



# ¿Qué es RUP?



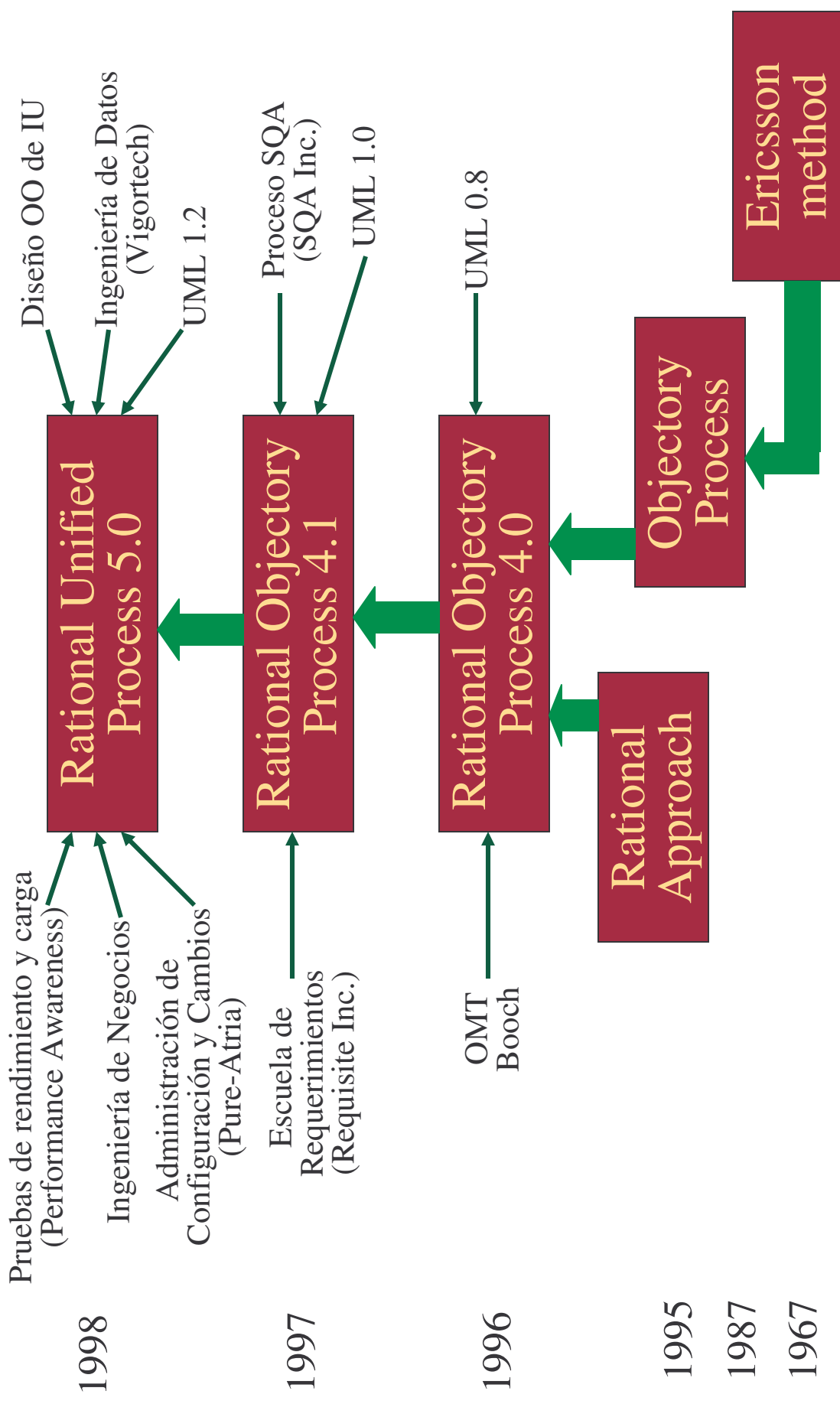
- **RUP es un proceso de desarrollo de software:**
  - Forma disciplinada de asignar tareas y responsabilidades en una empresa de desarrollo (quién hace qué, cuándo y cómo).
- **Objetivos:**
  - Asegurar la producción de software de calidad dentro de plazos y presupuestos predecibles. Dirigido por casos de uso, centrado en la arquitectura, iterativo (mini-proyectos) e incremental (versiones).
- **Es también un producto:**
  - Desarrollado y mantenido por Rational.
  - Actualizado constantemente para tener en cuenta las mejores prácticas de acuerdo con la experiencia.

# ¿Qué es RUP?

---

- Aumenta la productividad de los desarrolladores mediante acceso a:
  - Base de conocimiento, plantillas y herramientas.
- Se centra en la producción y mantenimiento de modelos del sistema más que en producir documentos.
- RUP es una guía de cómo usar UML de la forma más efectiva.
- Existen herramientas de apoyo a todo el proceso:
  - Modelamiento visual, programación, pruebas, etc.

# ¿Qué es RUP?



# Las mejores prácticas

---

- RUP pretende implementar las mejores prácticas actuales en ingeniería de software:
  - Desarrollo iterativo del software
  - Administración de requerimientos
  - Uso de arquitecturas basadas en componentes
  - Modelamiento visual del software
  - Verificación de la calidad del software
  - Control de cambios

# Desarrollo iterativo

---

- El software moderno es complejo y novedoso. No es realista usar un modelo lineal de desarrollo como el de cascada.
- Un proceso iterativo permite una comprensión creciente de los requerimientos a la vez que se va haciendo crecer el sistema.
- RUP sigue un modelo iterativo que aborda las tareas más riesgosas primero.
- Con esto se logra reducir los riesgos del proyecto y tener un subsistema ejecutable tempranamente.

# Administración de requerimientos

---

- RUP describe cómo:
  - Obtener los requerimientos
  - Organizarlos
  - Documentar requerimientos de funcionalidad y restricciones
  - Rastrear y documentar decisiones
  - Captar y comunicar requerimientos del negocio
- Los casos de uso y los escenarios indicados por el proceso han probado ser una buena forma de captar requerimientos y guiar el diseño, la implementación y las pruebas.

# Arquitecturas basadas en componentes

---

- El proceso se basa en diseñar tempranamente una arquitectura base ejecutable.
- La arquitectura debe ser:
  - Flexible
  - Fácil de modificar
  - Intuitivamente comprensible
  - Promueve la reutilización de componentes
- RUP apoya el desarrollo basado en componentes, tanto nuevos como preexistentes.



# Modelamiento visual

---

- Modelamiento visual de la estructura y el comportamiento de la arquitectura y los componentes.
- Bloques de construcción:
  - Ocultan detalles
  - Permiten la comunicación en el equipo de desarrollo
  - Permiten analizar la consistencia:
    - entre las componentes
    - entre diseño e implementación
- UML es la base del modelamiento visual de RUP.

# Verificación de cualidades

---

- No sólo la funcionalidad es esencial, también el rendimiento y la confiabilidad.
- RUP ayuda a planificar, diseñar, implementar, ejecutar y evaluar pruebas que verifiquen estas cualidades.
- El aseguramiento de la calidad es parte del proceso de desarrollo y no la responsabilidad de un grupo independiente.

# Control de cambios

---

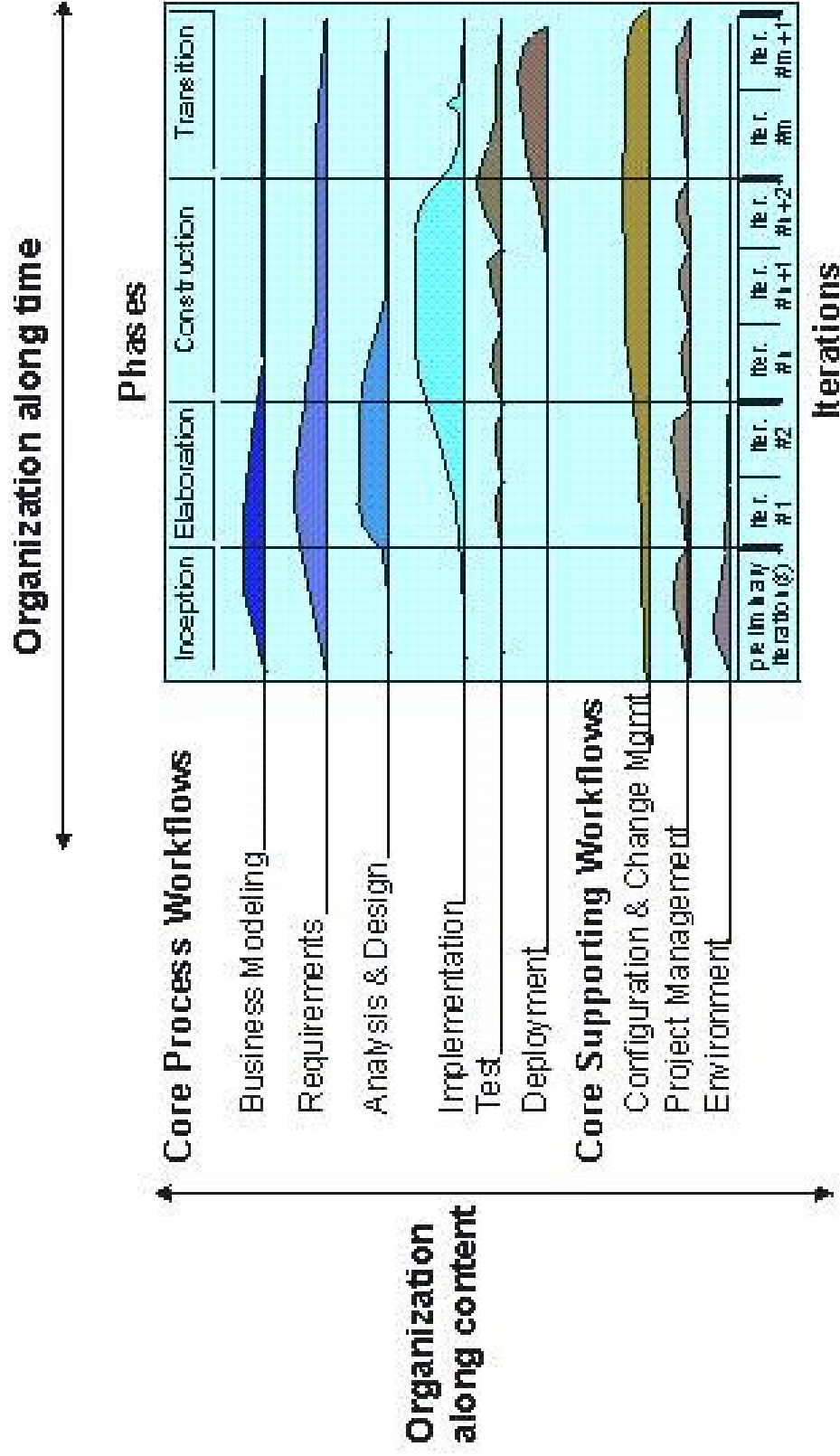
- Los cambios son inevitables, pero es necesario evaluar si éstos son necesarios y rastrear su impacto.
- RUP indica como controlar, rastrear y monitorear los cambios dentro del proceso iterativo de desarrollo.

# Ciclos y fases

---

- RUP divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo.
- Cada ciclo se divide en cuatro Fases:
  - Inicio
  - Elaboración
  - Construcción
  - Transición
- Cada fase concluye con un hito bien definido donde deben tomarse ciertas decisiones.

# Fases de RUP



# Fases de RUP: Inicio

---

- Se establece la oportunidad y alcance el proyecto.
- Se identifican todas las entidades externas con las que se trata (actores) y se define la interacción a un alto nivel de abstracción:
  - Identificar todos los casos de uso
  - Describir algunos en detalle
- La oportunidad del negocio incluye:
  - Criterios de éxito
  - Identificación de riesgos
  - Estimación de recursos necesarios
  - Plan de las fases incluyendo hitos

# Fases de RUP: Inicio

---

## Productos:

- Un documento de visión general:
  - Requerimientos generales del proyecto
  - Características principales
  - Restricciones
- Modelo inicial de casos de uso (10% a 20 % listos).
- Glosario.
- Caso de negocio:
  - Contexto
  - Criterios de éxito
  - Pronóstico financiero
- Identificación inicial de riesgos.
- Plan de proyecto.
- Uno o más prototipos.

# Fases de RUP: Inicio

---

## Hito:

Objetivos del  
Ciclo de Vida



- Las partes interesadas deben acordar el alcance y la estimación de tiempo y costo.
- Comprensión de los requerimientos plasmados en casos de uso.



# Fases de RUP: Elaboración

---

- Objetivos:
  - Analizar el dominio del problema
  - Establecer una arquitectura base sólida
  - Desarrollar un plan de proyecto
  - Eliminar los elementos de mayor riesgo para el desarrollo exitoso del proyecto
- Visión de “una milla de amplitud y una pulgada de profundidad” porque las decisiones de arquitectura requieren una visión global del sistema.

# Fases de RUP: Elaboración

---

## Productos:

- Es la parte más crítica del proceso:
  - Al final toda la ingeniería “dura” está hecha
  - Se puede decidir si vale la pena seguir adelante
- A partir de aquí la arquitectura, los requerimientos y los planes de desarrollo son estables.
- Ya hay menos riesgos y se puede planificar el resto del proyecto con menor incertidumbre.
- Se construye una arquitectura ejecutable que contemple:
  - Los casos de uso críticos
  - Los riesgos identificados

# Fases de RUP: Elaboración

---

## Productos:

- Modelo de casos de uso (80% completo) con descripciones detalladas.
- Otros requerimientos no funcionales o no asociados a casos de uso.
- Descripción de la Arquitectura del Software.
- Un prototipo ejecutable de la arquitectura.
- Lista revisada de riesgos y del caso de negocio.
- Plan de desarrollo para el resto del proyecto.
- Un manual de usuario preliminar.

# Fases de RUP: Elaboración

---

## Hito:

Arquitectura de  
Ciclo de Vida



- Condiciones de éxito de la elaboración:
  - ¿Es estable la visión del producto?
  - ¿Es estable la arquitectura?
  - ¿Las pruebas de ejecución demuestran que los riesgos han sido abordados y resueltos?
  - ¿Es el plan del proyecto algo realista?
  - ¿Están de acuerdo con el plan todas las personas involucradas?

# Fases de RUP: Construcción

---

- En esta fase todas las componentes restantes se desarrollan e incorporan al producto.
- Todo es probado en profundidad.
- El énfasis está en la producción eficiente y no ya en la creación intelectual.
- Puede hacerse construcción en paralelo, pero esto exige una planificación detallada y una arquitectura muy estable.

# Fases de RUP: Construcción

---

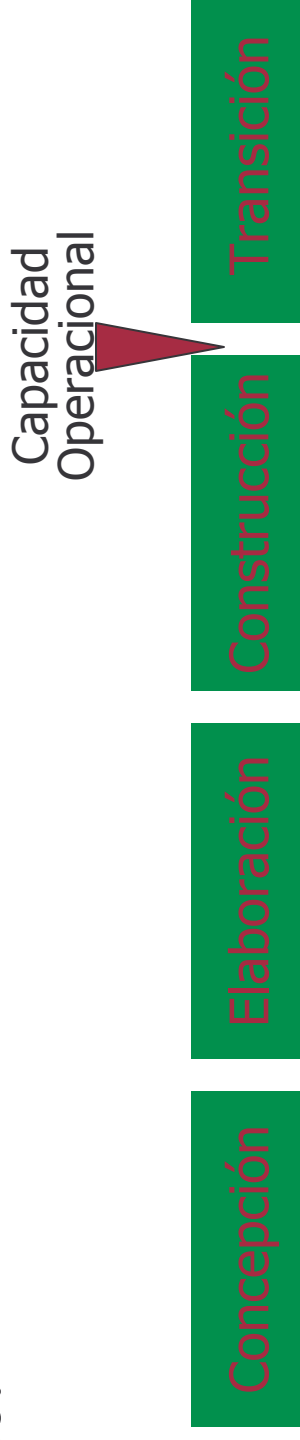
## Productos:

- El producto de software integrado y corriendo en la plataforma adecuada.
- Manuales de usuario.
- Una descripción del “release” actual.

# Fases de RUP: Construcción

---

## Hitos:



- Se obtiene un producto Beta que debe decidirse si puede ponerse en ejecución sin mayores riesgos.
- Condiciones de éxito:
  - ¿El producto está maduro y estable para instalarlo en el ambiente del cliente?
  - ¿Están los interesados listos para recibirlo?

# Fases de RUP: Transición

---

- El objetivo es traspasar el software desarrollado a la comunidad de usuarios.
- Una vez instalado surgirán nuevos elementos que implicarán nuevos desarrollos (ciclos).
- Incluye:
  - Pruebas Beta para validar el producto con las expectativas del cliente
  - Ejecución paralela con sistemas antiguos
  - Conversión de datos
  - Entrenamiento de usuarios
  - Distribuir el producto



# Fases de RUP: Transición

---

## Objetivos:

- Obtener autosuficiencia de parte de los usuarios.
- Concordancia en los logros del producto de parte de las personas involucradas.
- Lograr el consenso cuanto antes para liberar el producto al mercado.

Concepción

Elaboración

Construcción

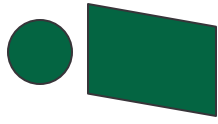
Transición



Producto

# Definiciones

---



## Trabajador

- Un *trabajador* define el comportamiento y las responsabilidades de un individuo.
- Es como un “sombbrero” que la persona usa durante el proyecto:
  - Una persona puede tener varios sombreros
  - Es el rol que desempeña en un momento dado
- Responsabilidades:
  - Hacer una serie de actividades
  - Ser el responsable de una serie de artefactos

# Definiciones

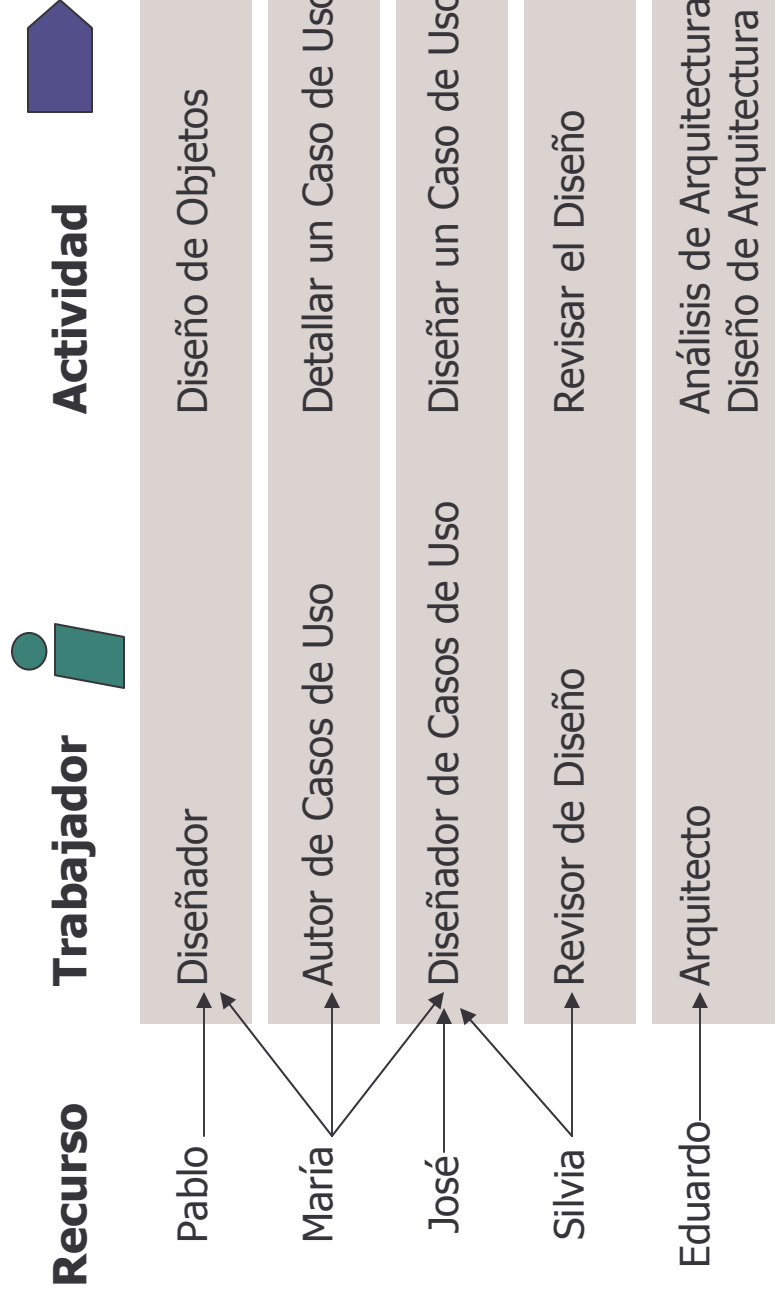
---

## Actividades

- Una actividad es una unidad de trabajo que se asigna a un trabajador. Ej.:
  - Crear o modificar un artefacto
- Una actividad lleva entre un par de horas y un par de días, involucra un solo trabajador y un número pequeño de artefactos.
- Las actividades se consideran en la planificación y evaluación del progreso del proyecto.
- Ejemplos:
  - Planificar una iteración - Administrador de proyecto
  - Encontrar actores y casos de uso - Analista
  - Revisar el diseño - Revisor de diseño
  - Ejecutar pruebas de performance - Ing. de pruebas de performance

# Asignación de actividades

---



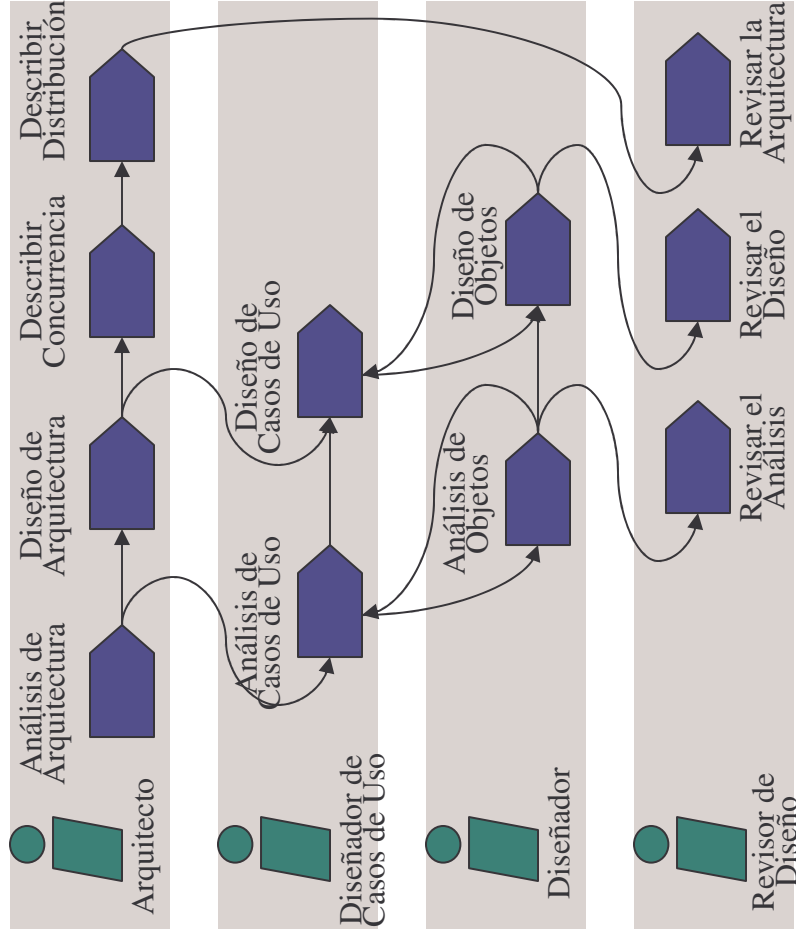
# Artefactos

---

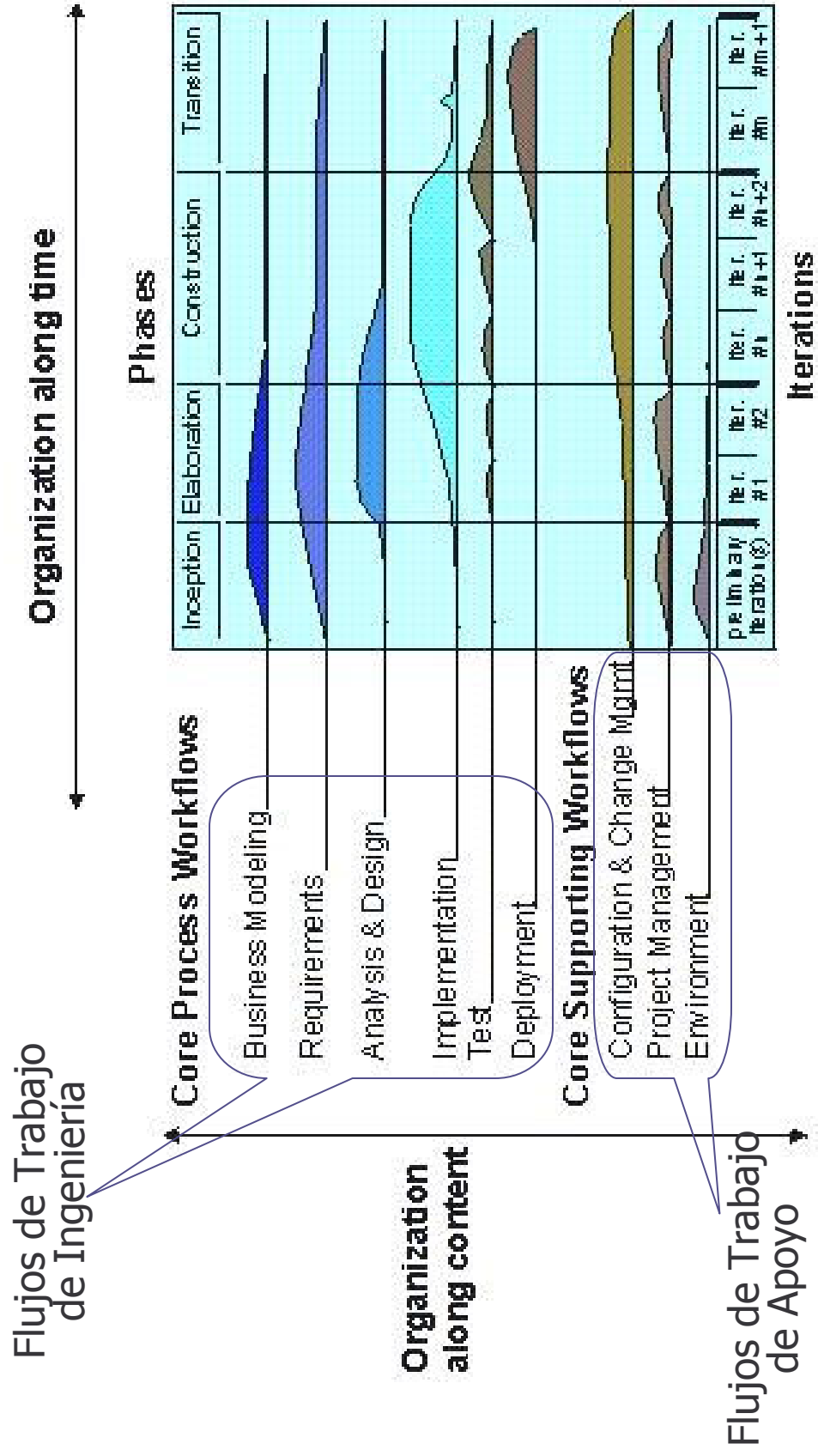
- Elementos de información producidos, modificados o usados por el proceso.
- Son los productos tangibles del proyecto.
- Son usados por los trabajadores para realizar nuevas actividades y son el resultado de esas actividades.
- Ejemplos:
  - Un modelo, como el modelo de casos de uso o el modelo de diseño.
  - Un elemento del modelo, como una clase o un caso de uso.
  - Un documento tal como el Caso del Negocio o la Arquitectura del Software.
  - Código fuente.
  - Código ejecutable.

# Flujos de trabajo

- Una lista de actividades, trabajadores y artefactos constituye un proceso.
- Un flujo de trabajo es una secuencia de actividades que produce un resultado valioso.
- No siempre es posible representar flujos de trabajo.



# Flujos de trabajo esenciales



# Flujos de trabajo

---

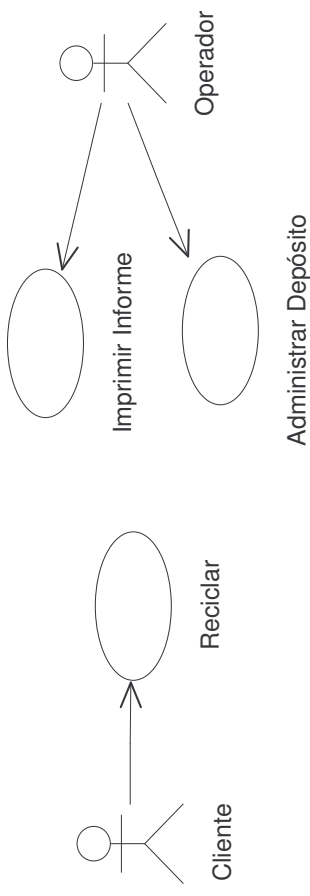
- Existen habitualmente problemas de comunicación entre ingenieros de software e ingenieros de negocios.
- RUP proporciona un lenguaje y proceso común para estos dos ámbitos.
- Para el modelamiento del negocio se usan “business use cases” (casos de uso del negocio):
  - La forma en que el software dará apoyo al negocio.



# Requerimientos

- Los desarrolladores y clientes deben acordar qué es lo que el sistema debe hacer:

- Relevar requerimientos
- Documentar funcionalidad y restricciones
- Documentar decisiones
- Identificar actores
- Identificar casos de uso



- Los casos de uso describen la funcionalidad.
- Los requerimientos no funcionales se incluyen en una especificación complementaria.

# Análisis y diseño

---

- Descripción de cómo se implementará el sistema: un plano
- Diseñar y validar la arquitectura es una tarea esencial.
- Debe:
  - Ejecutar las tareas y funciones descritas en los casos de uso
  - Satisfacer todos los requerimientos
  - Flexible a cambios
- El modelo de diseño consta de
  - Clases estructuradas en paquetes
  - Diseños de subsistemas con interfaces definidas (componentes)
  - Forma de colaboración entre las clases.
- El diseño se centra en la noción de arquitectura.

# Implementación

---

- **Propósito:**
  - Definir la organización del código
  - Implementar clases y objetos en forma de componentes (fuente, ejecutables, etc.)
  - Probar las componentes desarrolladas
  - Integrar las componentes en un sistema ejecutable

# Pruebas

---

- Propósito:
  - Verificar la interacción entre los objetos
  - Verificar la integración apropiada de componentes
  - Verificar que se satisfacen los requerimientos
  - Identificar los defectos y corregirlos antes de la instalación
- RUP describe como planear y ejecutar estas pruebas.
- RUP propone probar las componentes desde el principio:
  - Confiabilidad, funcionalidad y performance
- Las pruebas de regresión son importantes en desarrollos iterativos.
- Rational tiene herramientas para automatizar algunas pruebas.

# Distribución

---

- Producir un producto y hacerlo llegar a sus usuarios finales.
- Incluye varias actividades:
  - Producir un “release”
  - Empaquetar el software
  - Distribuir el software
  - Instalar el software
  - Apoyar a los usuarios
- A veces también incluye:
  - Realizar pruebas beta
  - Migración de datos
  - Aceptación formal
- La mayor parte de la distribución ocurre durante la transición.
- Este es uno de los flujos de trabajo menos documentados en RUP.

# Administración de proyectos

---

- Es el arte de balancear objetivos contrarios, manejar riesgos y producir software que satisfice a clientes y usuarios.
- Existen pocos proyectos realmente exitosos.
- RUP incluye:
  - Un framework para manejo de proyectos de software
  - Guías para planificación, provisión de personal, ejecución y monitoreo de planes
  - Un framework para manejar riesgos

# Administración de configuración y cambios

---

- Forma de controlar los artefactos producidos por las personas que trabajan en el proyecto.
- Algunos problemas habituales:
  - Actualizaciones simultáneas
  - Múltiples versiones
- RUP da guías para:
  - Desarrollos en paralelo
  - Automatizar la construcción
  - Administrar defectos

# Ambiente

---

- Ambiente y herramientas de desarrollo que harán posible llevar a cabo el proyecto.
- RUP guía en la configuración de un ambiente de proceso apropiado a cada proyecto.