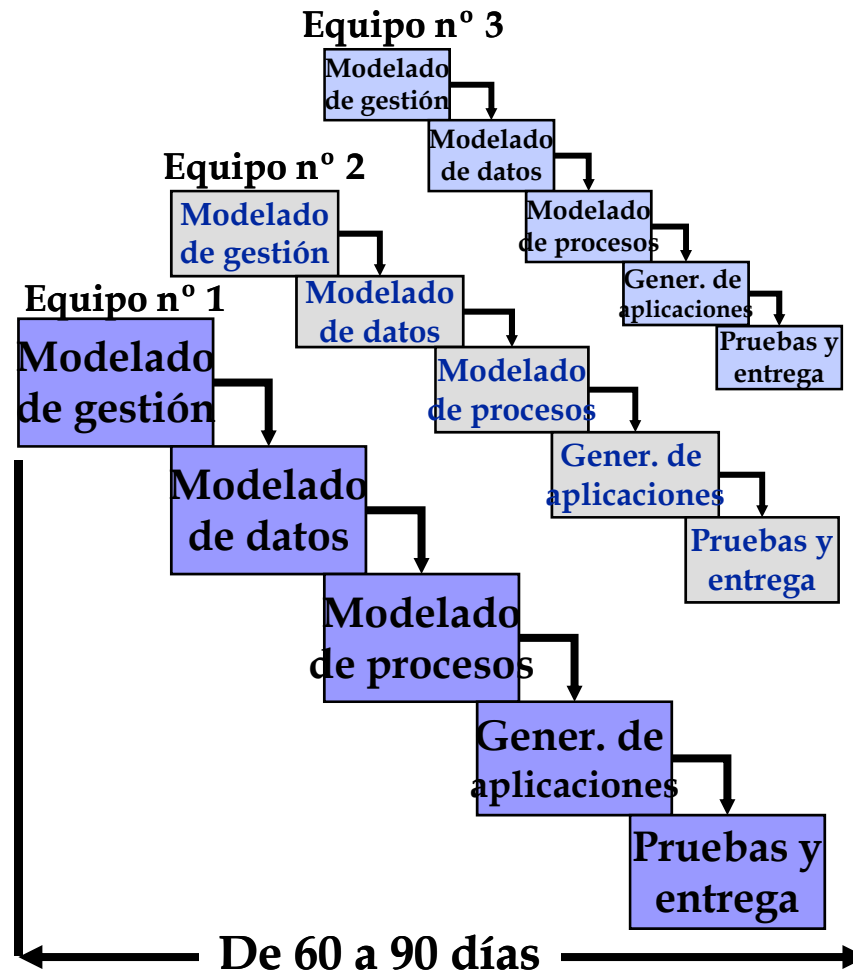
# Modelos de proceso

## Desarrollo rápido de aplicaciones (I)

- El modelo de desarrollo rápido de aplicaciones, DRA (RAD – *Rapid Application Development*) o modelo de la caja de tiempo surgió como respuesta al modelo formal y al ciclo en espiral
- Enfatiza un ciclo de desarrollo extremadamente corto
  - Modelo funcional en 60 ó 90 días
- No es un modelo bien definido
  - Secuencia de integraciones de un sistema evolutivo o de prototipos que se revisan con el cliente → descubrimiento de los requisitos
  - Cada integración se restringe a un período de tiempo bien definido (caja de tiempo)
- Características
  - Modelo secuencial: Separación en fases de cada *caja de tiempo*
  - Integraciones constantes
  - Centrado en el código más que en la documentación
  - Desarrollo basado en componentes
  - Uso efectivo de herramientas y *frameworks*
  - Participación activa del usuario

# Modelos de proceso

## Desarrollo rápido de aplicaciones (II)



Modelo DRA [Kerr y Hunter, 1994]

- Cuando se utiliza en S.I. Automatizados, comprende las fases [Kerr y Hunter, 1994]

- Modelado de gestión
- Modelado de datos
- Modelado del proceso
- Generación de aplicaciones
- Pruebas y entrega

## Modelos de proceso

### Desarrollo rápido de aplicaciones (III)

- Las limitaciones de tiempo demandan un ámbito de escalas
- Si una aplicación de gestión puede modularse de forma que pueda completarse cada una de las funciones principales en menos de tres meses, es un candidato del DRA. Cada una de estas funciones puede ser afrontadas por un equipo DRA diferente y ser integradas en una sola aplicación
- Inconvenientes [Butler, 1994]
  - Los proyectos grandes necesitan los recursos humanos suficientes para crear el número correcto de equipos
  - Se requiere de un compromiso de las partes involucradas

# Modelos de proceso

## Modelos orientados a la reutilización (I)

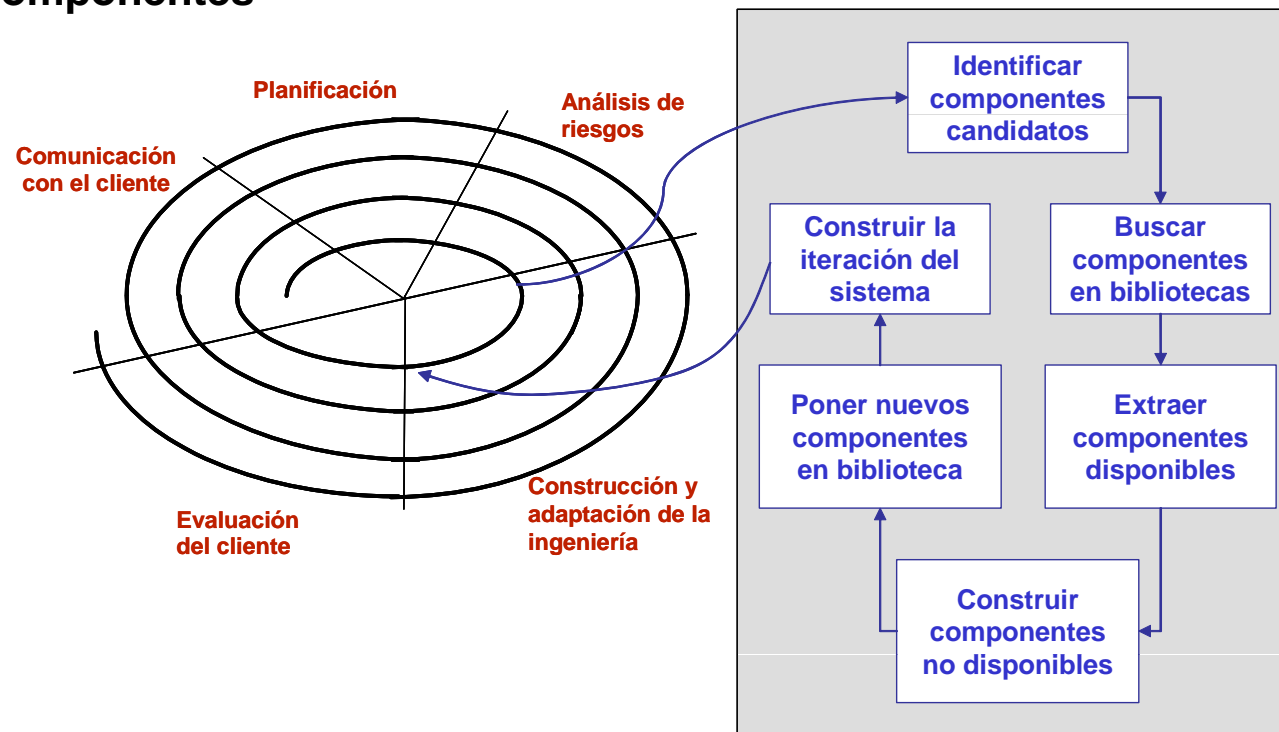
- Enfoque de desarrollo que trata de maximizar la reutilización de software existente [Sommerville, 2002]
  - Las **unidades software reutilizables** pueden ser de diferente tamaño:
    - **Sistemas de aplicaciones**: se reutiliza la totalidad del sistema
      - Sin ningún cambio (reutilización de productos COTS)
      - Desarrollo de familias de aplicaciones para plataformas diferentes o necesidades específicas
    - **Componentes**: la reutilización va desde subsistemas hasta objetos simples
    - **Funciones**: componentes de software que implementan una sola función
  - **Familias de aplicaciones** o líneas de productos: conjunto relacionado de aplicaciones que tiene una arquitectura común de dominio específico. Existen varios tipos de especialización:
    - **De la plataforma**: varias versiones de la aplicación se desarrollan para diferente plataforma
    - **De la configuración**: se crean diferentes versiones para manejar diversos dispositivos periféricos
    - **De la funcionalidad**: diferentes versiones para clientes con requisitos diferentes

# Modelos de proceso

## Modelos orientados a la reutilización (II)

### ■ Desarrollo basado en componentes (I)

- Configura aplicaciones a partir de componentes de software preparados [Pressman, 2002]
- Enfoque iterativo y evolutivo [Nierstrasz, 1999]
- Se enmarca en un contexto más amplio: **ingeniería del software basada en componentes**



# Modelos de proceso

## Modelos orientados a la reutilización (III)

- **Desarrollo basado en componentes (II)**
  - Un componente es una unidad ejecutable e independiente
  - Los componentes publican su interfaz y todas las interacciones son a través de ella
    - Una interfaz que se suministra define los servicios que ofrece el componente
    - Una interfaz que se solicita especifica qué servicios deben estar disponibles
  - Para el desarrollo con reutilización:
    - Debe ser posible encontrar los componentes reutilizables apropiados
    - Se debe confiar en que los componentes que se utilizan se comportan conforme a lo especificado y son fiables
    - Los componentes deben tener documentación asociada para ayudar a comprenderlos y adaptarlos a una nueva aplicación

# Modelos de proceso

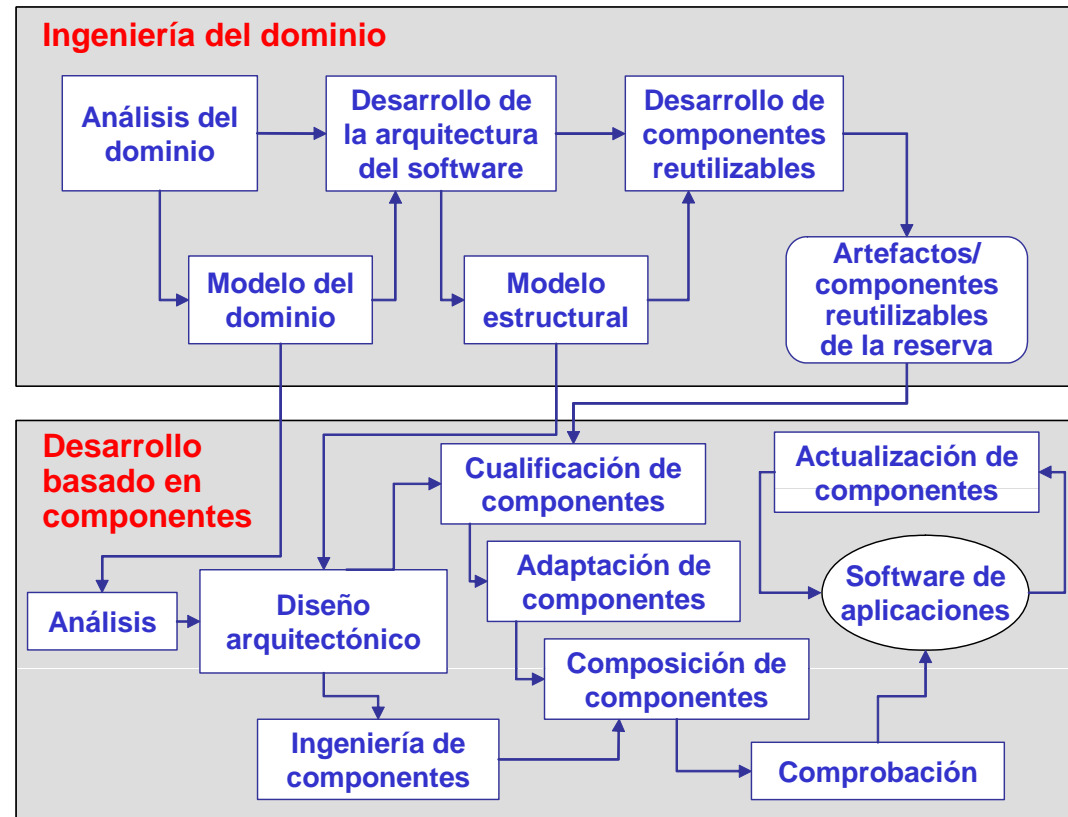
## Modelos orientados a la reutilización (IV)

### ■ Ingeniería del software basada en componentes

- Ingeniería del dominio
- Desarrollo basado en componentes

*El objetivo de la **ingeniería del dominio** es identificar, construir, catalogar y diseminar un conjunto de componentes de software que tienen aplicación en el software actual y futuro dentro de un dominio de aplicación particular*

[Presman, 2001]



Ingeniería del software basada en componentes



# Modelos de proceso

## Modelos orientados a la reutilización (V)

### ■ Actividades de la ingeniería del dominio

#### □ **Análisis del dominio:**

- Definir el dominio a investigar
- Categorizar los elementos extraídos del dominio
- Recoger una muestra representativa de las aplicaciones del dominio
- Analizar cada aplicación de la muestra
- Desarrollar un modelo de análisis para los objetos
- Definir un lenguaje del dominio: hace posible la especificación y construcción posterior de aplicaciones dentro del dominio

#### □ **Modelo del dominio:** resultado de las actividades anteriores

#### □ **Modelado estructural:** Enfoque de ingeniería basado en **tramas** que opera efectuando la suposición consistente de que todo dominio de aplicación posee tramas repetidas (de función, de datos y de comportamiento) que tienen un potencial de reutilización

- Todo dominio de aplicación se puede caracterizar por un modelo estructural
- Un modelo estructural es un **estilo arquitectónico reutilizable**
- **Punto de estructura:** estructura bien diferenciada dentro de un modelo estructural (genéricos: aplicaciones cliente, bases de datos, motores de cálculo, función de reproducción de informes, editor de aplicaciones)

# Modelos de proceso

## Modelos orientados a la reutilización (VI)

- **Actividades del desarrollo basado en componentes**
  - **Cualificación de componentes:** Asegura que un componente candidato llevará a cabo la función necesaria, encajará en el estilo arquitectónico del sistema y tendrá la calidad requerida
  - **Adaptación de componentes:** Elimina conflictos de integración
    - Enmascaramiento de caja blanca, gris o negra
  - **Composición de componentes:** Ensambla componentes cualificados, adaptados y diseñados para la arquitectura establecida
  - **Ingeniería de componentes:** Diseño de componentes para su reutilización
  - **Actualización de componentes:** El software actual se reemplaza a medida que se dispone de nuevas versiones de componentes

# Modelos de proceso

## Modelos para sistemas OO (I)

### ■ Características

- Eliminación de las fronteras entre fases
- Desarrollo basado en componentes reutilizables
- Desarrollo iterativo e incremental
  - Las tareas de cada fase se realizan de forma iterativa
  - Existe un ciclo de desarrollo que permite la evolución del sistema
  - Alto grado de iteración y solapamiento
  - El sistema se divide en un conjunto de particiones que se van desarrollando e integrando de forma incremental
- Se pueden combinar con modelos tradicionales

## Modelos de proceso

### Modelos para sistemas OO (II)

#### ■ Modelo de agrupamiento (I)

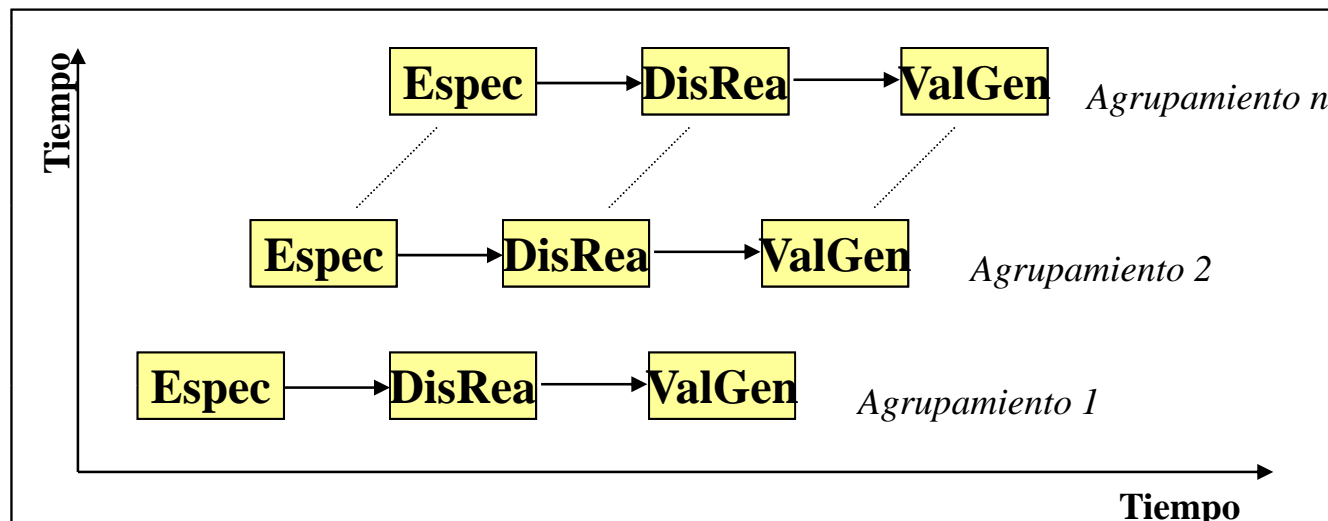
- Propuesto por **Bertrand Meyer** [Meyer, 1990]
- Concepto clave: **AGRUPAMIENTO (*cluster*)** [Meyer, 1999]
  - Unidad organizativa básica
  - Grupo de clases relacionadas o, recursivamente, *clusters* relacionados
  - Unidad natural para el desarrollo por parte de un único desarrollador
  - Evita el efecto todo-nada propio del modelo en cascada
- Tiene un componente secuencial y un componente concurrente
  - Existencia de diferentes subciclos de vida (uno para cada *cluster*) que pueden solaparse en el tiempo
  - Cada subciclo de vida que gobierna el desarrollo de un *cluster* está formado por
    - Especificación, Diseño, Implementación, Verificación/Validación y Generalización

## Modelos de proceso

### Modelos para sistemas OO (III)

#### ■ Modelo de agrupamiento (II)

- Enfoque ascendente
- La ocultación de la información posibilita la forma del modelo de *clusters* de ingeniería concurrente



Distribución temporal de las fases de cada agrupamiento

## Modelos de proceso

### Modelos para sistemas OO (IV)

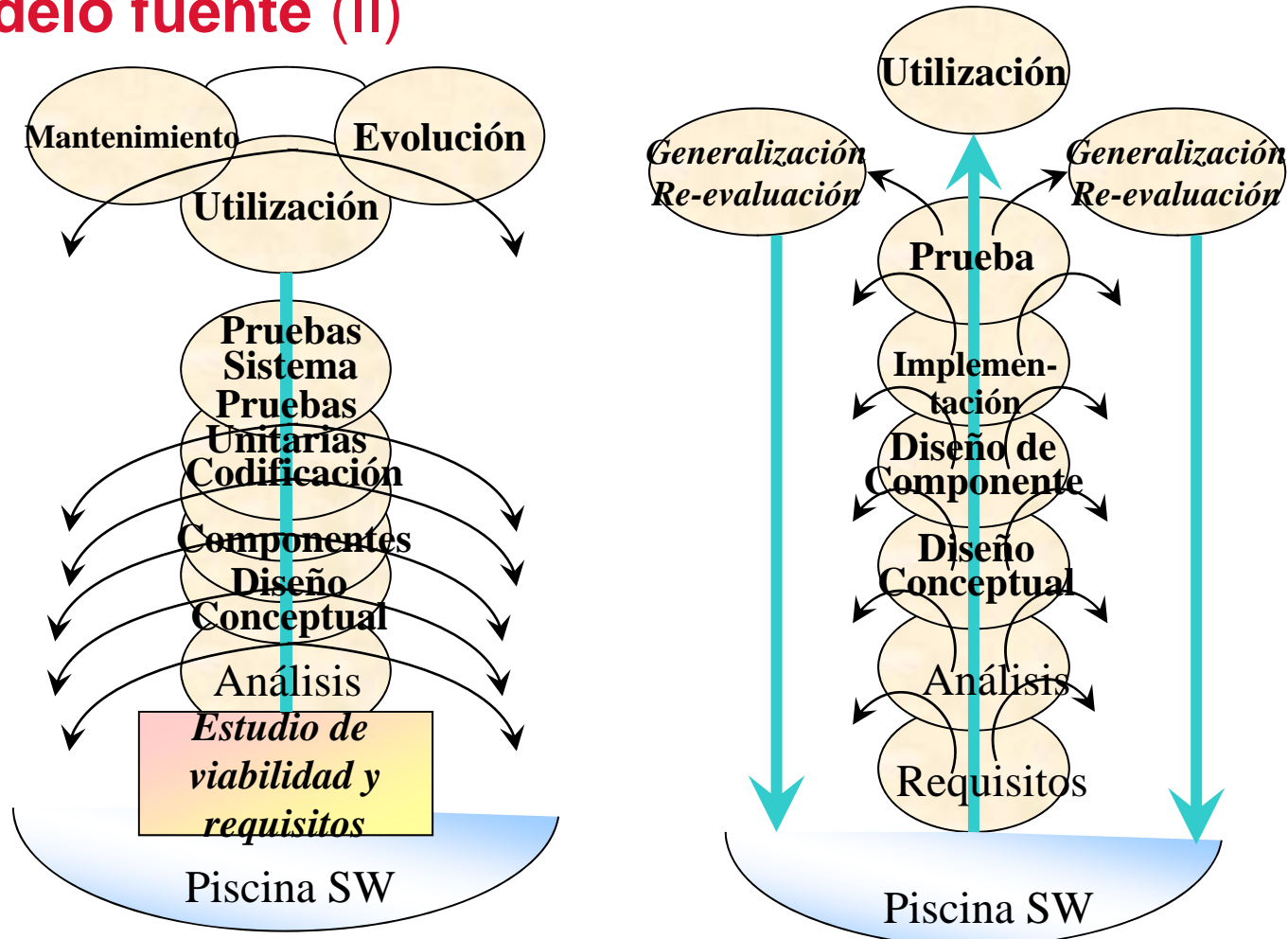
#### ■ Modelo fuente (I)

- Definido por Henderson-Sellers y Edwards en 1990 [Henderson-Sellers y Edwards, 1990]
- Representa gráficamente el alto grado de iteración y solapamiento que hace posible la tecnología de objetos
- Propone dos modelos de ciclo de vida
  - Para el sistema completo
  - Para cada clase o módulo: Cada clase puede estar en una fase diferente del ciclo de vida durante el desarrollo del sistema
- El modelo permite la integración del **análisis de dominio**: identificación, análisis y especificación de requisitos comunes de un dominio de aplicación específico

# Modelos de proceso

## Modelos para sistemas OO (V)

### ■ Modelo fuente (II)



Modelo fuente para el sistema y para un componente

# Modelos de proceso

## Modelos para sistemas OO (VI)

### ■ El proceso unificado (I)

- Definido por **Rational Software Corporation** [Jacobson et al., 2000]
  - Evolución del proceso Objectory de Rational
  - Utilización de UML [Booch et al., 1999] como lenguaje de modelado
  - Basado en componentes
- Características
  - **Conducido por casos de uso**
    - Los casos de uso se implementan para asegurar que toda la funcionalidad se realiza en el sistema y verificar y probar el mismo. Como los casos de uso contienen las descripciones de las funciones, afectan a todas las fases y vistas
  - **Centrado en la arquitectura**
    - La arquitectura se describe mediante diferentes vistas del sistema. Es importante establecer una arquitectura básica pronto, realizar prototipos, evaluarla y finalmente refinarla durante el curso del proyecto
  - **Iterativo e incremental**
    - Resulta práctico dividir los grandes proyectos en mini proyectos, cada uno de los cuales es una iteración que resulta en un incremento



## Modelos de proceso

### Modelos para sistemas OO (VII)

#### ■ El proceso unificado (II)

- El Proceso Unificado se repite a lo largo de una serie de ciclos
- Cada ciclo consta de cuatro **fases**:
  - **Inicio**: se define el alcance del proyecto y se desarrollan los casos de negocio
  - **Elaboración**: se planifica el proyecto, se especifican en detalle la mayoría de los casos de uso y se diseña la arquitectura del sistema
  - **Construcción**: se construye el producto
  - **Transición**: el producto se convierte en versión beta. Se corrigen problemas y se incorporan mejoras sugeridas en la revisión

## Modelos de proceso

### Modelos para sistemas OO (VIII)

#### ■ El proceso unificado (III)

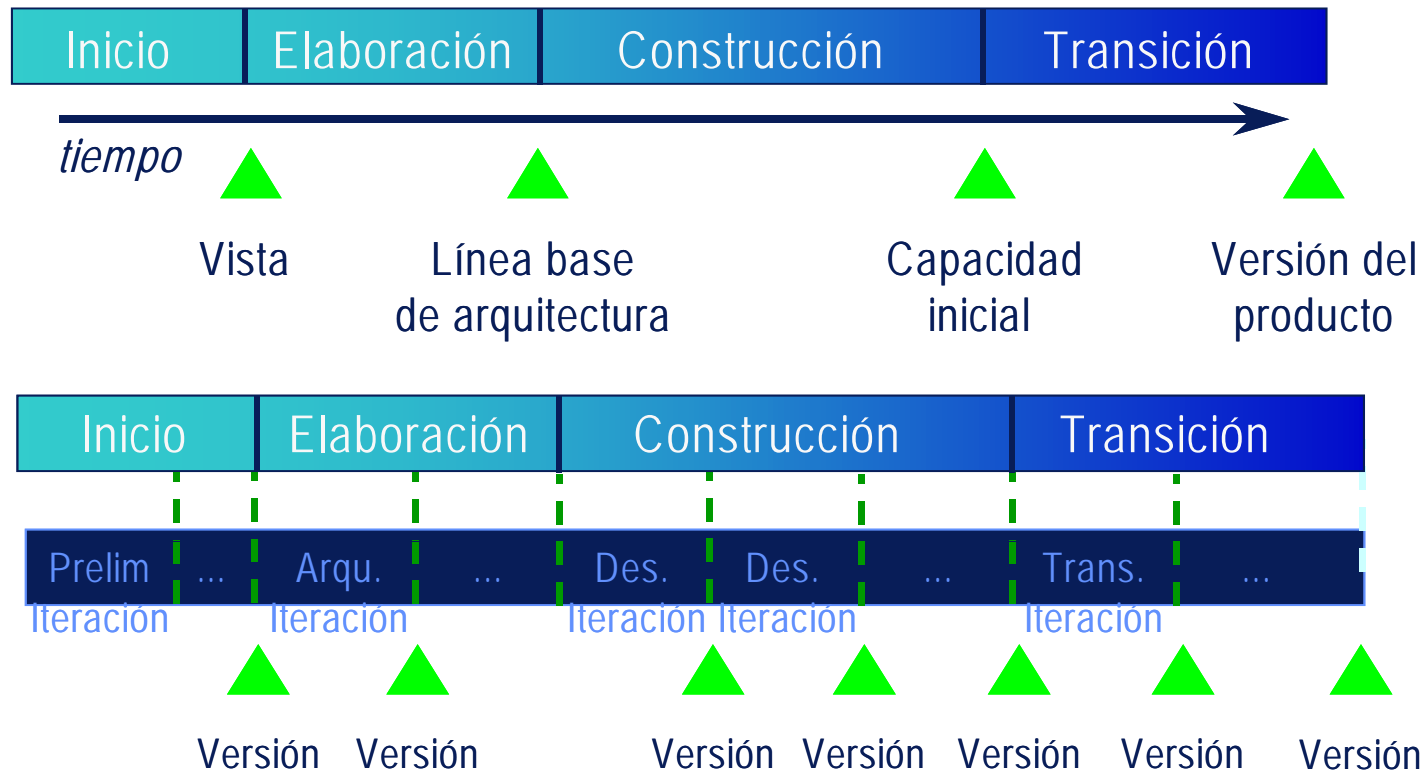
- Dentro de cada fase se puede, a su vez, descomponer el trabajo en **iteraciones** con sus incrementos resultantes
- Cada fase termina con un **hito**, cada uno de los cuales se caracteriza por la disponibilidad de un conjunto de componentes de software
  - Objetivos de los hitos:
    - Toma de decisiones para continuar con la siguiente fase
    - Controlar el progreso del proyecto
    - Proporcionar información para la estimación de tiempo y recursos de proyectos sucesivos

## Modelos de proceso

### Modelos para sistemas OO (IX)

#### ■ El proceso unificado (IV)

- Cada ciclo concluye con una versión del producto para los clientes

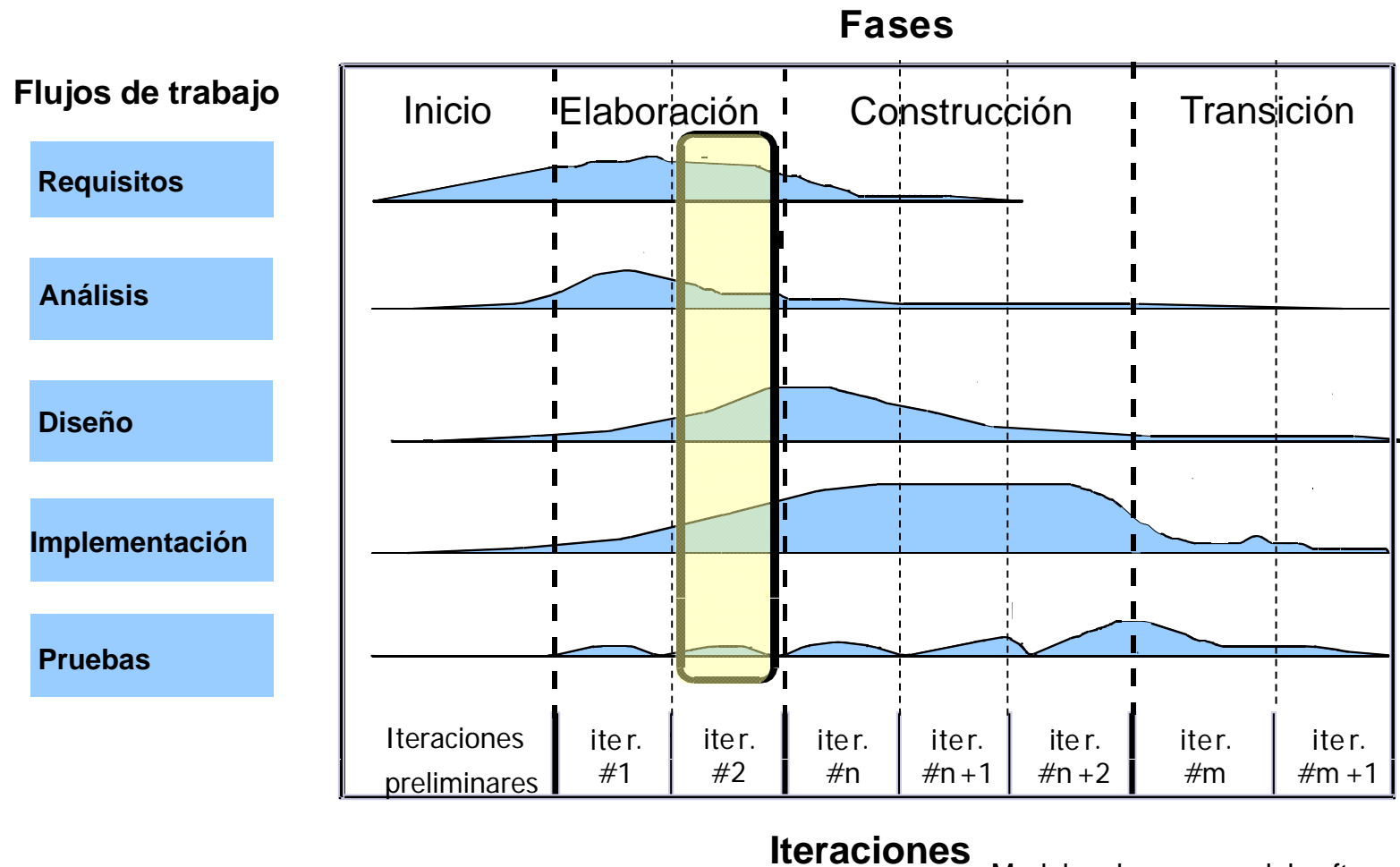


# Modelos de proceso

## Modelos para sistemas OO (X)

### ■ El proceso unificado (IV)

- Las iteraciones discurren a lo largo de los flujos de trabajo



# Modelos de proceso

## Procesos ágiles (I)

- Los procesos ágiles constituyen un nuevo enfoque en el desarrollo de software cuyas principales características son:
  - Menor énfasis en el análisis, diseño y documentación
  - Equipos pequeños
  - Desarrollo incremental
  - Programación (planificación temporal) en cajas de tiempo
  - Supervivencia en un entorno caótico
- Las aproximaciones ágiles emplean procesos técnicos y de gestión que continuamente se adaptan y se ajustan a (Turk et al., 2002)
  - Cambios derivados de las experiencias ganadas durante el desarrollo
  - Cambios en los requisitos
  - Cambios en el entorno de desarrollo
- Diversas aproximaciones
  - XP (*eXtreme Programming*) [Beck, 1999]
  - *Crystal* [Alistair Cockburn, 1999]
  - Proceso Software Adaptativo [Jim Highsmith, 2000]
  - *Scrum* [Schwaber, 1995]

# Modelos de proceso

## Procesos ágiles (II)

- **Programación extrema** [Beck, 1999]
  - Nuevo y controvertido enfoque de desarrollo de software basado en el modelo incremental
  - Está indicado para
    - Equipos de tamaño mediano o pequeño
    - Requisitos imprecisos y cambiantes
  - **Características:**
    - El juego de la planificación
    - Versiones pequeñas
    - Metáfora
    - Diseño sencillo
    - Hacer pruebas
    - *Refactoring*
    - Programación en parejas
    - Propiedad colectiva
    - Integración continua
    - Cliente in-situ
    - Estándares de codificación
  - Según Beck (2000) XP descansa sobre cuatro valores
    - Comunicación
    - Sencillez
    - Realimentación
    - Valentía

# Modelos de proceso

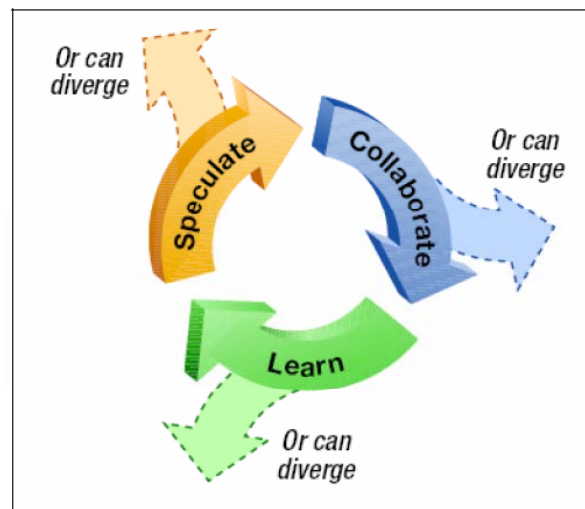
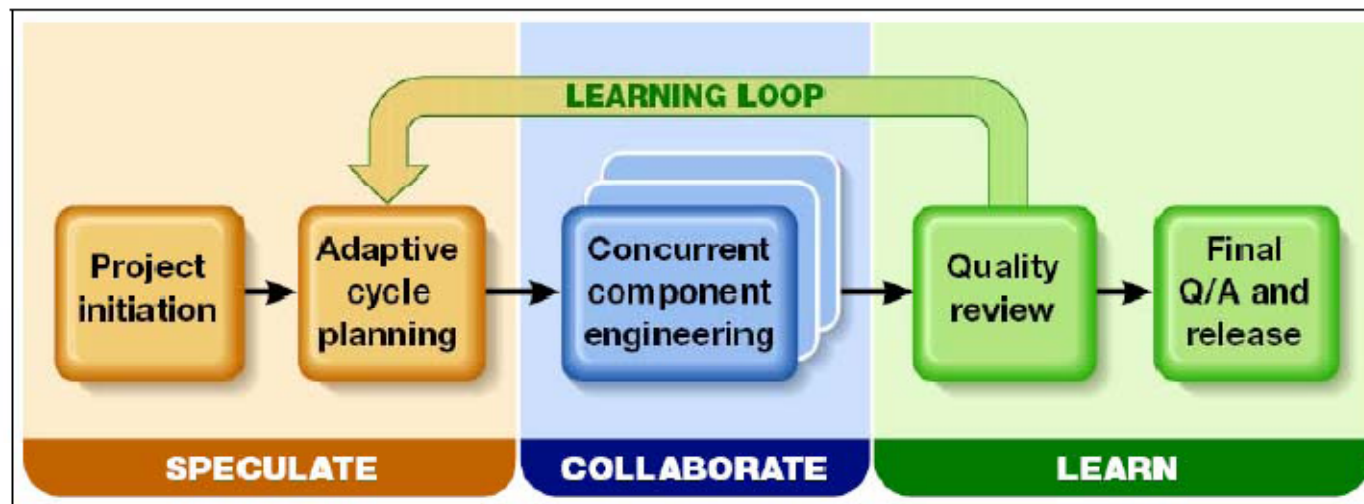
## Procesos ágiles (III)

- **Desarrollo de software adaptativo (I)** [Highsmith, 2000]
  - Modelo ágil y adaptativo basado en la colaboración y orientado al desarrollo de sistemas complejos
  - **Fases del ciclo de vida:**
    - **Especulación**
      - Inicio del proyecto
      - Planificación del ciclo adaptativo: enunciado, restricciones y requisitos básicos
        - Plan de lanzamiento: definición de un conjunto de ciclos (incrementos)
    - **Colaboración**
      - Construir la funcionalidad definida en la fase anterior
      - Uso de técnicas JAD (*Joint Application Development*) y trabajo colaborativo
    - **Aprendizaje**
      - Revisión de calidad al final de cada ciclo
      - Aprendizaje
        - Grupos enfocados
        - Revisiones técnicas formales
        - Post mortem

# Modelos de proceso

## Procesos ágiles (IV)

### ■ Desarrollo de software adaptativo (II)





## Modelos de proceso

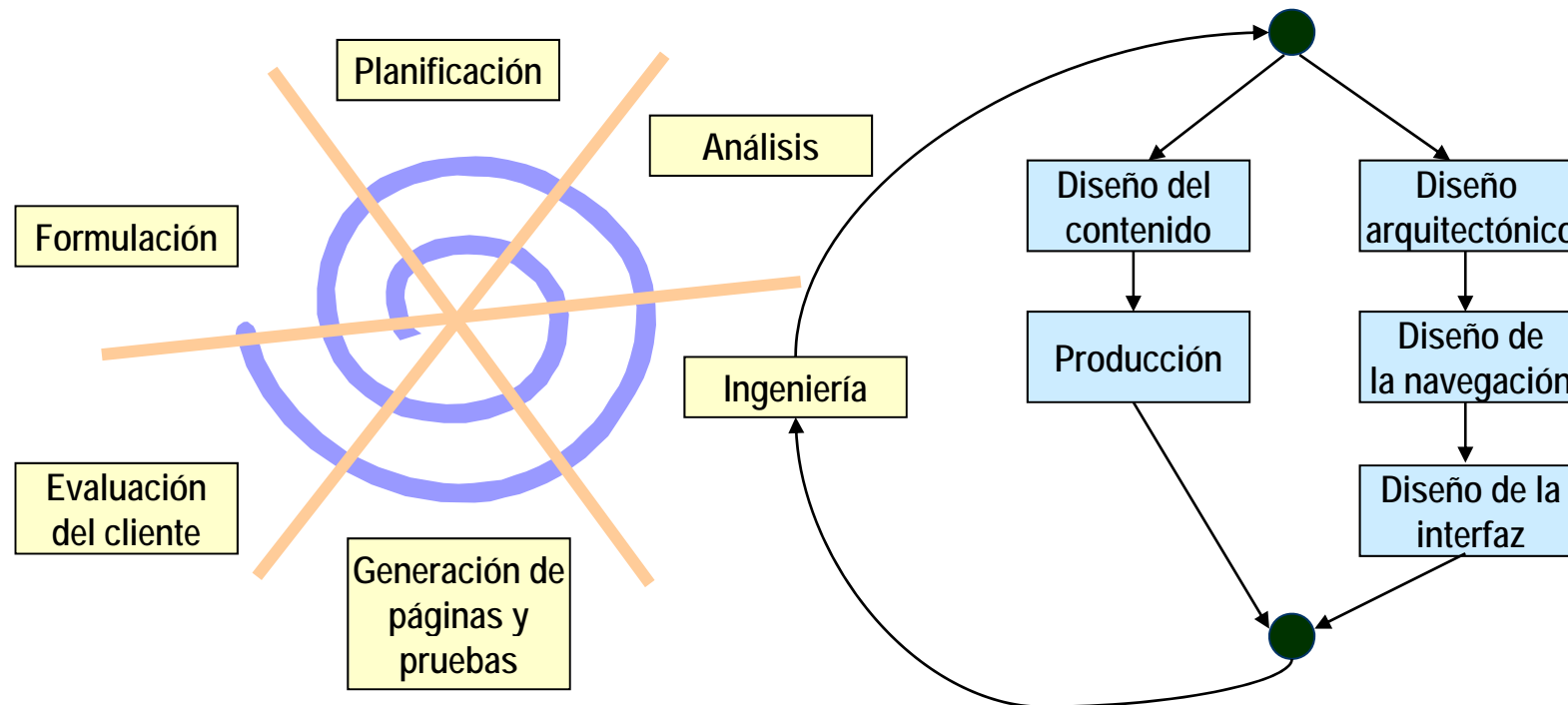
### Modelos de proceso de la Ingeniería Web (I)

- Las **características** de sistemas y aplicaciones basados en Web influyen enormemente en el proceso de Ingeniería Web (IWeb):
  - Intensivas de red
  - Controladas por contenido
  - Evolución continua
  - Inmediatez
  - Estética ...
- El ciclo de desarrollo de una aplicación Web consta de las siguientes fases de ingeniería:
  - Definición y análisis de los sistemas Web
  - Diseño de los sistemas Web
    - Diseño arquitectónico
    - Diseño de la navegación
    - Diseño de la interfaz
  - Pruebas de las aplicaciones Web

# Modelos de proceso

## Modelos de proceso de la Ingeniería Web (II)

### ■ Modelo de Pressman (I) [Pressman, 2002]



Modelo de proceso de IWEB [Pressman, 2002]

## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (III)

#### ■ **Modelo de Pressman (II)**

- **Formulación:** identificación de metas y objetivos
- **Planificación:** estimación de costes, evaluación de riesgos y planificación temporal del proyecto
- **Análisis:** establecimiento de requisitos
- **Ingeniería:** dos grupos de tareas paralelas,
  - Técnicas (diseño arquitectónico, de navegación y de interfaz)
  - No técnicas (diseño del contenido y producción)
- **Generación de páginas y pruebas**
  - El contenido se fusiona con los diseños arquitectónico, de navegación y de interfaz para elaborar páginas web ejecutables en HTML, JSP...
  - Integración con el software intermedio (*middleware*) de componentes
- **Evaluación con el cliente:** revisión de cada incremento y solicitud de cambios

## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (IV)

#### ■ **Modelo de Pressman (III)** [Pressman, 2006]

##### □ **Comunicación con el cliente:**

- Análisis de negocio
- Formulación

##### □ **Planificación:** definición de tareas y calendario para el desarrollo de un incremento

##### □ **Modelado:** las actividades de análisis y diseño convencionales se adaptan y se funden con las específicas de las aplicaciones Web

##### □ **Construcción:** construcción y prueba de un incremento

##### □ **Despliegue**

- Configuración de la aplicación para su ambiente operativo
- Entrega a los usuarios
- Evaluación

Las actividades se realizan siguiendo un flujo de proceso incremental

## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (V)

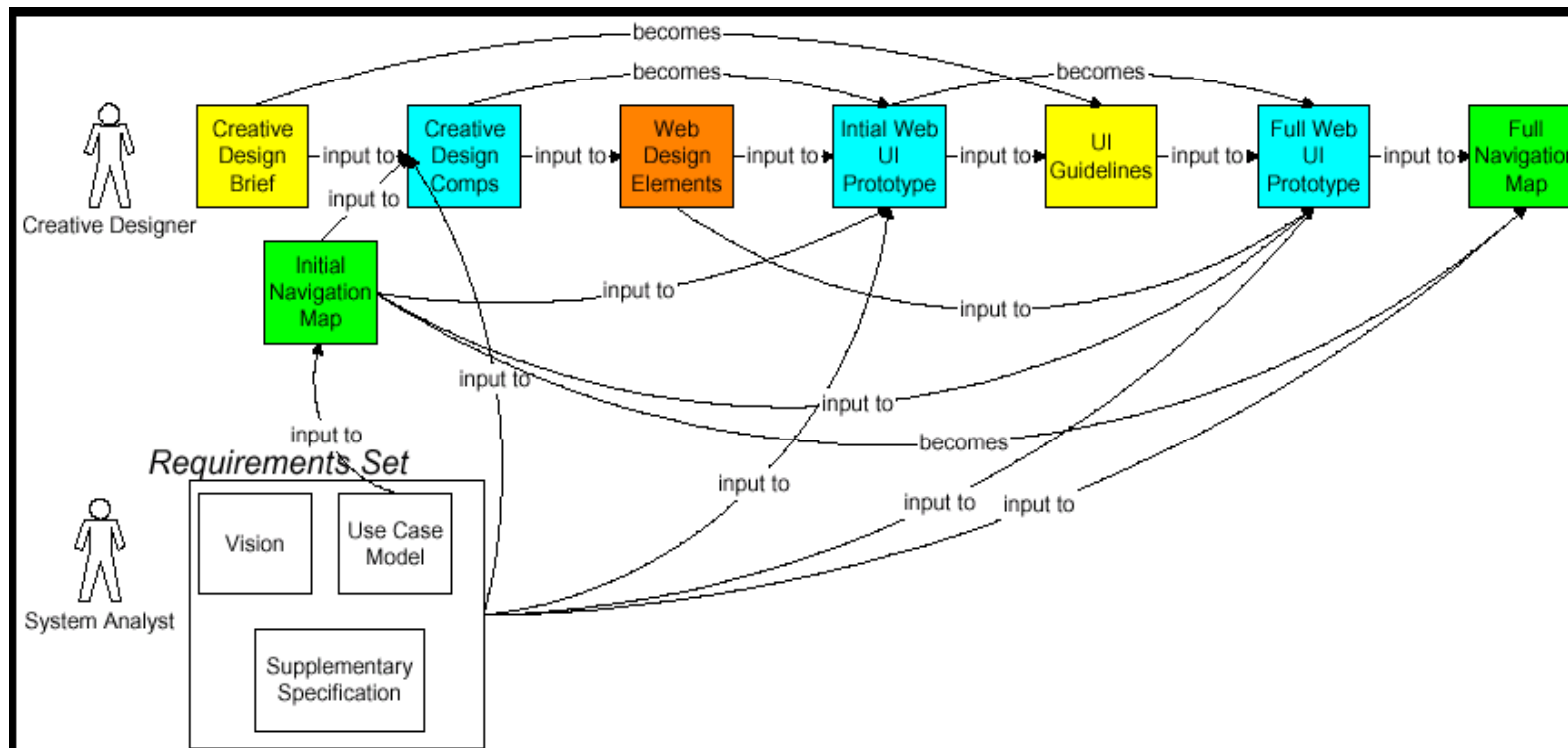
#### ■ El proceso unificado en la Ingeniería Web (I)

- La clave para utilizar el Proceso Unificado en el desarrollo de aplicaciones Web la dan los casos de uso (Ward y Kroll, 1999)
  - Integran el marco de ingeniería, que ofrece el Proceso Unificado, con el proceso de diseño creativo que caracteriza a las aplicaciones Web
  - Ofrecen una forma de expresar en términos comunes un entendimiento compartido del comportamiento esperado de la aplicación Web
  - Juegan el papel de lengua franca en los proyectos software, es decir, son el lenguaje hablado por todos los implicados en la definición y el desarrollo del sistema Web
- Integración del diseño creativo en el desarrollo
  - Requisitos
  - Diseño creativo
  - Mapa de navegación
  - Simulación del diseño creativo
  - Elementos de diseño Web
  - Prototipo inicial de IU Web
  - Guías IU
  - Prototipo completo de IU Web
  - Mapa de navegación completo

# Modelos de proceso

## Modelos de proceso de la Ingeniería Web (VI)

### ■ El proceso unificado en la Ingeniería Web (II)



# Modelos de proceso

## Modelos de proceso de la Ingeniería Web (VII)

### ■ El proceso unificado en la Ingeniería Web (III)

#### □ Requisitos

##### ■ Visión

- Acuerdo sobre los problemas que deben resolverse
- Definición de los límites del sistema
- Descripción de las características más importantes del sistema

##### ■ Modelo de casos de uso: documentación de los requisitos que permite a los usuarios articular sus necesidades (servicios del sistema)

- Los actores representan a los usuarios
- Los casos de uso representan los servicios

##### ■ Especificación suplementaria: contiene los requisitos no funcionales. Conviene desarrollar un glosario con la terminología común del proyecto

#### □ **Diseño creativo:** guías iniciales de la interfaz de usuario

- El “humor del sitio”
- Cómo accederán los usuarios al sitio, qué navegadores usarán
- Si el sitio tendrá marcos
- Limitaciones de color del sitio
- Aspectos relativos a gráficos, animaciones...

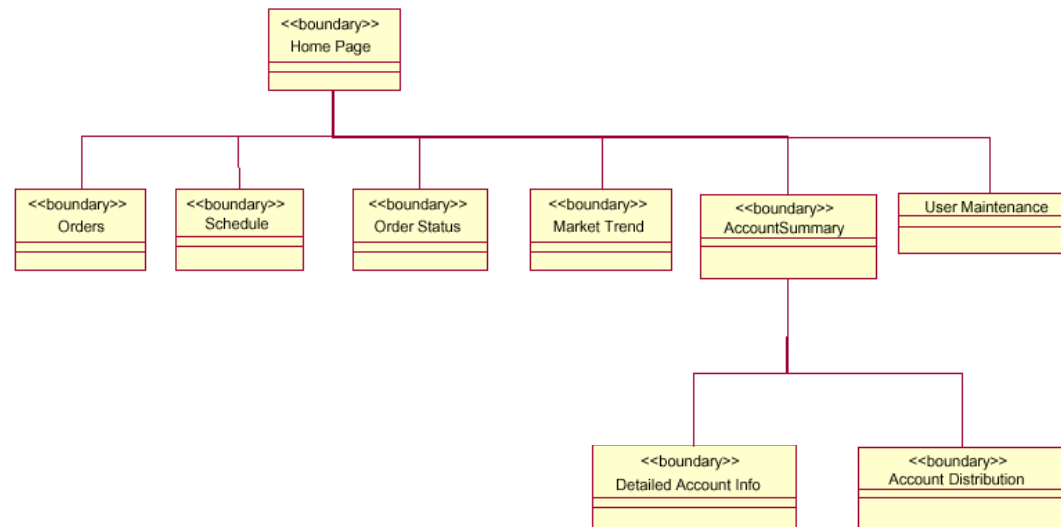
# Modelos de proceso

## Modelos de proceso de la Ingeniería Web (VIII)

### ■ El proceso unificado en la Ingeniería Web (IV)

- **Mapa de navegación:** representación jerárquica de la navegación de los usuarios en el sitio (*site map*)
  - El número de niveles representa el número de *clicks* necesarios para llegar a una página
  - Toma como referencia el modelo de casos de uso
  - Se identifican “páginas lógicas” candidatas para la interfaz de usuario. Se representan en el análisis con el constructor UML ***boundary class***

Full Site Navigation Map





## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (IX)

#### ■ El proceso unificado en la Ingeniería Web (V)

- **Simulación del diseño creativo:** actividad de *prototipado de la interfaz de usuario* de RUP
  - Se toma un caso de uso principal y se generan diseños alternativos
  - Se construyen prototipos de los diseños seleccionados por los *stakeholders*
- **Elementos de diseño Web:** imágenes gráficas discretas que se ensamblan para construir la página web
  - Reutilización de componentes gráficos estándar
  - Identificación de componentes a partir de casos de uso
  - Se crean en paralelo con el diseño inicial del prototipo de IU
- **Prototipo inicial de IU Web:** actividad RUP de prototipado de la interfaz de usuario
  - Soporta sólo una porción del sistema
  - Se utilizan los elementos de diseño identificados en la etapa anterior

## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (X)

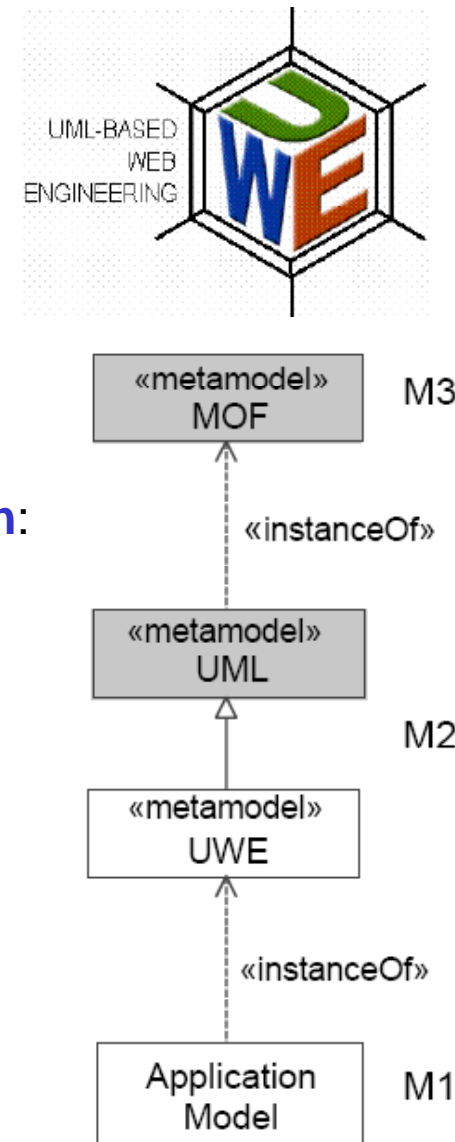
#### ■ **El proceso unificado en la Ingeniería Web (VI)**

- **Guías IU:** actividad RUP de *Guías de desarrollo de IU*
- **Prototipo completo de IU Web:** extiende el prototipo inicial para cubrir todos los casos de uso
  - Muestra la navegación completa entre pantallas y todos los elementos visuales del sitio
  - Las páginas del prototipo se refinarán de forma iterativa en el desarrollo
- **Mapa de navegación completo**
  - Se basa en el mapa inicial y en la definición completa de los casos de uso
  - Incluye todas las páginas/pantallas del prototipo IU Web

## Modelos de proceso

### Modelos de proceso de la Ingeniería Web (XI)

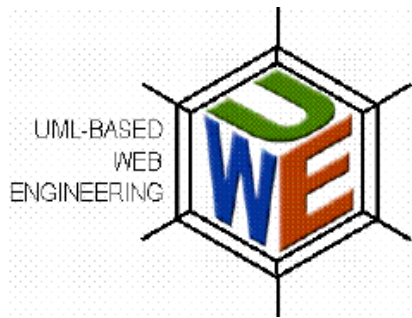
- **UWE (UML-based Web Engineering) (I)**  
[Koch, 2001; Hennicker y Koch, 2000]
  - **Desarrollo iterativo e incremental:** basado en el Proceso unificado
  - **Uso de UML:** perfil UML propio
  - **Centrado en la sistematización y automatización:**
    - Proceso sistemático de diseño
    - Generación semiautomática de aplicaciones web a través de un *framework* de publicación XML (UWEXML)
  - **UWE comprende:**
    - Una **notación**
    - Un **método**
    - Un **metamodelo**
    - Un **proceso de desarrollo**
    - Una **herramienta CASE**



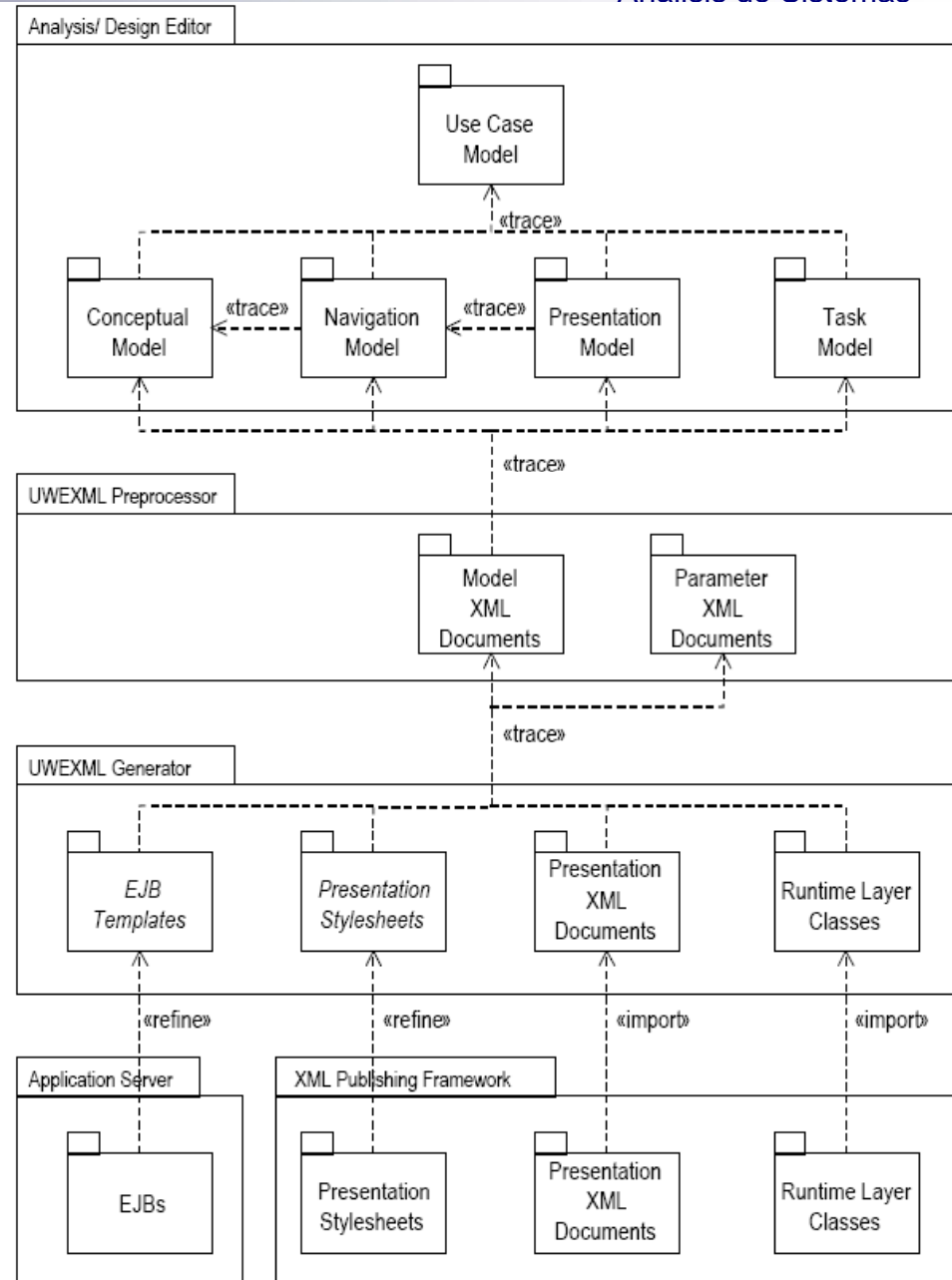
# Modelos de proceso

## Modelos de proceso de la Ingeniería Web (XII)

- **UWE (UML-based Web Engineering) (II)**



Vista general del proceso UWE



## Bibliografía

- Ambler, S. W. "***In search of a generic SDLC for object systems***". Object Magazine, 4(6): 76-78, 1994.
- Beck, K. "***Embracing Change with Extrem Programming***", IEEE Computer 32, pp. 70-77, 1999.
- Beck, K. "***Extreme Programming Explained. Embrace Change***". Addison-Wesley, 2000.
- Birrel, N. D., Ould, M.A. "***A practical Handbook for Software Development***". Cambridge University Press, 1985.
- Boehm, B. W. "***A Spiral Model of Software Development and Enhancement***". ACM Software Engineering Notes, 11(4):22-42. 1986.
- Boehm, B. W. "***A Spiral Model of Software Development and Enhancement***". Computer , 21(5): 61-72 , 1988.
- Boehm, B., Egyed, A., Port, D., Shah, A., Kwan, J., Madachy, R. "***A Stakeholder Win-Win Approach to Software Engineering Education***". Annals of Software Engineering , 6, 295-321, 1998.
- Booch, G., Rumbaugh, J., Jacobson, I. "***El Lenguaje Unificado de Modelado***". Addison Wesley, 1999.
- Butler, J. "***Rapid Application Development in Action***". Managing System Development, Applied Computer Research, 14(5):6-8. May, 1994.
- Cockburn, A. "***Software Development as a Cooperative Game***", Humans and Tecnology inc., 1999.  
<http://alistair.cockburn.us/crystal/articles/sdacg/softwaredevelopmentasacooperativegame.html>
- Graham, I. "***Métodos orientados a objetos***", Adison-Wesley, 1996.
- Hennicker, R. y Koch, N. "***A UML-based Methodology for Hypermedia Design***". En *Proceedings of the Unified Modeling Language Conference (UML'2000)*. A. Evans y S. Kent (Eds.). Lecture Notes in Computer Science LNCS Vol. 1939. Páginas 410-424. Springer-Verlag, 2000.
- Henderson-Sellers, B., Edwards, J. M. "***The fountain Model for object-oriented systems development***", Object Magazine, julio/agosto, pp 71-79, 1993.
- Henderson-Sellers, B., Edwards, J. M. "***The object-oriented systems life cycle***", Communications of the ACM, 33(9): 143-159, 1990.
- Highsmith, J., "***Adaptive Software Development: A Colaborative Approach to Managing Complex Systems***", Dorset House, 2000.
- IEEE. "***IEEE Software Engineering Standards Collection 1999 Edition. Volume 2: Process Standards***". IEEE Computer Society Press, 1999.
- ISO/IEC. "***Information Technology – Software Life Cycle Processes***". Technical ISO/IEC 12207:1995(E), 1995.
- ISO/IEC. "***Systems and Software engineering - Software life cycle processes***". ISO/IEC 12207:2008, 2008.

## Bibliografía

- Jacobson, I., Booch, G., Rumbaugh, J. **“El Proceso Unificado de Desarrollo”**, Addison Wesley, 2000.
- Kerr, J., Hunter, R. **“Inside RAD”**, McGraw-Hill, 1994.
- Koch, N. **“Software Engineering for Adaptive Hypermedia Applications. Reference Model, Modeling Techniques and Development Process”**. PhD. Thesis, Ludwig-Maximilians-Universität München, 2001.
- Kruchten, P. **“Un processus de développement de logiciel itératif et centré sur l’architecture”**, Proceedings of the 4th International Conference on Software Engineering. Toulouse, Paris, 1991.
- Martin, J. **“Rapid Application Development”**, Prentice Hall, 1991.
- Meyer, B. **“La Nueva Cultura del Desarrollo de Software”**, Systems, pp. 12-13. Septiembre, 1990.
- Meyer, B. **“Construcción de software orientado a objetos”**, Prentice Hall, 1999.
- Mills, H. D., Dyer, M., Linger, R. **“Cleanroom Software Engineering”**, IEEE Software, 4(5): 19-25. September 1987.
- Muller, P. A. **“Modelado de objetos con UML”**, Eyrolles-Ediciones Gestión 2000, 1997.
- Nierstrasz, O., Gibbs, S.J., Tsichritzis, D. **“Component-Oriented Software Development”**. CACM, 35(9): 160-165, 1992.
- Piattini, M.G. et al. **“Análisis y diseño detallado de aplicaciones Informáticas de Gestión”**, Rama, 1996.
- Pressman, R. S. **“Ingeniería del Software, un enfoque práctico”**, 5ª Edición. Mc Graw Hill, 2002.
- Pressman, R. S. **“Ingeniería del Software, un enfoque práctico”**, 6ª Edición. Mc Graw Hill, 2006.
- Royce, W. W. **“Managing the Development of Large Software Systems: Concepts and Techniques”**, In Proceedings WESCON. August, 1970.
- Rumbaugh, J. **“Over the waterfall and into the whirlpool”**, JOOP, mayo, pp 23-26, 1992.
- Sommerville, I. **“Software Engineering”**, 6th ed., Addison Wesley, 2001.
- Schwaber, K. **“SCRUM Development Process”**. OOPSLA’95 Workshop on Business Object Design and Implementation, <http://www.tiac.net/Users/jsuthfoopsla/oo95summary.html>, 10 Dec 95.
- Turk, D., France, R. y Rumpe, B. (2002) **Limitations of Agile Software Processes**. En Proceedings of 4th International Conference on eXtreme Programming and Agile Processes in Software Engineering, XP2002. (Alghero, Sardinia, Italy, April 2002), pp. 43-46, 2002.
- Ward, S. , Kroll, P., **“Building Web Solutions with the Rational Unified Process: Unifying the Creative Design Process and the Software Engineering Process”**, Rational Software & Context Integration white paper, 1999.