

# Ingeniería de Software

---

Silvia Guardati

# Rational Unified Process (RUP)

---

- Introducción
- Mejores prácticas
- Características
- Estructura: estática y dinámica

# Introducción

---

**Objetivo**: asegurar la producción con alta calidad (software satisface los requisitos del cliente en tiempo y costo estimados).

- Basado en Unified Process (UP). Es de Rational (IBM).
- Asigna actividades, responsabilidades y productos en una organización dedicada al desarrollo de software.
- Compuesto por diferentes procesos: **marco de trabajo adaptable** y **extensible** que puede ajustarse a las necesidades de cada organización.

# Mejores prácticas usadas

---

1. Planeación de actividades/productos, etc.
2. Desarrollo de software iterativo
3. Administración de los requisitos
4. Arquitectura basada en componentes
5. Modelo visual de software (UML)
6. Verificación de la calidad (pruebas en todas las etapas)
7. Administración de la configuración
8. Administración del riesgo

# Principales Características de RUP

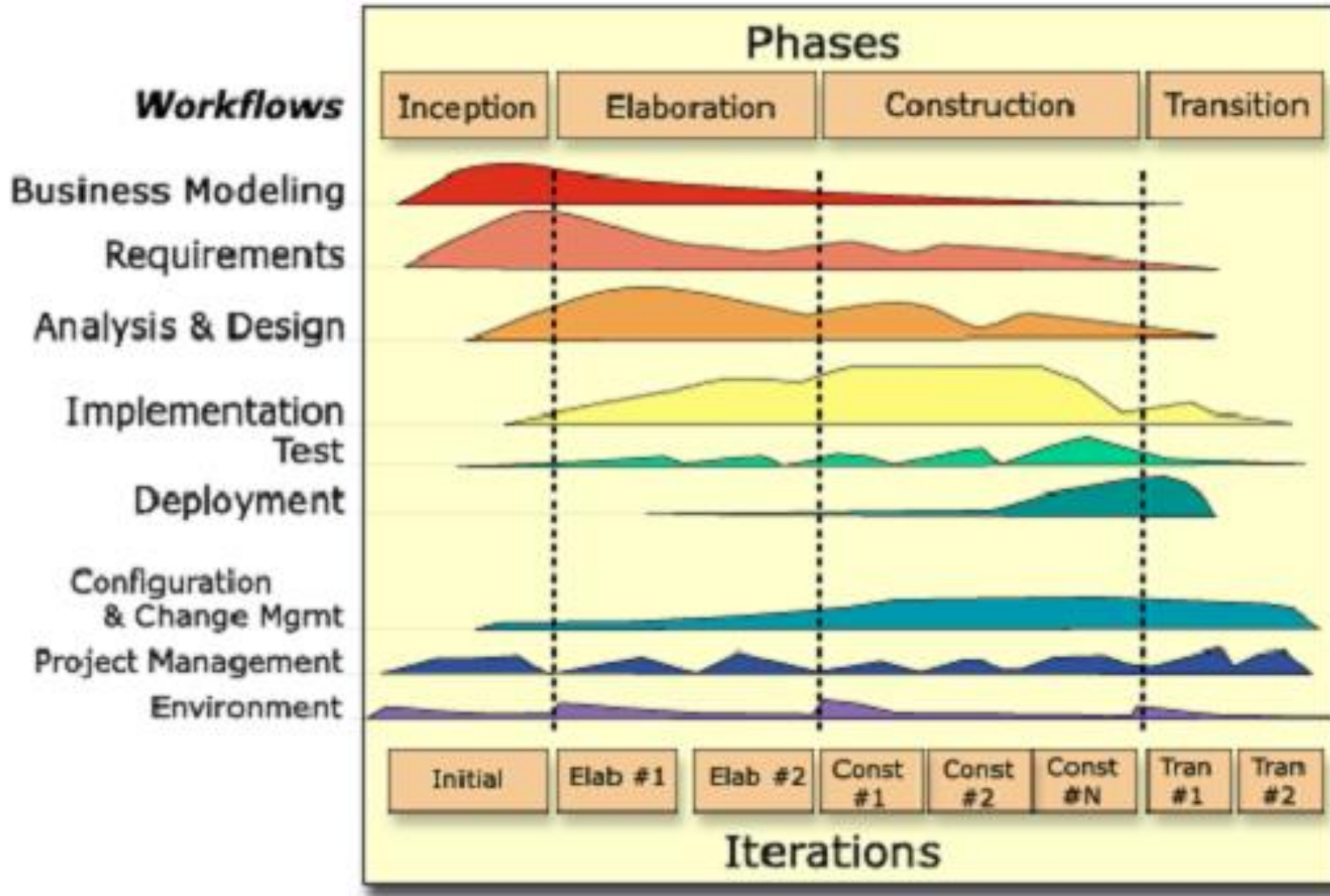
---

1. Dirigido por Casos de Uso (CU)
2. Centrado en la Arquitectura
3. Iterativo e Incremental

# Iteraciones



# Estructura de RUP



# Estructura de RUP

---

- **Estructura Dinámica**: tiempo. Compuesta por 4 fases: *inicial, elaboración, construcción y transición*.
- **Estructura Estática**: disciplinas. Intervienen roles, actividades, artefactos y flujos de trabajo.

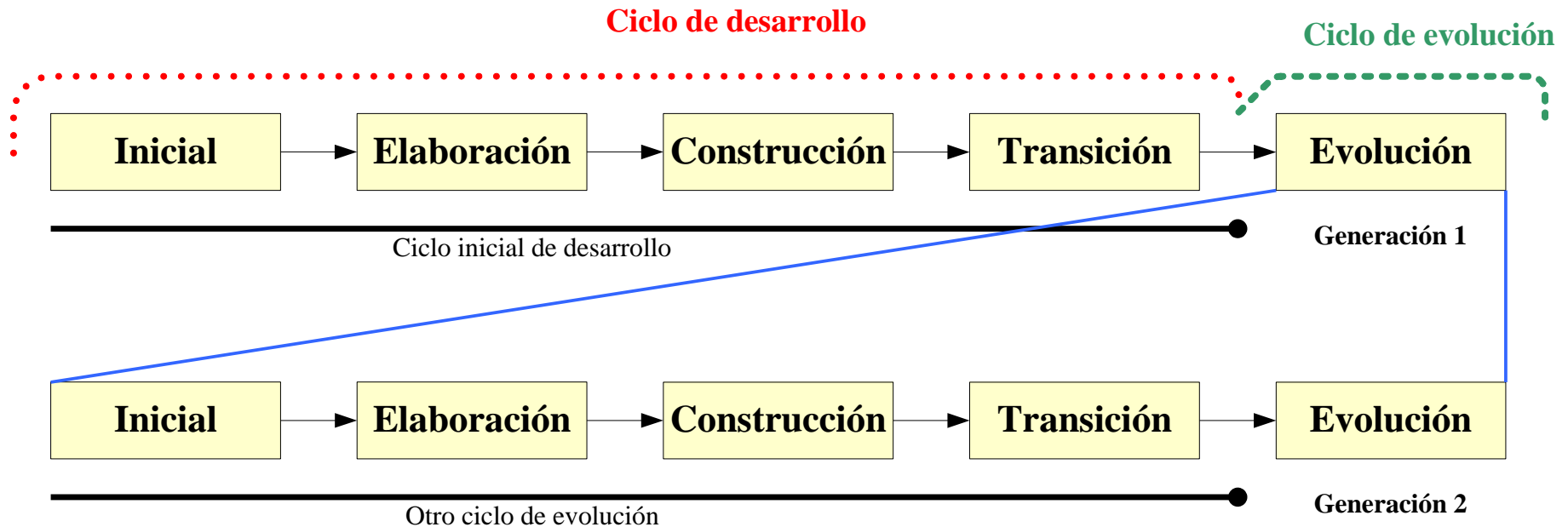


# Estructura Dinámica

---

- RUP repite una serie de ciclos.
- Resultado de cada ciclo: (parte del) producto.
- Ciclo  $\rightarrow$  4 fases  $\rightarrow$   $n$  iteraciones ( $n$  variable).
  - Cada fase tiene objetivos. Para alcanzarlos se desarrollan los pasos indicados en la estructura estática.
  - Cada fase itera sobre la estructura estática, tantas veces como se haya planeado.

# Ciclos - Iteraciones



# Iteraciones

---

Cada fase se divide en iteraciones.

**Iteración:** es el desarrollo completo del ciclo, cuyo resultado(\*) es la liberación (interna o externa) de un producto (o parte de = incremento).

(\*)El resultado de una iteración es parte del producto final.

# Beneficios del enfoque iterativo

- Posibilidad de mitigar los riesgos en etapas tempranas.
- Los cambios son más fáciles de manejar.
- El equipo puede ir aumentando sus conocimientos y capacidades a lo largo del proyecto.
- Mejor calidad en general.

# Fases de la estructura dinámica

- Inicial
- Elaboración
- Construcción
- Transición

(cada una de las fases puede ajustarse a la organización y/o proyecto)

# Inicial - Objetivo

---

Alcanzar un acuerdo entre los interesados: entender el problema –necesidades del cliente-

Los principales objetivos de la *fase inicial* son:

1. Establecer el ámbito del sistema y sus límites.
2. Definir los CU, así como los escenarios más importantes del sistema.
3. Encontrar al menos una posible arquitectura que cubra los principales CU .
4. Estimar el costo global y la planificación de todo el proyecto (y estimaciones detalladas para la fase de elaboración).
5. Estimar los riesgos.

# Inicial - Resultados

---

1. Un documento de visión.
2. Modelo inicial de Casos de Uso .
3. Glosario inicial.
4. Lista de riesgos y plan de contingencia.
5. Plan del proyecto, mostrando fases e iteraciones.
6. Prototipos exploratorios para probar conceptos o la arquitectura considerada.
7. Modelo de negocio, si es necesario.

# Inicial –Evaluación

---

1. Todos los interesados coinciden en la definición del ámbito del sistema y las estimaciones hechas.
2. Casos de Uso principales reflejan los requisitos.
3. Las estimaciones de tiempo, costo y riesgo son fiables.
4. El prototipo de la arquitectura desarrollado demuestra que el sistema puede hacerse.
5. El costo hasta el momento se ajusta a lo planeado.



# Elaboración - Objetivos

---

1. Definir y validar la arquitectura.
2. Completar la visión.
3. Planear la fase de construcción.
4. Demostrar que la arquitectura propuesta se ajusta a las necesidades con un costo y en un tiempo razonables.

# Elaboración - Resultados

---

1. Modelo de CU completo (mínimo hasta el 80%): todos los CU identificados y la mayoría desarrollados.
2. Requisitos no funcionales.
3. Descripción de la arquitectura software.
4. Prototipo ejecutable de la arquitectura.
5. Lista de riesgos y caso de negocio revisados.
6. Plan de desarrollo para el proyecto.

# Elaboración - Evaluación

---

1. Visión del producto es estable.
2. Arquitectura es estable.
3. Con el prototipo se demostró que los principales riesgos (técnicos) están controlados.
4. Plan para la fase de construcción es detallado y preciso.
5. Las estimaciones son confiables.
6. Acuerdo entre interesados acerca del producto: factible con los planes y en el contexto actuales.
7. Costos : se ajustan a lo planeado.

# Evaluación

---

**Inicial y Elaboración:** si no se cumplen los criterios de evaluación, se debe evaluar la **posibilidad de abandonar el proyecto o redefinirlo convenientemente.**

# Construcción - Objetivos

---

1. Minimizar costos.
2. Calidad adecuada.
3. Versiones funcionales.
4. Producto completo preparado para la transición a su comunidad de usuarios.
5. Decidir si el software, los sitios y los usuarios están listos para la aplicación.
6. Paralelismo en el trabajo de los equipos de desarrollo.

# Construcción - Resultados

---

1. Modelos Completos.
2. Arquitectura.
3. Riesgos mitigados.
4. Plan del Proyecto para la fase de Transición.
5. Manual de Usuario.
6. Producto Operacional.
7. Caso del Negocio Actualizado.

# Construcción - Evaluación

---

1. Producto estable y maduro.
2. Los usuarios expertos están preparados para la transición a la comunidad de usuarios.
3. Costos hasta el momento se ajustan a los planeados.

# Transición (1)

---

- Garantizar que el software quede disponible para los usuarios.
- Puede implicar varias iteraciones.
- Incluye las pruebas del producto (liberación + ajustes menores).
- Retroalimentación de los usuarios: ajuste de configuración, instalación y utilización.
- Capacitar a los usuarios y administradores.



# Transición (2)

---

- Preparar el sistema para ser distribuido y vendido (si aplica).
- Obtener el visto bueno del cliente.
- Definir la evolución del proyecto.
- Funcionamiento en paralelo con los sistemas que están siendo sustituidos (si aplica).
- Conversión de las bases de datos operacionales (si aplica).

# Transición - Objetivos

---

1. El producto final cumple con los requisitos identificados.
2. El usuario opera el sistema.

# Transición - Resultados

---

1. Producto operacional.
2. Documentos legales.
3. Modelo del negocio completo (si corresponde).
4. Documentación del producto completa y consistente con el producto generado.

# Transición - Evaluación

---

1. Cliente/usuario satisfecho.
2. Costo real según lo planeado.

# Fases: hitos

---

- Cada fase concluye con un hito.
- En ese momento se evalúa la fase antes de pasar a la siguiente.
- El hito principal de cada fase se compone de hitos menores.
- Hitos menores → criterios aplicables a cada iteración.

# Hitos x Fases

---

## INICIAL

- Objetivos
- Visión

## ELABORACIÓN

- Arquitectura

## CONSTRUCCIÓN

- Producto operacional

## TRANSICIÓN

- Liberación del producto

# Estructura Estática

---

- **Estructura estática:** ¿quién hace qué?, ¿qué hace?, ¿cómo lo hace? y ¿cuándo lo hace?
- Secuencia de actividades que se desarrollan en cada fase de la estructura dinámica.
- Estructura estática → ciclos de las fases de la estructura dinámica.

# Elementos de la estructura estática

---

¿Quién? → roles

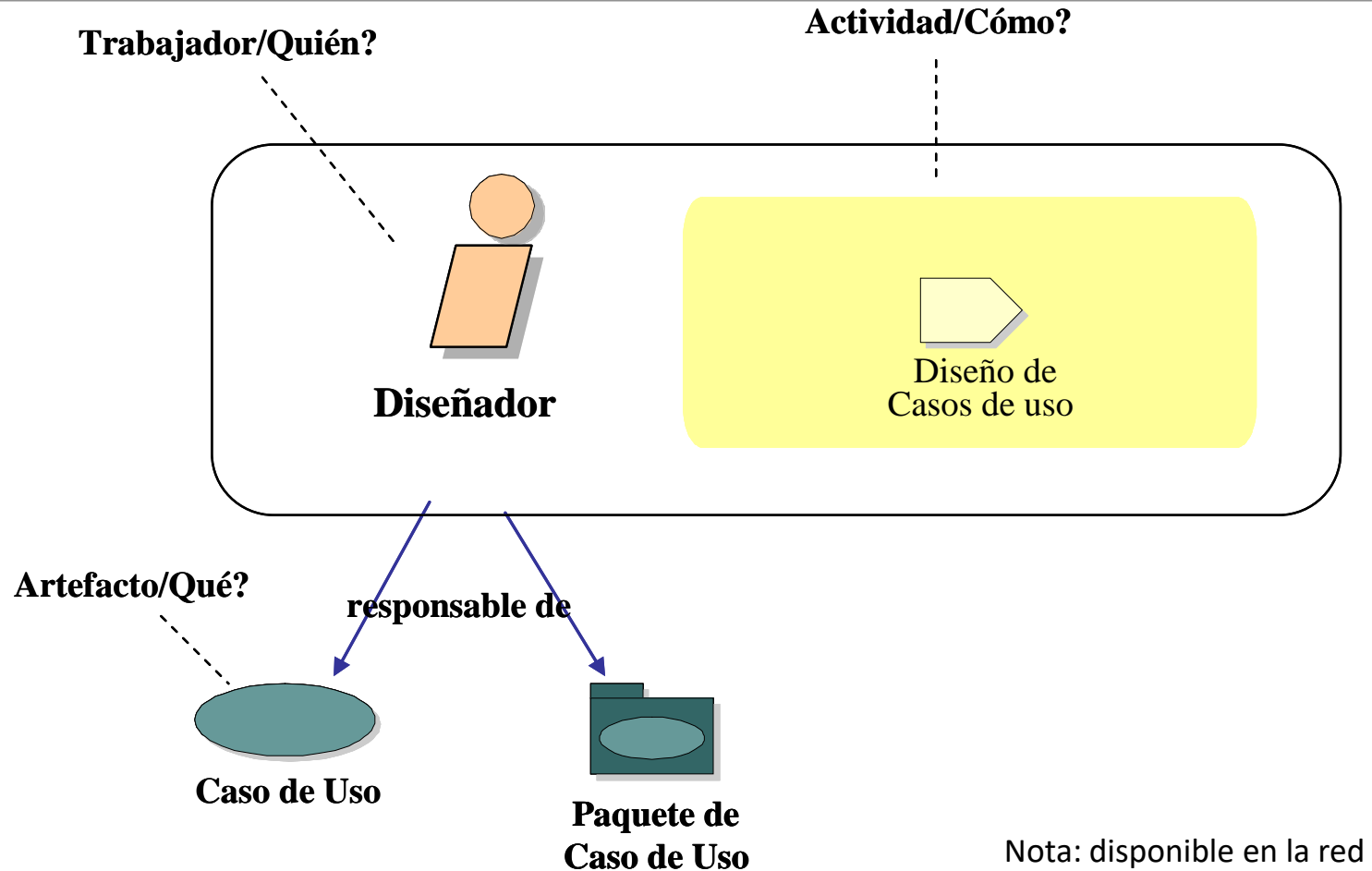
¿Cómo? → actividades

¿Qué? → artefactos (productos)

¿Cuándo? → flujos de trabajo



# Elementos de la estructura estática



# Roles

---

- Individuo o grupo de individuos con cierto comportamiento y responsabilidad.
- El comportamiento se expresa en términos de las *actividades* desempeñadas por el individuo.
- La responsabilidad se expresa en relación a los *productos de software* que el individuo crea, modifica o controla.
- Un individuo puede desempeñar varios roles, o un rol específico puede ser desempeñado por varios individuos.

# Grupos de roles

---

Agrupados por el tipo de actividades que desarrollan:

- **Analistas**: de procesos de negocio, analista del sistema, analista de requisitos
- **Desarrolladores**: arquitecto de sw, diseñador, diseñador de GUI, diseñador de BD, programador, integrador
- **Administradores**: líder de proyecto, de control de cambios, de configuración, de pruebas, ...
- **Apoyo**: documentador, especialista en herramientas, diseñador de cursos, artista gráfico, ...
- **Grupo de pruebas**: analista, diseñador, tester
- **Otros**: stakeholders, revisores, ...

# Actividades

---

- Es una unidad de trabajo que produce un resultado en el contexto del proyecto.
- La actividad tiene un objetivo (crear o modificar artefactos).
- Se asigna a un rol.

# Artefactos

---

- Son productos de trabajo bien definidos.
- Pueden servir de base para la definición de activos reutilizables.
- Los artefactos de software pueden ser:
  - Modelos (de CU, de diseño, etc.).
  - Elementos de un modelo (clases, CU, subsistemas, etc.).
  - Documentos (caso de negocio, contrato, licencias, etc.).
  - Código fuente.
  - Programas ejecutables.

# Flujos de Trabajo

---

- Secuencia de actividades + relación entre ellas = resultado observable

- Flujos de trabajo de RUP:

1. Modelo de Negocio.

2. Requisitos.

3. Análisis y Diseño.

4. Implementación.

5. Pruebas.

6. Liberación.

7. Administración del Proyecto.

8. Administración del Cambio y de la Configuración.

9. Ambiente.



**Ingeniería**



**Soporte**

# 1. Modelo de Negocio

---

- Permite entender mejor a la organización.
- **Objetivos del modelo de negocio:**
  - Entender la estructura y la dinámica de la organización.
  - Entender el problema actual e identificar potenciales mejoras.
  - Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización.
  - Derivar los requisitos del sistema para apoyar a la organización.

# 1. Modelo de Negocio (cont.)

---

- Permite: Visión de la organización
- A partir de la visión se definen procesos, roles y responsabilidades.
- **Artefactos:** modelo, glosario, etc.



## 2. Requisitos

---

- Uno de los flujos de trabajo más importantes.
- Establece qué tiene que hacer el sistema.
- Requisitos → base del contrato.

## 2. Requisitos

---

### Objetivos:

- Establecer y mantener un acuerdo entre clientes y otros *stakeholders* sobre lo que el sistema debe hacer.
- Que los desarrolladores entiendan los requisitos.
- Definir el ámbito del sistema.
- Base para la planeación de los contenidos técnicos de las iteraciones.
- Base para estimar costos y tiempo de desarrollo.
- Definir una interfaz de usuario para el sistema.

## 2. Requisitos (cont.)

---

Requisitos: funcionales y no funcionales.

- **Funcionales**: funcionalidad del sistema. Se modelan mediante diagramas de CU.
- **No funcionales**: atributos que debe mostrar el sistema (facilidad de uso, confiabilidad, eficiencia, portabilidad, etc.)

**¿De dónde?**: entrevistas, análisis de otros productos de sw, observación, etc.

**Artefactos**: especificación de requisitos, diseño de GUI, prototipos de GUI que se evalúan con el usuario final.

# 3. Análisis y Diseño

---

Traducir los requisitos a una notación que permita llegar a la implementación del sistema.

## **Objetivos del Análisis y Diseño:**

- Transformar los requisitos a un diseño.
- Definir /refinar la arquitectura.
- Adaptar el diseño al entorno de implementación.

**Análisis:** visión del sistema desde la perspectiva de los requisitos funcionales.

**Diseño:** refinamiento del análisis + requisitos no funcionales.

### 3. Análisis y Diseño (cont.)

---

- Al principio de la fase de elaboración: arquitectura candidata.
- Durante la fase de elaboración: refinar arquitectura hasta llegar a su versión final.

**Artefactos:** modelo de diseño, documentación de la arquitectura, etc.

# 4. Implementación

---

- Se implementa el sistema.
- Pruebas unitarias.
- En cada iteración:
  - Planear qué subsistemas deben ser implementados y en qué orden integrados → *Plan de Integración*.
  - Cada desarrollador decide en qué orden desarrolla los elementos del subsistema.
  - Los errores de diseño se notifican.
  - Subsistemas: se prueban individualmente.

## 4. Implementación (cont.)

---

- Integración: según el plan.
- Estructura de todos los elementos implementados → modelo de implementación.
- Prototipos para reducir riesgo.
- Prototipos pueden ser exploratorios (desechables) o evolutivos.

**Artefactos:** sistema ejecutable.

# 5. Pruebas

---

- Evaluar la calidad del producto.
- Esta disciplina brinda soporte a las otras disciplinas.
- **Objetivos:**
  - Encontrar y documentar defectos.
  - Da pautas sobre la calidad del software.
  - Validación de los supuestos hechos en el diseño y en la especificación de requisito.
  - Verificar las funciones del producto.
  - Verificar que cada requisito haya sido implementado.



## 5. Pruebas (cont.)

---

- Las actividades de este flujo se desarrollan desde el inicio del proyecto y continúan hasta el final.
- El flujo de trabajo: planear qué se va a probar, cómo se va a probar, hacer lo necesario para realizar las pruebas, ejecutar las pruebas, comunicar y registrar los resultados.

**Artefactos:** plan de pruebas, casos de prueba, etc.

# 6. Liberación

---

**Objetivo:** asegurar la aceptación (y adaptación) del producto por parte de los usuarios.

Las actividades a desarrollar son:

- Entregar el producto al cliente/usuario
- *Probar* el producto en el ambiente de operación.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Capacitar a los usuarios (o responsables de ventas).
- Migrar el software existente o convertir bases de datos.

## 6. Liberación (cont.)

---

- Inicia en fases anteriores: con actividades de planificación, elaboración del manual de usuario y tutoriales.
- Se desarrolla con mayor intensidad en la fase de transición.
- **Artefactos:** manuales, software empaquetado, cursos de capacitación, etc.

# 7. Administración del Proyecto

---

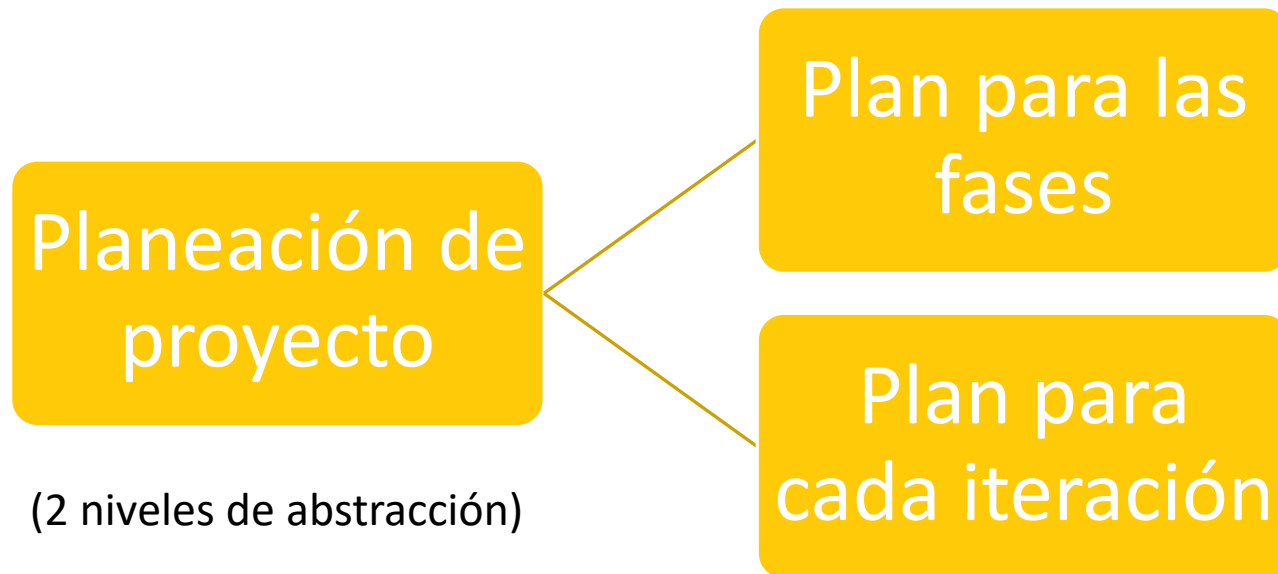
Lograr un balance entre objetivos, riesgos y restricciones con el desarrollo de un producto que satisfaga los requisitos de clientes y usuarios.

## **Objetivos:**

- Proveer un marco de trabajo para la adm. de proyectos de software.
- Proveer guías para planear, adm. personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

## 7. Administración del Proyecto (cont.)

---



**Artefactos:** planeaciones, bitácoras de riesgos, presupuestos, estimaciones, etc.

# 8. Administración del Cambio y de la Configuración

---

- Mantener la integridad de todos los artefactos que se crean en el proceso.
- Mantener información del proceso evolutivo que han seguido.
- **Artefactos:** plan de proyecto, bitácoras de versiones, etc.

# 9. Ambiente

---

- Dar soporte al proyecto con herramientas, procesos y métodos.
- Definir la instancia concreta del proceso que se va a seguir.
- Responsabilidades de este flujo de trabajo:
  - Seleccionar y adquirir las herramientas.
  - Instalar y configurar las herramientas.
  - Capacitar para el uso de herramientas.
  - Configurar el proceso.
  - Mejorar el proceso.
  - Servicios técnicos.

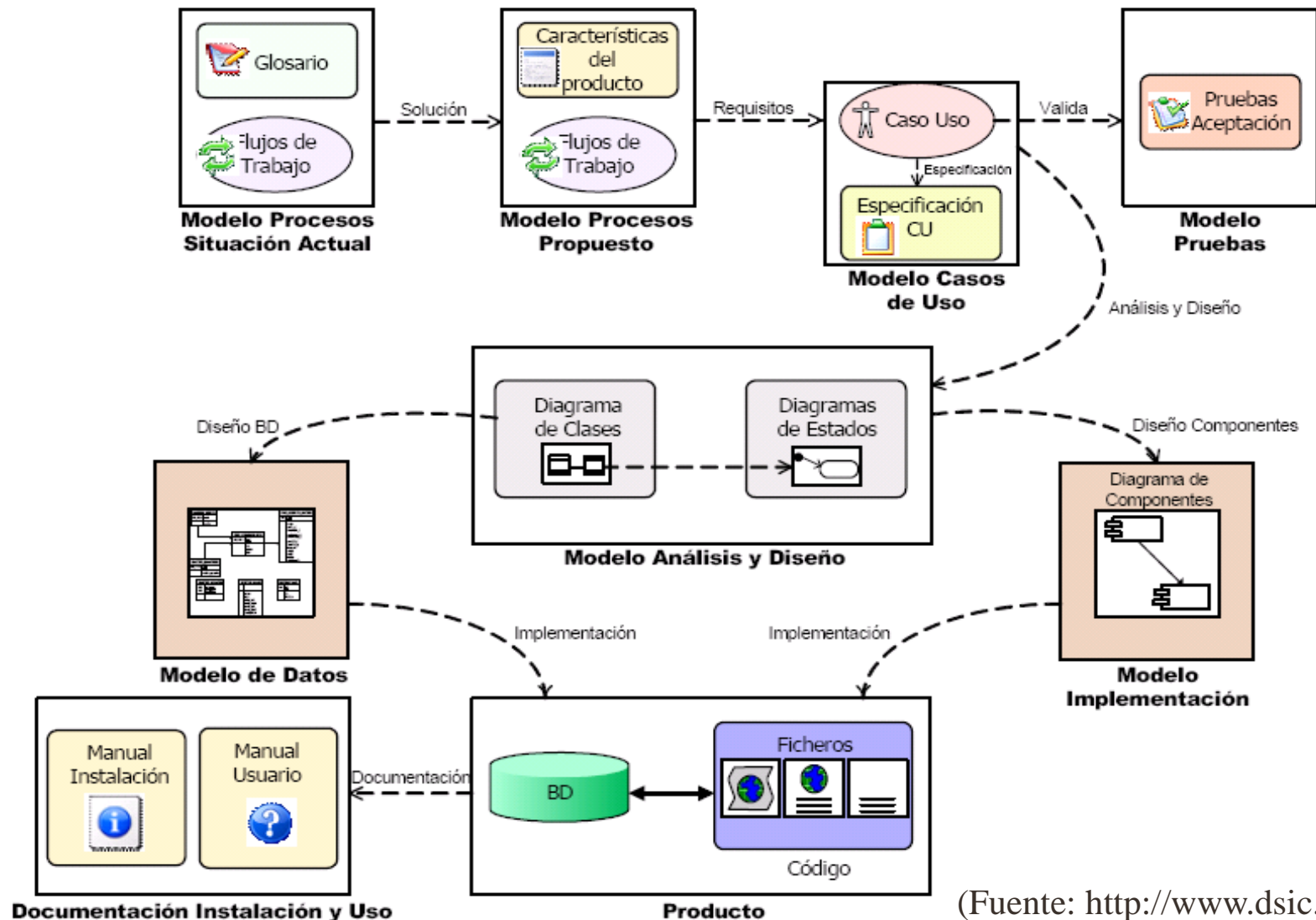
## 9. Ambiente (cont.)

---

**Artefactos:** plan de desarrollo (especifica como se aplicará el proceso, productos a utilizar y como se usarán) y guías para los distintos aspectos del proceso.



# Trazabilidad



(Fuente: <http://www.dsic.upv.es/>)

# Preguntas

---

1. ¿Qué es RUP?
2. ¿Diferencias entre RUP y las metodologías de desarrollo vistas?
3. ¿Diferencias entre RUP y metodologías ágiles?
4. ¿Les resulta aplicable?
5. ¿Qué productos –intermedios o finales- creen que sea importante generar durante todo el proyecto?
6. ¿Es necesario tener roles bien definidos?
7. Asignación de roles en los equipos: ¿rotación?

# Proyecto

---

Cada equipo usará RUP (ajustándolo) para el desarrollo del proyecto.

- ¿roles?
- ¿actividades?
- ¿artefactos?
- etc.

# Tarea

---

- Investigar y leer sobre: requisitos de sw. y casos de uso
- Anterior: Investigación de campo (metodologías usadas)