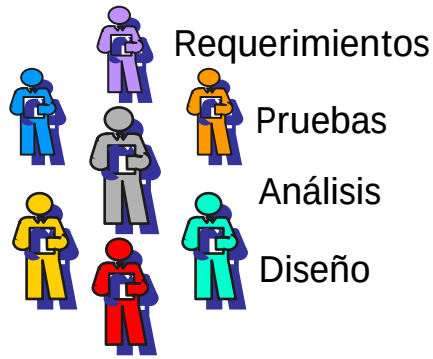
Rational Unified Process

Qué es un Proceso?

- Un proceso define **Quién** está haciendo **Qué**, **Cuándo** y **Cómo** para lograr un cierto objetivo. En la ingeniería de software el objetivo es construir un producto de software ó mejorar alguno existente.

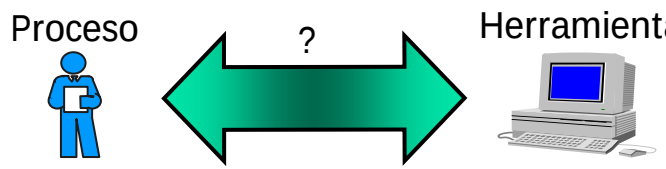
El Problema

•Si un proceso es utilizado, equipos funcionales diferentes normalmente utilizan procesos y lenguajes de modelación inconsistentes.



•La mayoría de los proyectos de software utilizan procesos que no están bien definidos. En su lugar los miembros del equipo (re)inventan sus propios procesos.

•Los procesos no están apropiadamente relacionados con herramientas, ó no están propiamente automatizados.



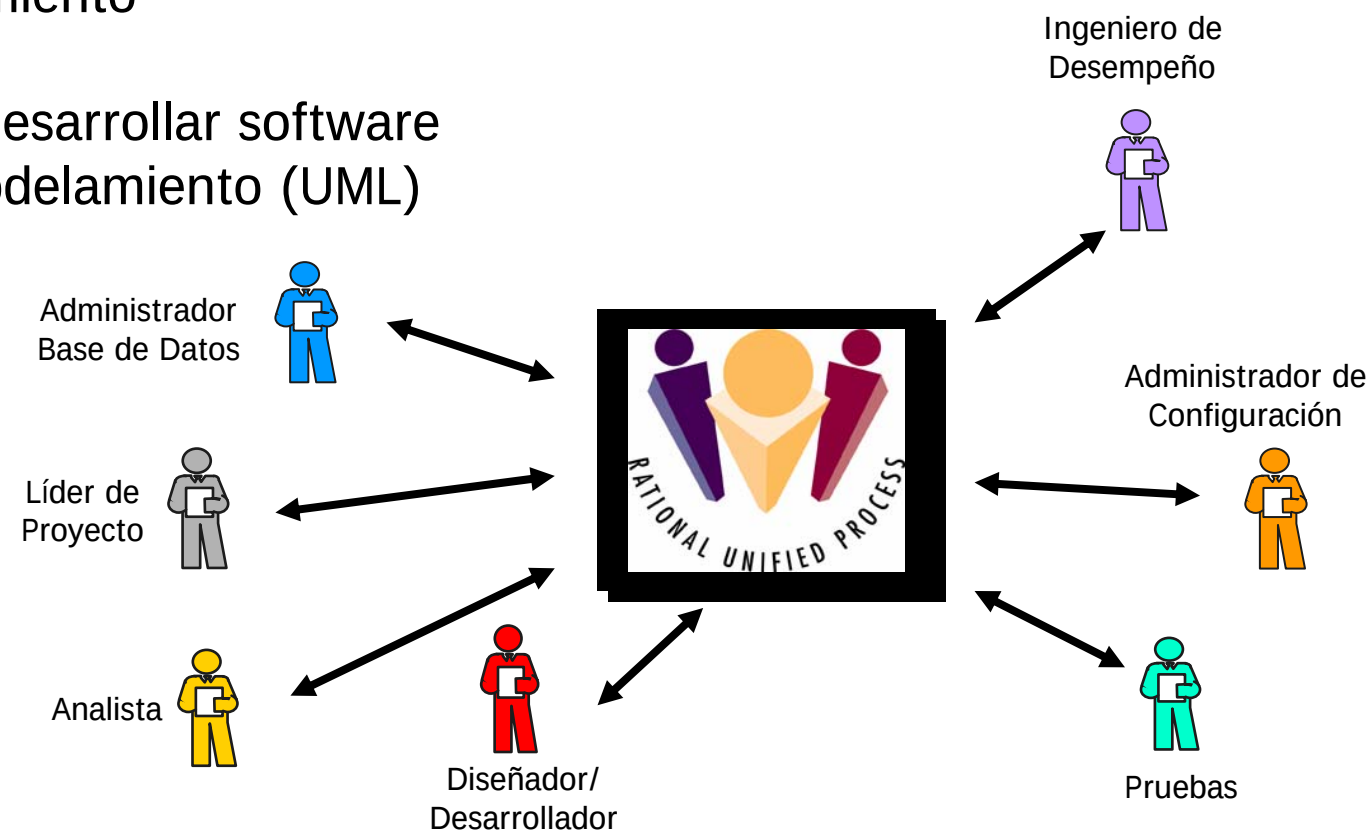
Rational Unified Process (RUP)

- Captura varias de las *mejores prácticas* en el desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones.
- Es una guía de cómo utilizar de manera efectiva *UML*.
- Provee a cada miembro de un equipo un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas de desarrollo.
- Crea y mantiene *modelos,* en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación.

Incremento de la Productividad en Equipo

Todos los miembros del equipo comparten

- 1 Base de conocimiento
- 1 Proceso
- 1 Vista de cómo desarrollar software
- 1 Lenguaje de modelamiento (UML)



6 Mejores Prácticas (Best Practices)

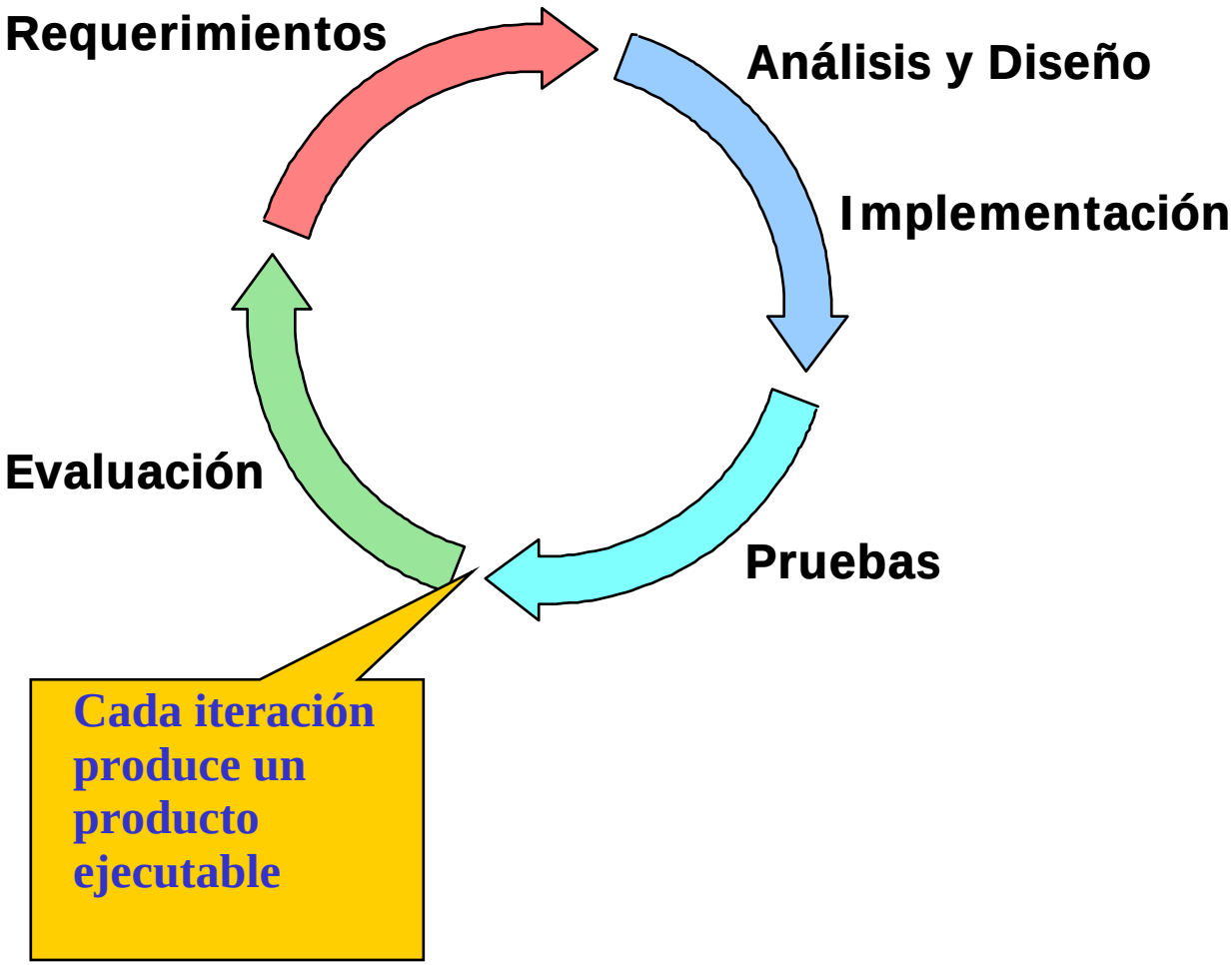
- RUP describe como utilizar de forma efectiva procedimientos comerciales probados en el desarrollo de software para equipos de desarrollo de software, conocidos como “mejores prácticas”.

Desarrollo Iterativo de Software

- Dados los sistemas de software sofisticados de la actualidad, no es posible hacer de manera secuencial la definición completa del problema, diseñar la solución completa, construir el software y por último probarlo.
- El descubrimiento de defectos en fases posteriores de diseño dan como resultado un aumento en los costos y/ó la cancelación del proyecto.

*El tiempo y dinero gastados en la implementación de un diseño fallido, son **no recuperables***

Desarrollo Iterativo



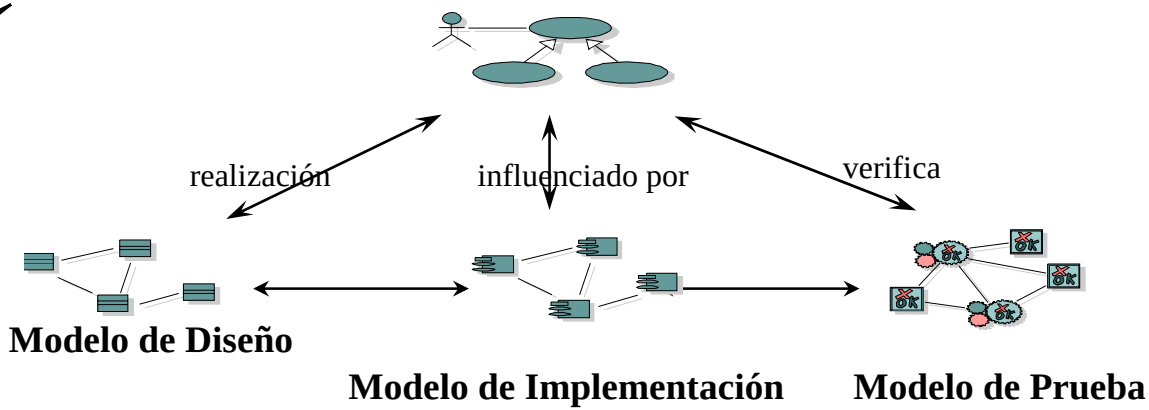
Características del Desarrollo Iterativo

- Permite un entendimiento incremental del problema a través de refinamientos sucesivos.
- Habilita una fácil retroalimentación de usuario.
- Metas específicas permiten que el equipo de desarrollo mantenga su atención en producir resultados.
- El progreso es medido conforme avanzan las implementaciones.

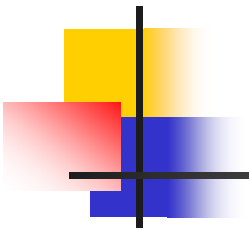
Administración de Requerimientos

- Licitar, organizar, y documentar la funcionalidad y restricciones requeridas.
- Llevar un registro y documentación de cambios y decisiones.
- Los requerimientos de negocio son fácilmente capturados y comunicados a través de casos de uso.
- Los casos de uso son instrumentos importantes de planeación.

Los casos de uso dirigen el trabajo desde el análisis hasta las pruebas

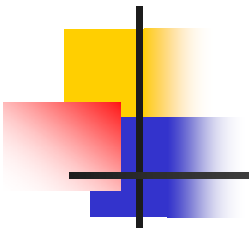


Arquitectura Basada en Componentes



- Se enfoca en el pronto desarrollo de una arquitectura ejecutable robusta.
  - Resistente al cambio mediante el uso de interfaces bien definidas.
  - Intuitivamente comprensible.
  - Promueve un reuso más efectivo de software.
  - Es derivada a partir de los casos de uso más importantes.

Modelación Visual de Software



- Captura la estructura y comportamiento de arquitecturas y componentes.
- Muestra como encajan de forma conjunta los elementos del sistema.
- Mantiene la consistencia entre un diseño y su implementación.
- Promueve una comunicación no ambigua.

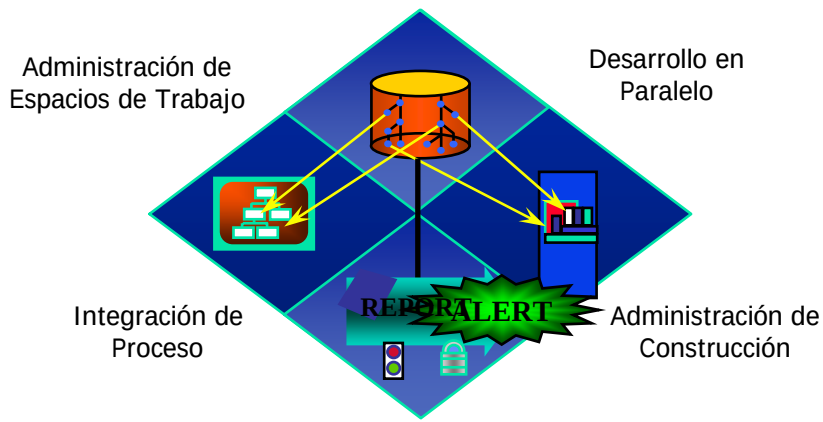
Verificación de la Calidad del Software

- Crea pruebas para cada escenario (casos de uso) para asegurar que todos los requerimientos están propiamente implementados.
- Verifica la calidad del software con respecto a los requerimientos basados en la confiabilidad, funcionalidad, desempeño de la aplicación y del sistema.
- Prueba cada iteración

Los problemas del software son de 100 a 1000 veces mas costosos de encontrar y reparar después del desarrollo

Control de Cambios del Software

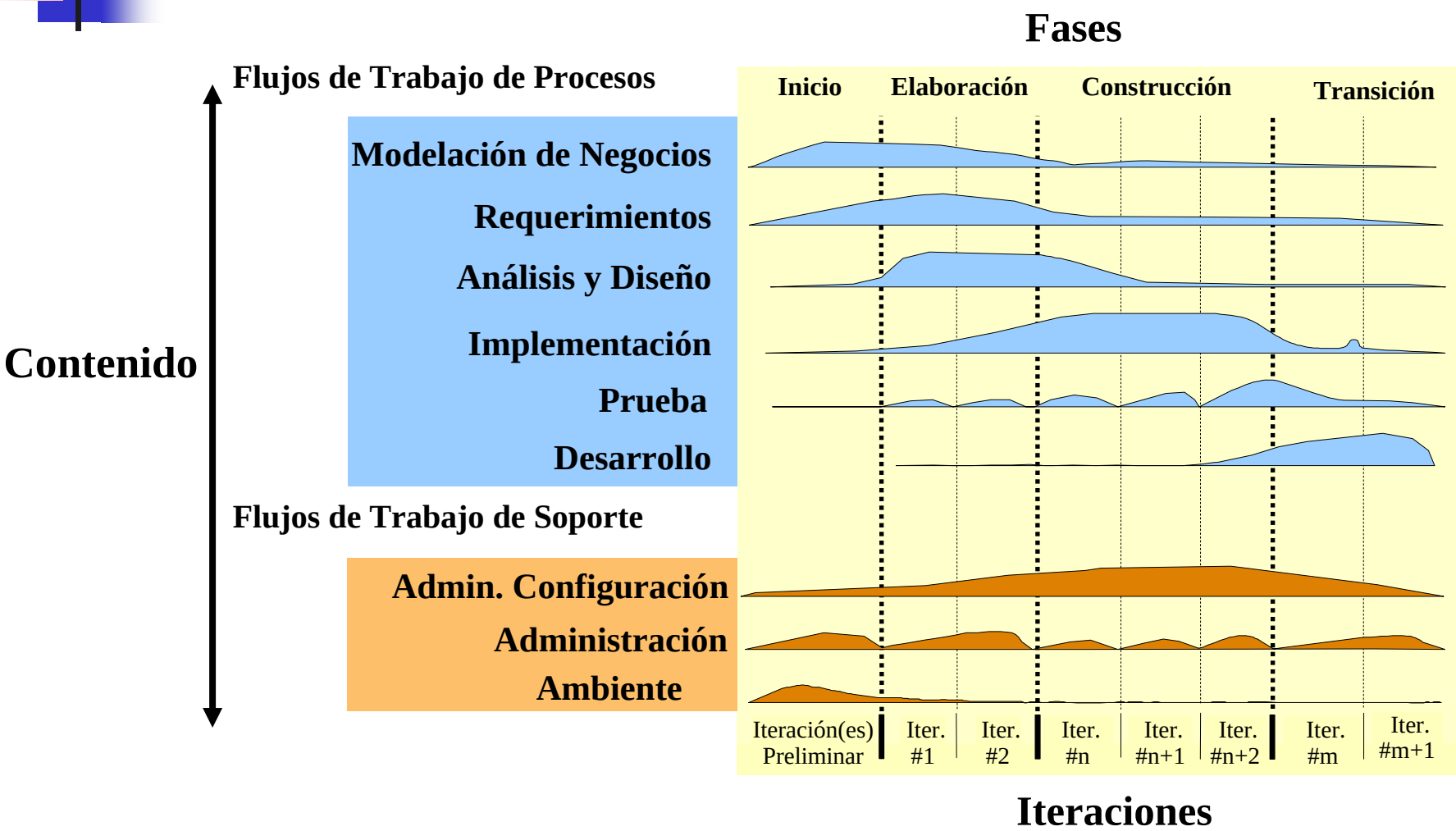
- Controlar, llevar un registro y monitorear cambios para permitir un desarrollo iterativo.
- Establece espacios de trabajo seguros para cada desarrollador
  - Provee aislamiento de cambios hechos en otros espacios de trabajo
  - Controla todos los artefactos de software – modelos, código, documentos, etc...



Estructura de RUP

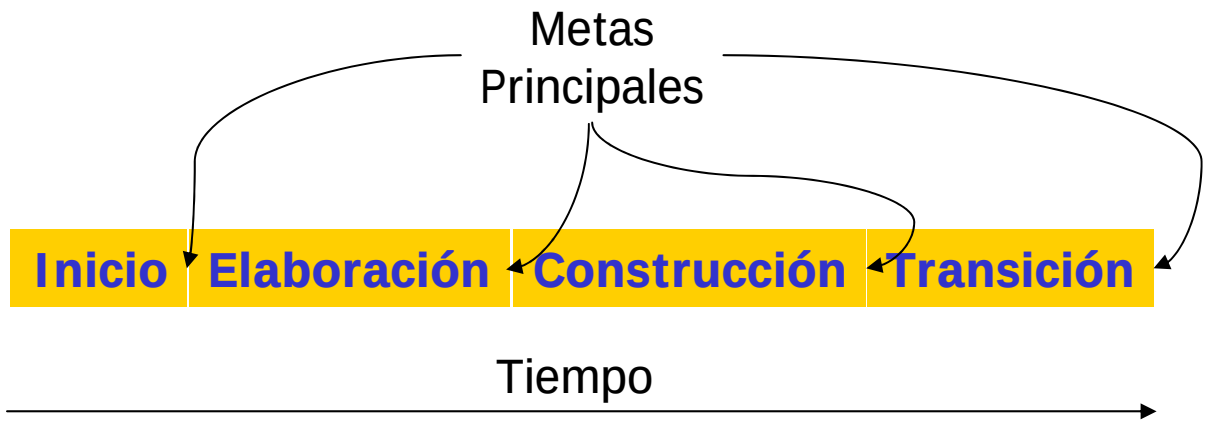
- El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes:
  - El eje horizontal representa *tiempo* y muestra el aspecto dinámico del proceso, expresado en términos de *ciclos*, *fases*, *iteraciones*, y *metas*.
  - El eje vertical representa el aspecto estático del proceso; como está descrito en términos de *actividades*, *artefactos*, *trabajadores* y *flujos de trabajo*.

Estructura de RUP Cont.



Fases en RUP

- **Inicio** – Define el alcance del proyecto
- **Elaboración** – Plan del proyecto, especificación de características, arquitectura base
- **Construcción** – Construir el producto
- **Transición** – Transición del producto a la comunidad del usuario

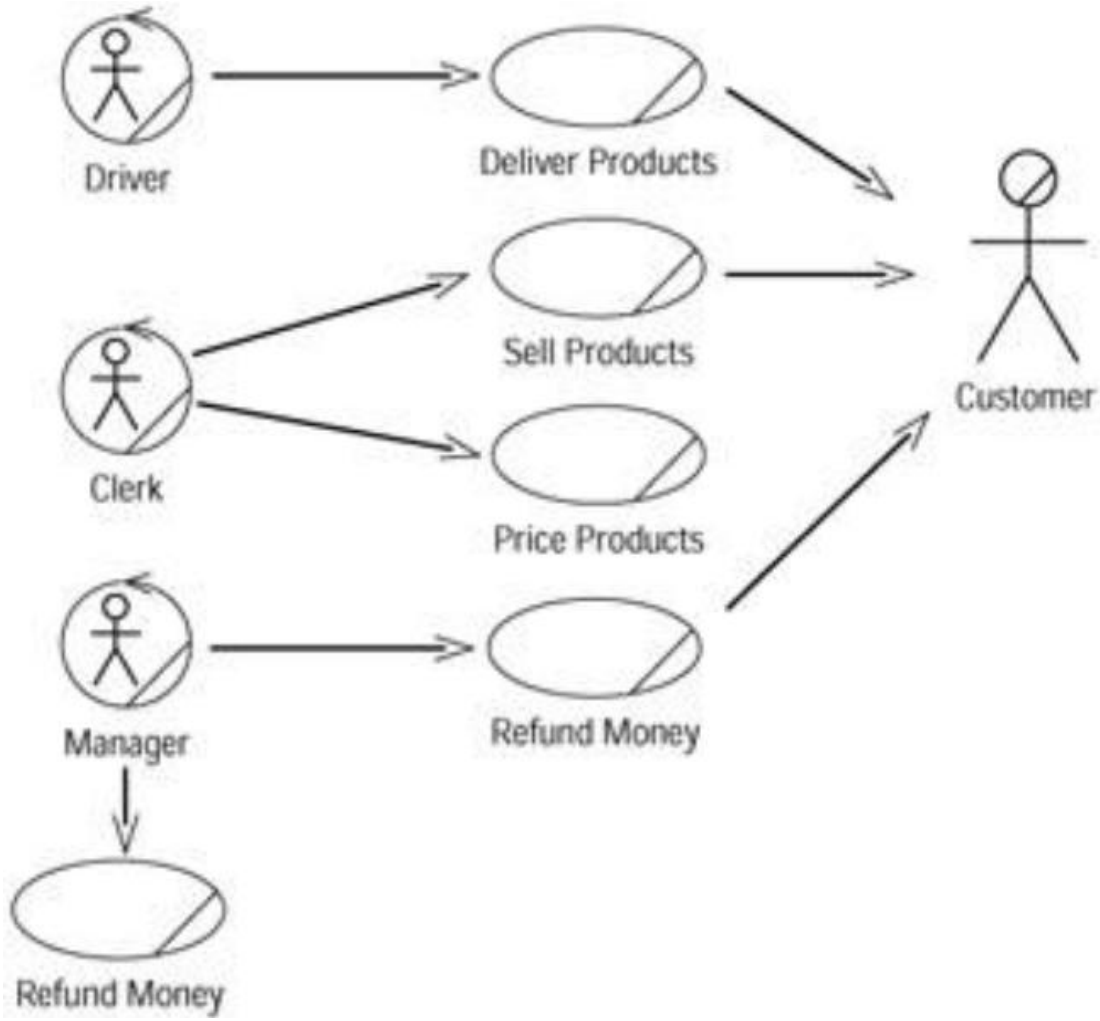


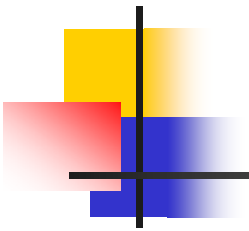
Fase de Inicio

- Propósito
  - Establecer caso de negocios para un nuevo sistema o para alguna actualización importante de un sistema existente
  - Especificar el alcance del proyecto
- Resultado
  - Una visión general de los requerimientos del proyecto, i.e., los requerimientos principales
    - Un modelo inicial de casos de uso (10-20%)
  - Un caso de negocios inicial, incluyendo:
    - Evaluación inicial de riesgos
    - Una estimación de los recursos requeridos

Ejemplo de Diagrama de Caso de Uso de Negocios

Caso de Negocios: modelar la empresa (como funciona la empresa a la que se le va a desarrollar el software)





Fase de Elaboración

- Propósito
  - Analizar el dominio del problema
  - Establecer una buena arquitectura
  - Lidiar con los elementos de riesgo más altos del proyecto
  - Desarrollar un plan comprensivo mostrando como el proyecto será completado
- Resultado
  - Un modelo de casos de uso 80% completo
  - Requerimientos suplementarios que capturen los requerimientos no funcionales y cualesquiera requerimientos que no estén asociados con un caso de uso específico
  - Una lista de riesgos revisada

Fase de Construcción

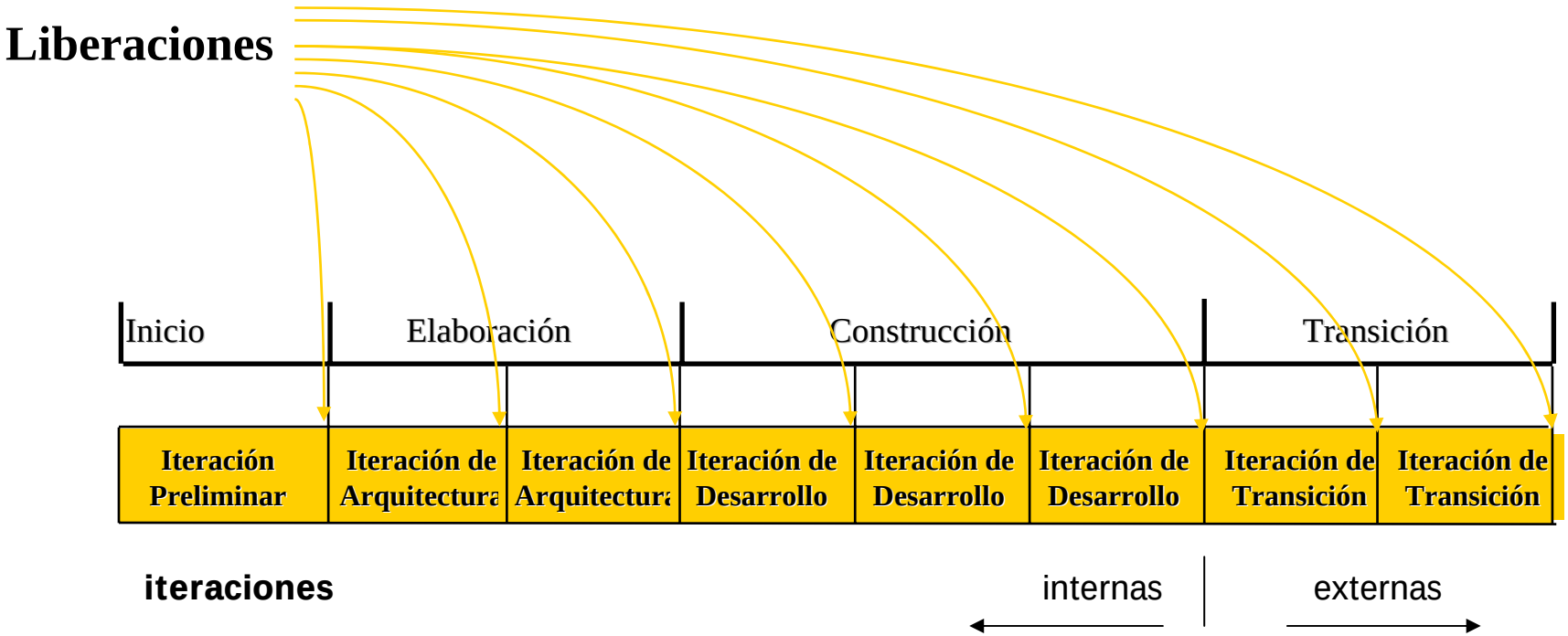
- Propósito
  - Desarrollar incrementalmente producto de software completo el cual estará listo para ser transferido al usuario
- Productos
  - Un modelo completo de diseño y casos de uso
  - Documentación de usuario
  - Una liberación “beta” del producto

Fase de Transición

- Hacer la transición final del producto de software al usuario
- Productos
  - Liberaciones ejecutables de producto
  - “Pruebas beta” para validar el nuevo sistema vs. las expectaciones del usuario
  - Manuales de usuario actualizados
  - Documentación de desarrollo actualizada
- Está el usuario satisfecho?
- Gastos reales de los recursos vs. Gastos previstos ← Aceptables?

Iteraciones

- Cada fase en RUP puede descomponerse en iteraciones. Una *iteración* es un ciclo de desarrollo completo dando como resultado una entrega de producto ejecutable (interna o externa)



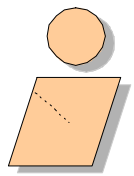
Noción de Proceso

Describe una unidad de trabajo que puede ser asignada a un trabajador.

Trabajador/Quién?

Actividad/Cómo?

***Rol* que puede ser desempeñado por un individuo o conjunto de individuos en la organización de desarrollo**



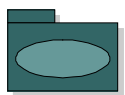
Diseñador

Diseño de
Casos de uso

responsable de

Artefacto/Qué?

Caso de Uso



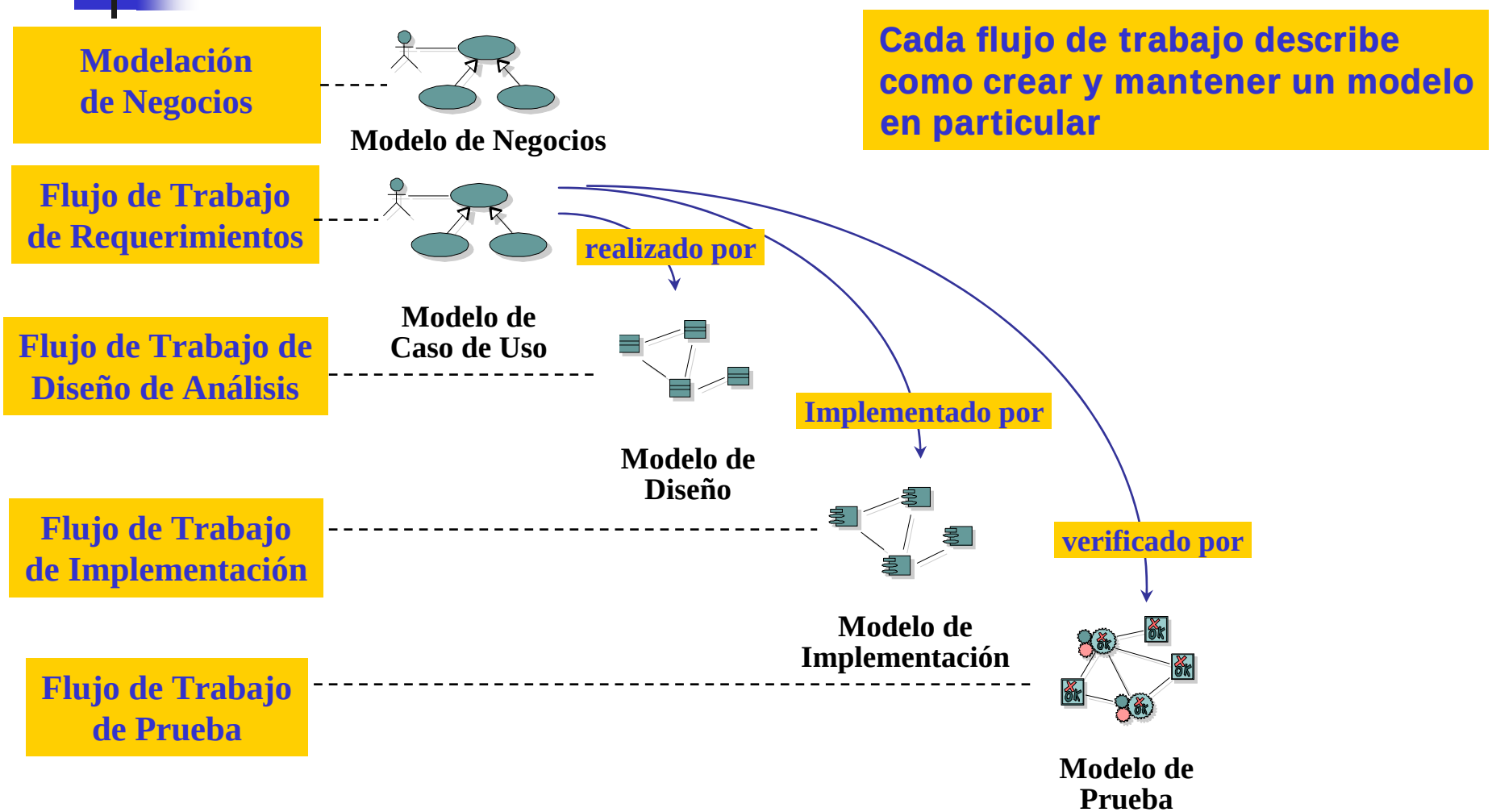
Paquete de Caso de Uso

Pieza de información que es producida, modificada, ó utilizada por un proceso

Modelos y Flujos de Trabajo

- Una mera enumeración de todos los trabajadores, actividades y artefactos no constituyen un proceso. Se necesita una forma de describir secuencias significativas que produzcan algún resultado válido, y que muestre la interacción entre trabajadores.
- Un *flujo de trabajo* es una secuencia de actividades que producen un resultado de valor observable.
- En términos de UML pueden ser expresados como un diagrama de secuencia, un diagrama de colaboración, ó como un diagrama de actividad.
- Los grupos de trabajo agrupan actividades en forma lógica

Modelos y Flujos de Trabajo Cont.



Referencias

- **A Simplified Approach to RUP**
  Gary K. Evans
  President, Evanetics, Inc.
  http://www.therationaledge.com/content/jan_01/t_rup_ge.html

- **UML y Patrones, Introducción al Análisis y Diseño Orientado a Objetos**
  Craig Larman
  Prentice-Hall

- **Rational Unified Process, Best Practices for Software Development Teams**
  A Rational Software Corporation White Paper