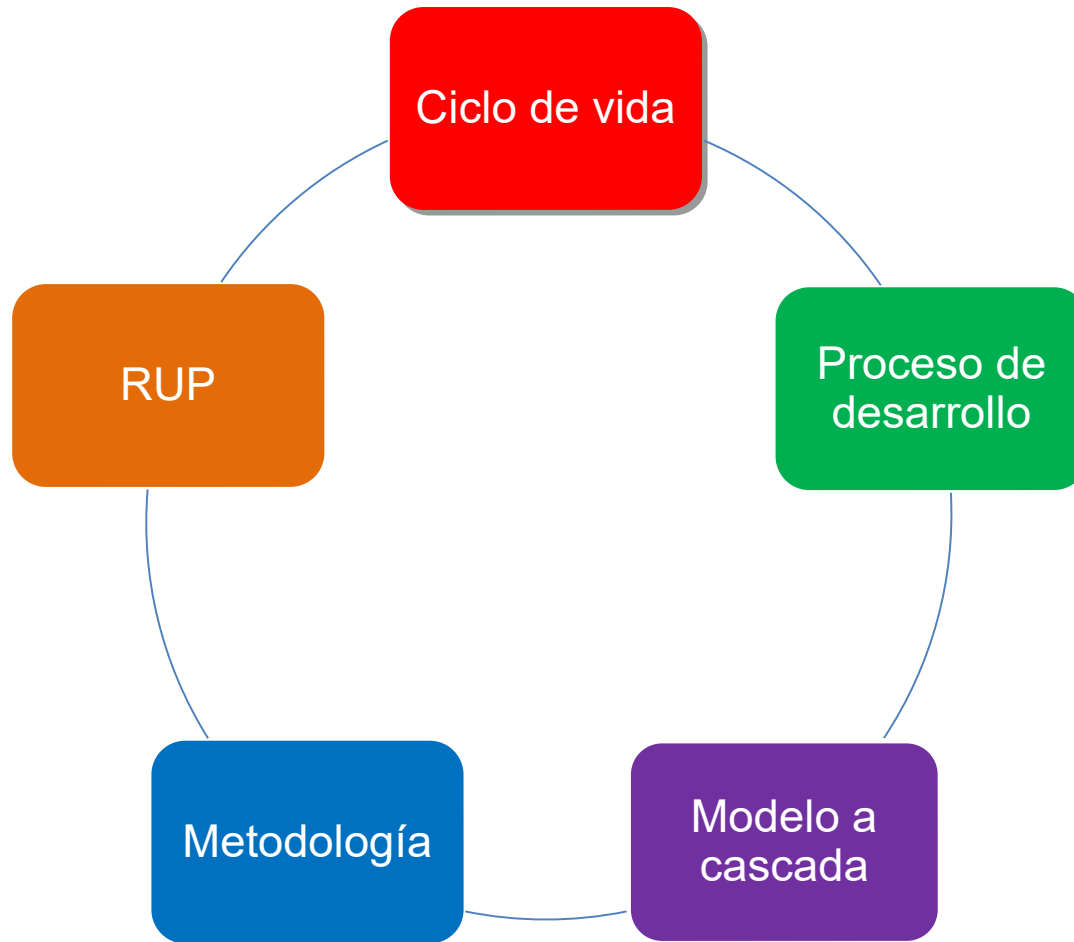


Análisis de Sistemas

CIF 5555

2022-1



EL CICLO DE VIDA DEL SOFTWARE

Síndrome del lavadero (requisitos imprevistos)

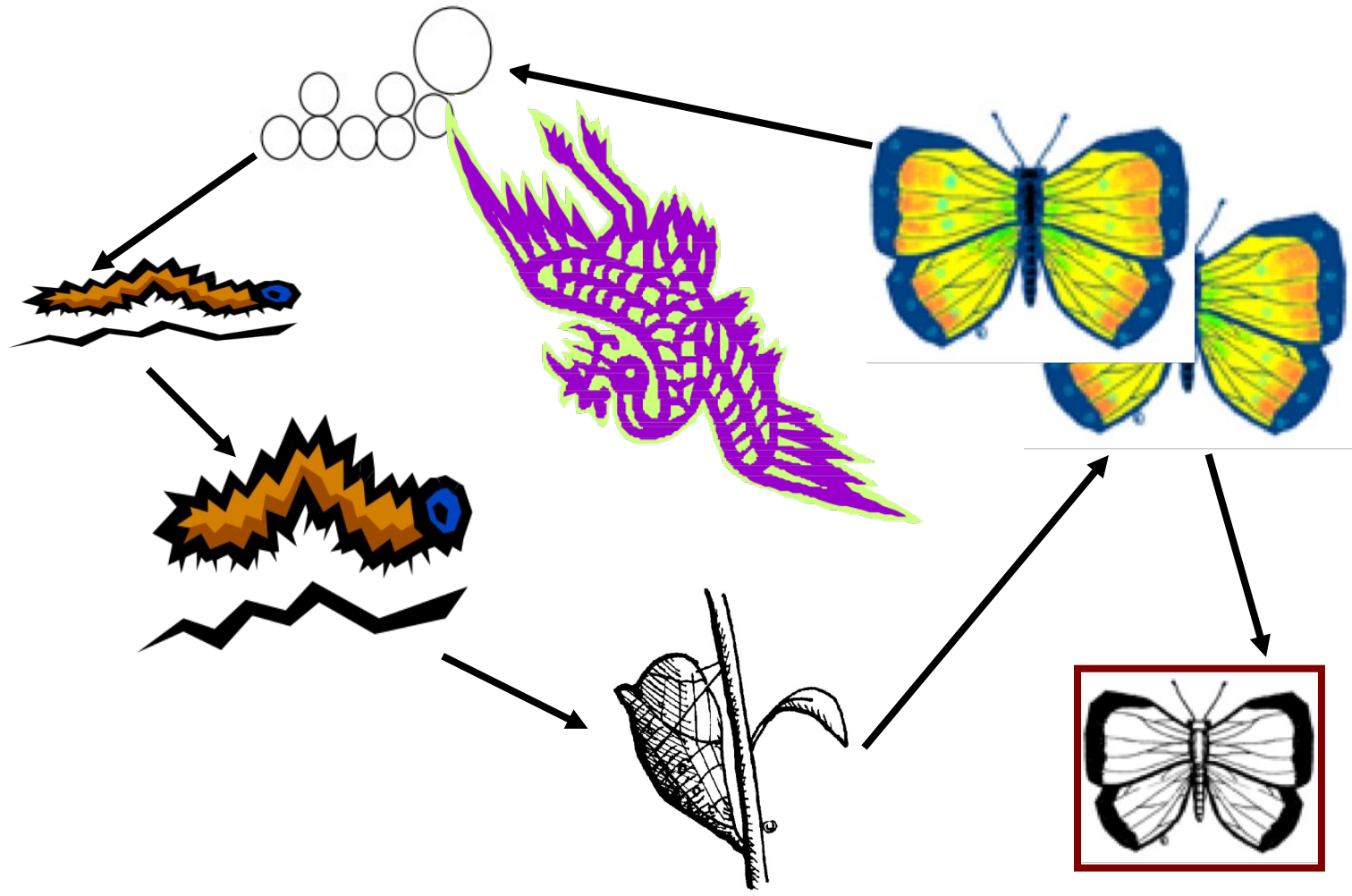


Desarrollo de software ad-hoc

- desarrollo ad-hoc : creando software sin ninguna guía o proceso formales
- ¿cuáles son algunas desventajas del desarrollo ad-hoc?
 - algunas acciones importantes (pruebas, diseño) pueden pasar siendo ignoradas
 - no es claro cuando comenzar o detenerse realizando cada tarea
 - no escala bien a muchas personas
 - no es fácil de revisar o evaluar el propio trabajo



El Concepto de Ciclo de Vida

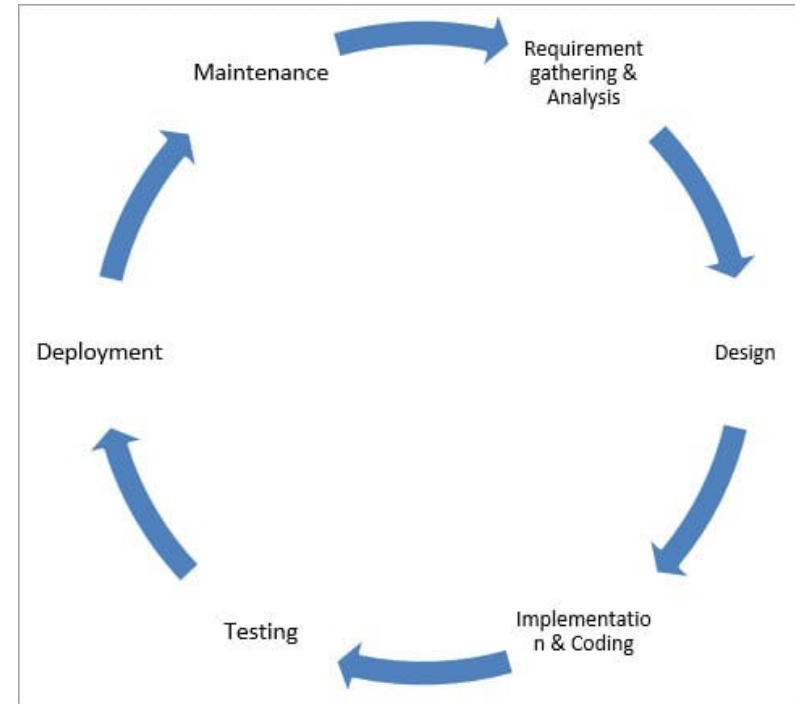


El ciclo de vida del software

Ciclo de vida del software:

series de pasos / fases, a través de las cuales se produce software

- puede tomar meses o años para completarse
- metas de cada fase:
 - señalar un conjunto claro de pasos a realizar
 - producir un documento o ítem tangible
 - contemplar la revisión del trabajo
 - especificar acciones a realizar en la siguiente fase



<https://www.softwaretestinghelp.com/software-development-life-cycle-sdlc/>

Modelo de ciclo de vida

- Organización de actividades
 - Clasificación de actividades
 - Criterios para finalizar y pasar a la siguiente

- Ejemplo: preparar un postre
 - Podemos dar un modelo de desarrollo genérico:
 - Ir de compras mientras el horno se calienta: amasar, meter en la sartén y hornear.
 - No es la receta del postre: es independiente del postre.

Fases del ciclo de vida del software

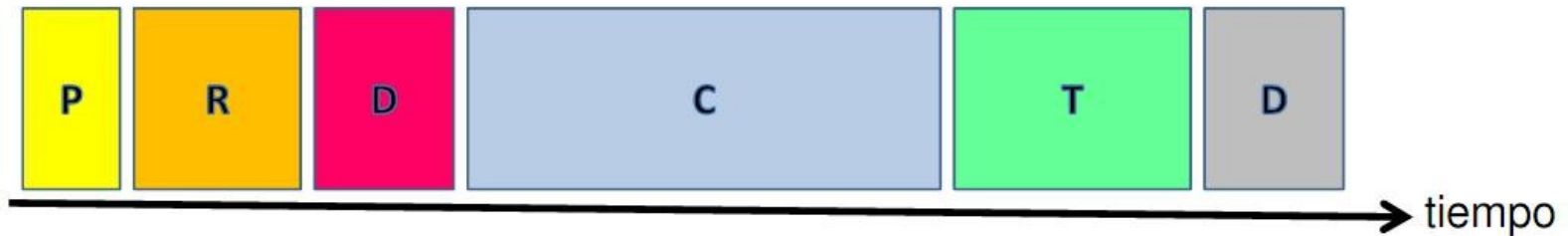
- Fases estándares
 - **Requisitos**, Análisis y Especificación
 - **Diseño**
 - **Construcción**, Codificación, Implementación, Integración
 - **Testing**, Profiling, Quality Assurance
 - **Despliegue**, Operación y Manutención
- Otras fases posibles
 - **Planificación**
 - **Operación y mantención**
 - **Ventas**
 - **QA**
 - **Marketing**

Definición del modelo de proceso



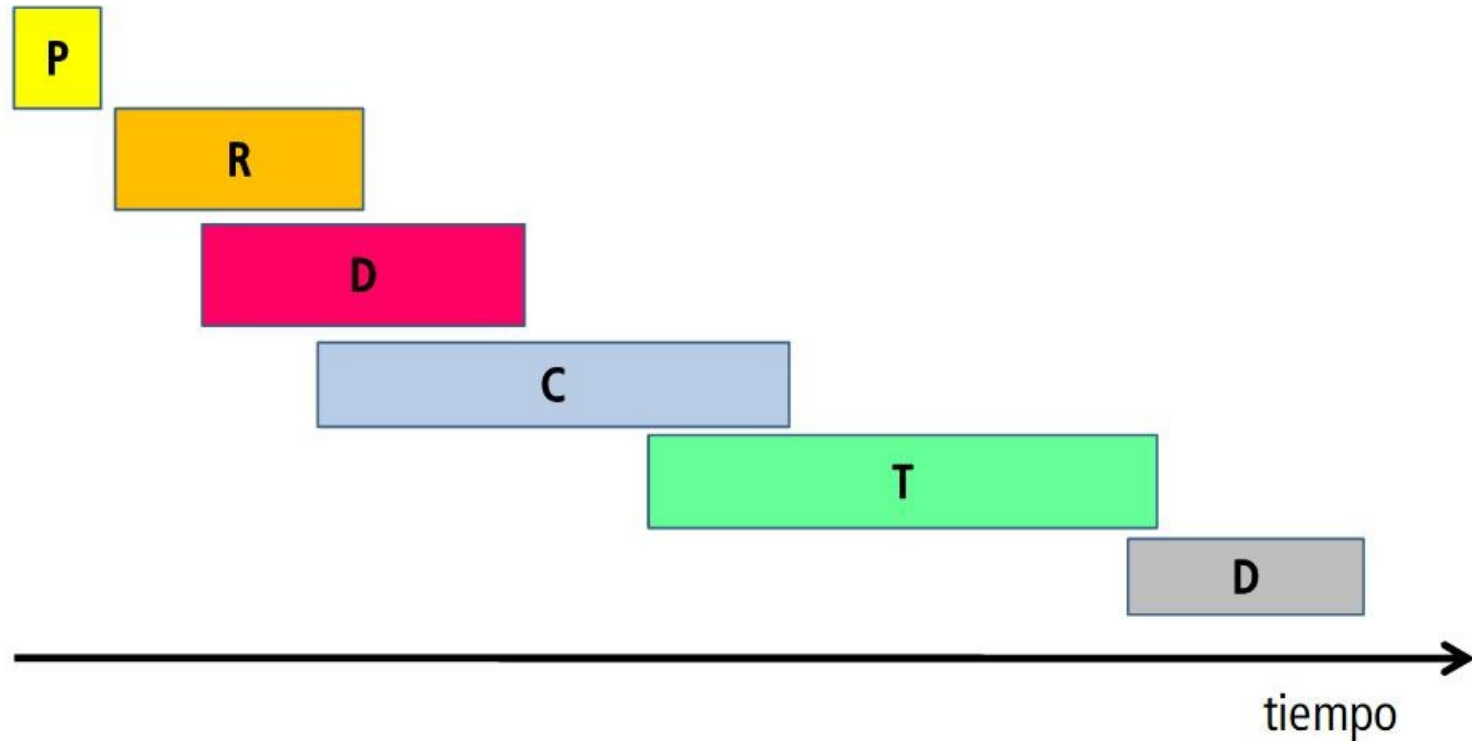
tiempo

Proceso secuencial o transaccional

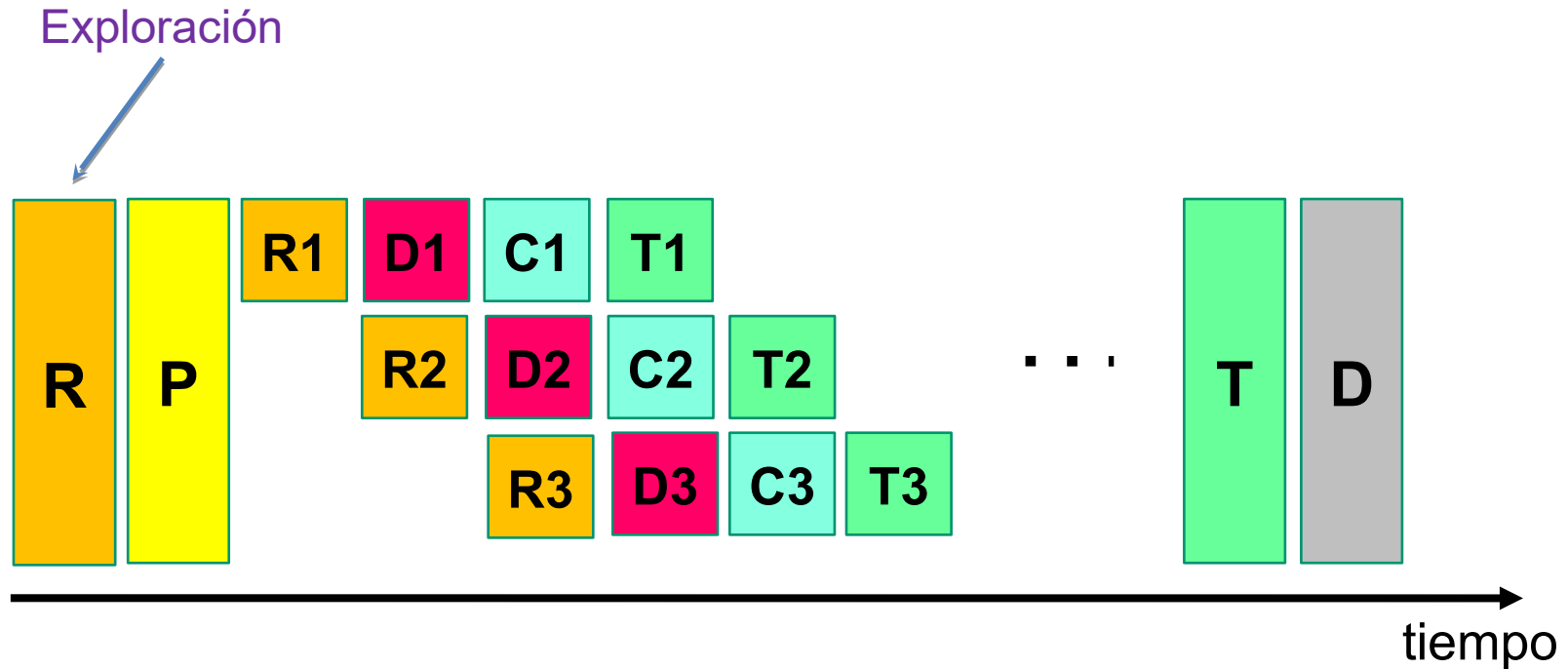


Parece que es mejor traslapar las actividades

Proceso en cascada



Proceso incremental



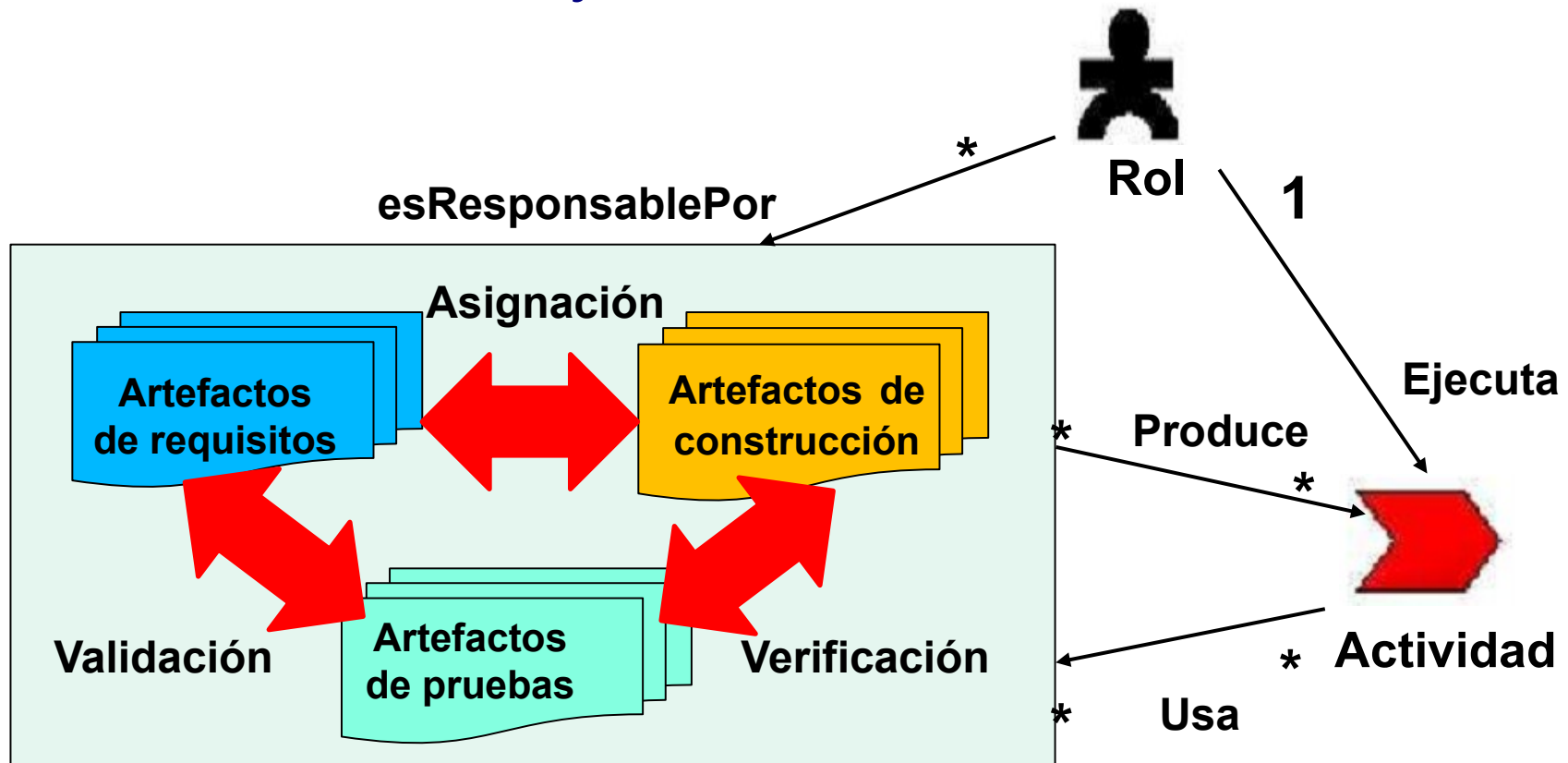
**Si se quiere validación frecuente con cliente →
Desarrollo Iterativo**

Modelos de proceso y metodologías

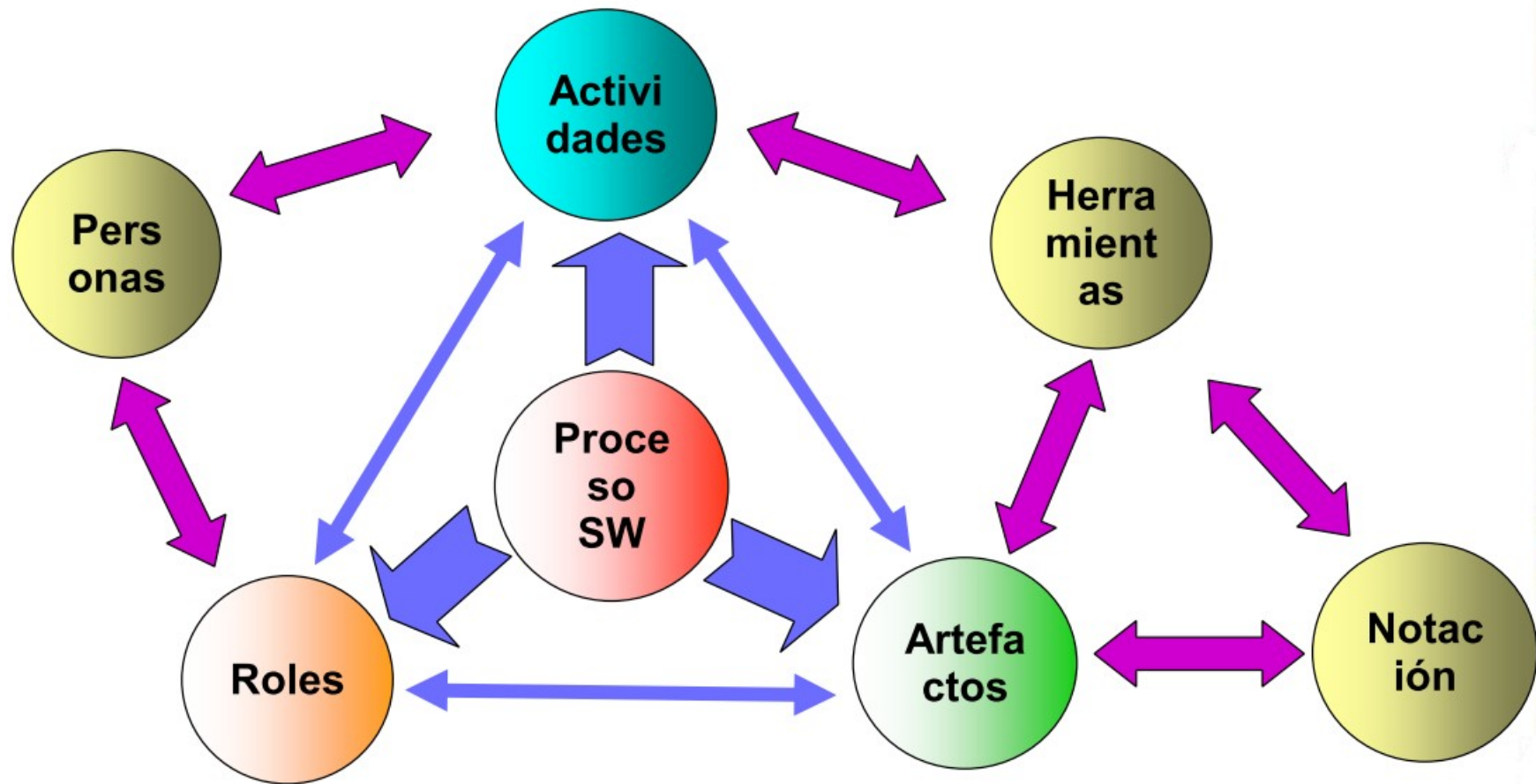
- El **proceso de desarrollo** tiende a ejecutarse iterativamente a través de las fases del **ciclo de vida del software** en vez de linealmente
- Se han propuesto varios modelos para describir este proceso para el ciclo de vida del software:
 - **Codificar y corregir (Code-and-fix)**
 - **Desarrollo en cascada**
 - **Desarrollo evolutivo**
 - Desarrollo incremental
 - Desarrollo en espiral
 - **Prototipado rápido**

¿Qué es una metodología?

- Una metodología define **Quién** debe hacer **Qué**, **Cuándo** y **Cómo** debe hacerlo



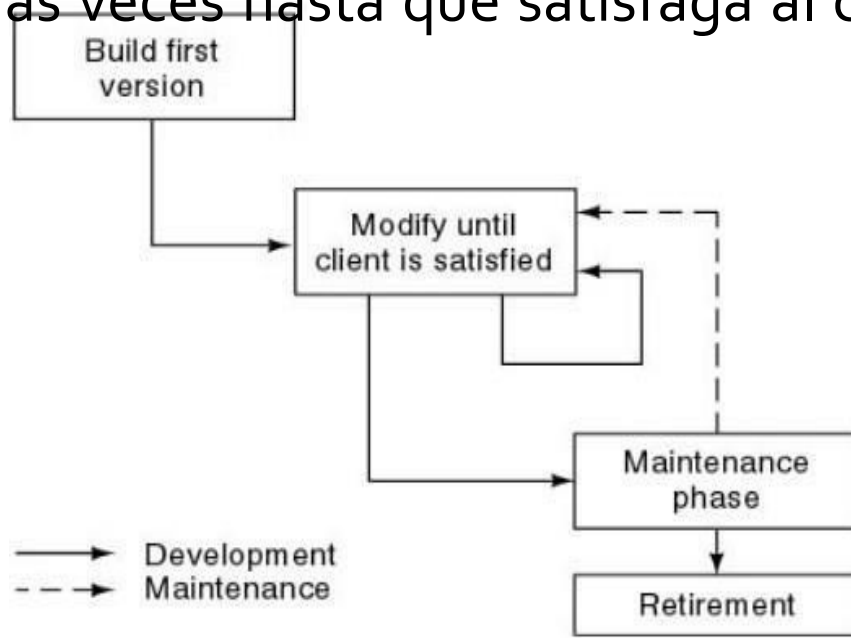
Elementos de una Metodología



EL PROCESO DE SOFTWARE Y MODELOS DE CICLO DE VIDA

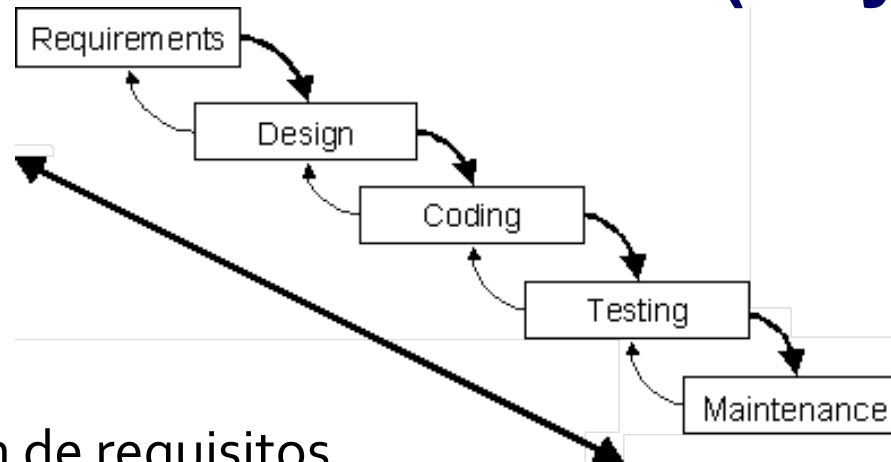
Modelo Build-and-Fix (Code-and-Fix)

- El producto se desarrolla sin especificación y sin intento de diseño.
- el desarrollador escribe un programa que luego se modifica varias veces hasta que satisfaga al cliente



- Quizás adecuado para un proyecto de código de 100 líneas
- Se vuelve totalmente impráctico para productos de tamaño razonable.
- Además, mantener un producto sin especificaciones o documentación que explique su diseño es extremadamente difícil.
- Antes de iniciar el desarrollo de un proyecto, debemos elegir un modelo de ciclo de vida "real".

Ciclo de Vida Tradicional: Modelo en Cascada (Royce)

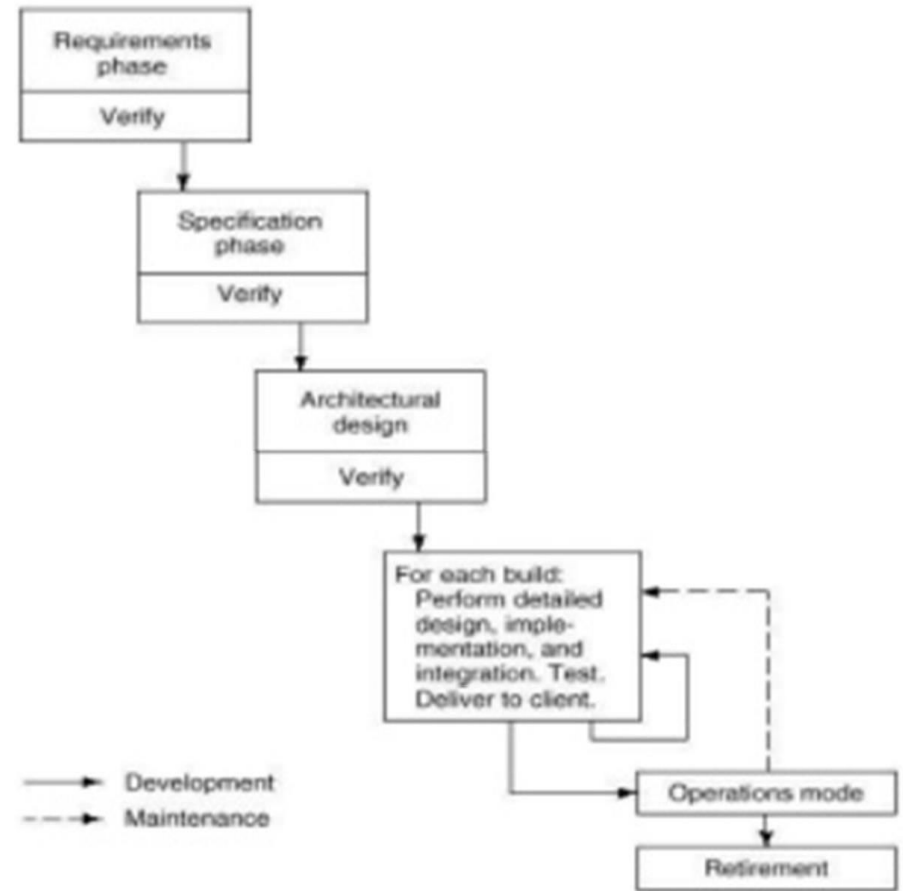


- Recolección de requisitos
- Análisis de Requisitos (análisis)
- Diseño abstracto (descomposición/composición de alto nivel)
- Diseño detallado (comunicación entre módulos, realización interna).
- Codificación
- Verificación de Consistencia y completitud, simulación
- Mejoras correctivas

- La verdadera debilidad es que no hay interacción con el cliente, que solo ve el producto terminado al final del proceso.
 - A menudo, existe una diferencia sustancial entre cómo el cliente imagina un producto y cómo se fabrica el producto.
 - En ese momento, si hay discrepancias (¡y las hay!), Se debe repetir todo el proceso.
- Otro punto crítico es la producción excesiva de documentos:
 - Ninguna fase está completa hasta que el documento para esa fase también haya sido completado y aprobado por el panel de evaluadores (Software Quality Assurance)
- Recordemos, pues, el modelo de cascada por el valor que históricamente ha tenido:
 - Definir y estructurar las fases de un proceso de sw

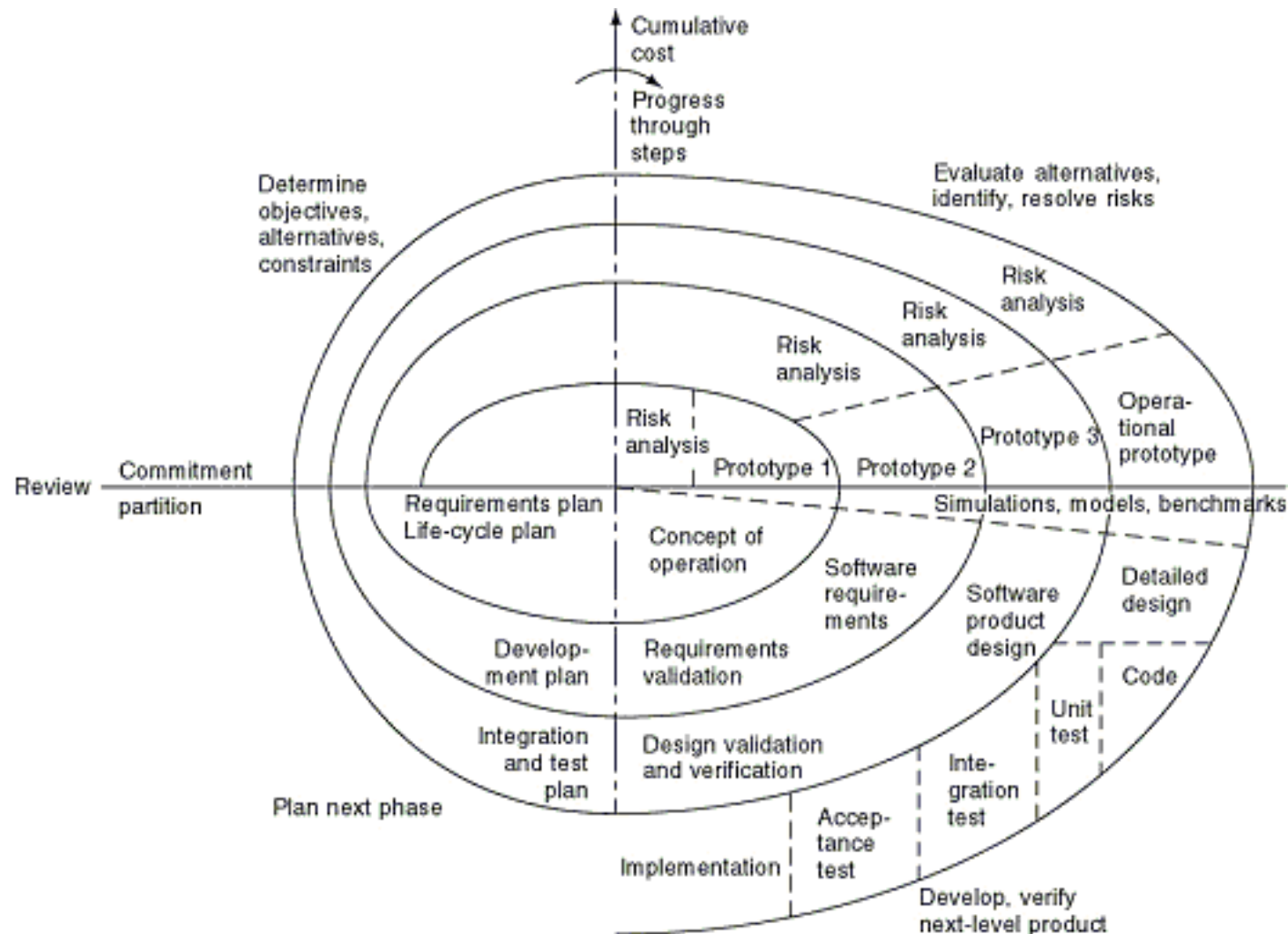
Modelo Incremental

- El sistema se construye agregando algo a lo que ya se ha hecho
 - Los requisitos y el proyecto se definen inicialmente, luego
- El sistema se implementa, integra y prueba con una serie de pasos incrementales



- Se aplica en caso de requisitos estables, sirve a
 - Retrasar la realización de componentes que dependen críticamente de factores externos (tecnologías, hardware experimental, etc.)
 - "Sal" rápidamente con algo
- Si no está bien diseñado, se convierte en un Build-and-Fix

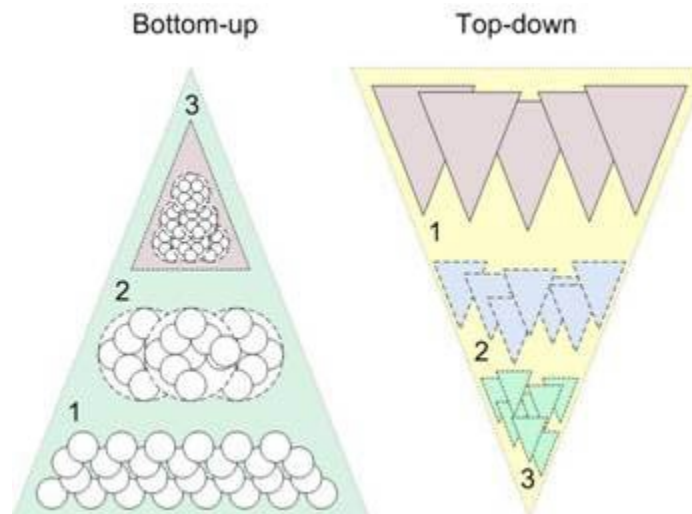
Ciclo de Vida a Espiral (Boehm 1986)



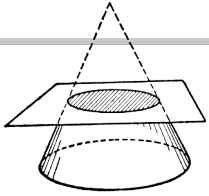
- Propuesto por Boehm en 1988
- Iterativo, cada iteración se organiza en 4 fases:
 - El establecimiento de metas
 - Análisis de riesgo
 - Desarrollo y validación
 - Planificación de nuevos ciclos
- Es un modelo abstracto
 - Debe ser especializado para decir qué hacer en la práctica en cada iteración y en cada una de sus fases

Ciclo de Vida Object Oriented

- Distintas partes del mismo sistema se pueden encontrar al mismo instante en fases distintas del ciclo de vida



- bottom-up (se parte de la clase que modela el sistema)
- top-down (se parte de las clases elementales para después definir aquellas que las usan)



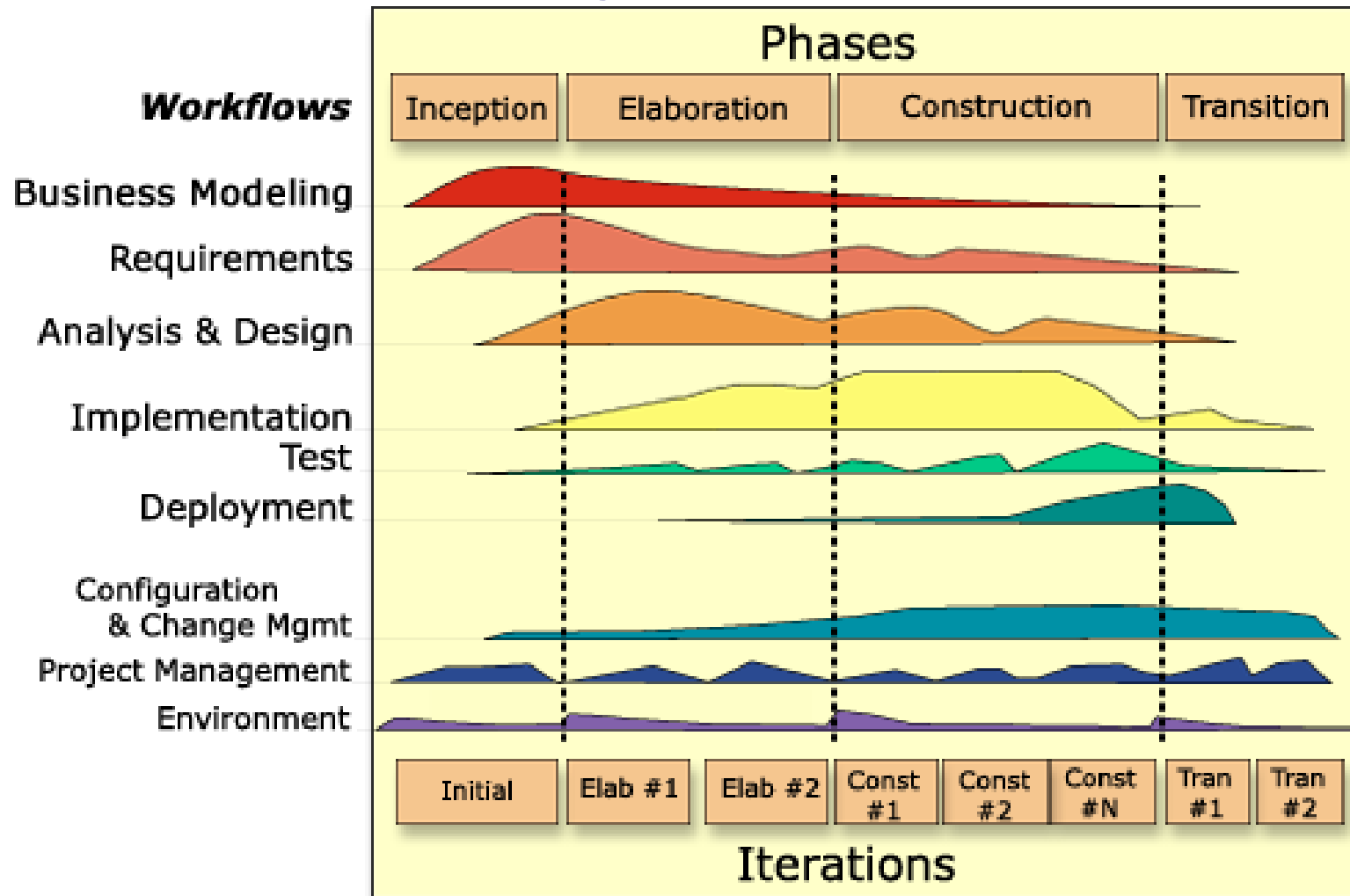
Rational Unified Process (RUP)



Introducción

- RUP (Rational Unified Process) es la metodología tradicional más conocida y mejor documentada.
- Actualmente es un producto de IBM, pero su esencia está descrita en el libro “The Unified Software Development Process” escrito por I. Jacobson, G. Booch y J. Runbaugh en 1999.

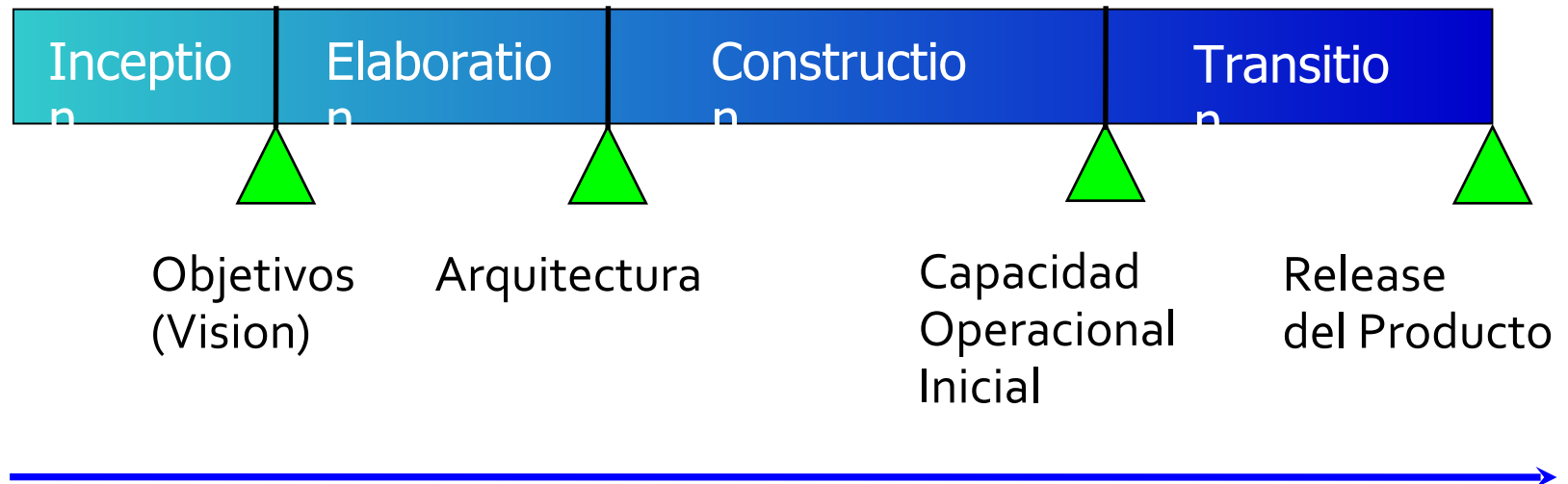
Fases y actividades



Fases del Proceso Unificado

- Inicio (valoraciones generales)
- Elaboración (implementación iterativa de la arquitectura central, identificación de requerimientos adicionales, valoraciones más detalladas)
- Construcción (implementación iterativa de los elementos restantes, preparación para el despliegue)
- Transición (pruebas beta, despliegue)

Fases e Hitos (Milestones)



Elementos en RUP

- Workflows (Disciplinas)

Workflows Primarios

- Business Modeling (Modelado del Negocio)
- Requirements (Requisitos)
- Analysis & Design (Análisis y Diseño)
- Implementation (Implementación)
- Test (Pruebas)
- Deployment (Despliegue)

Workflows de Apoyo

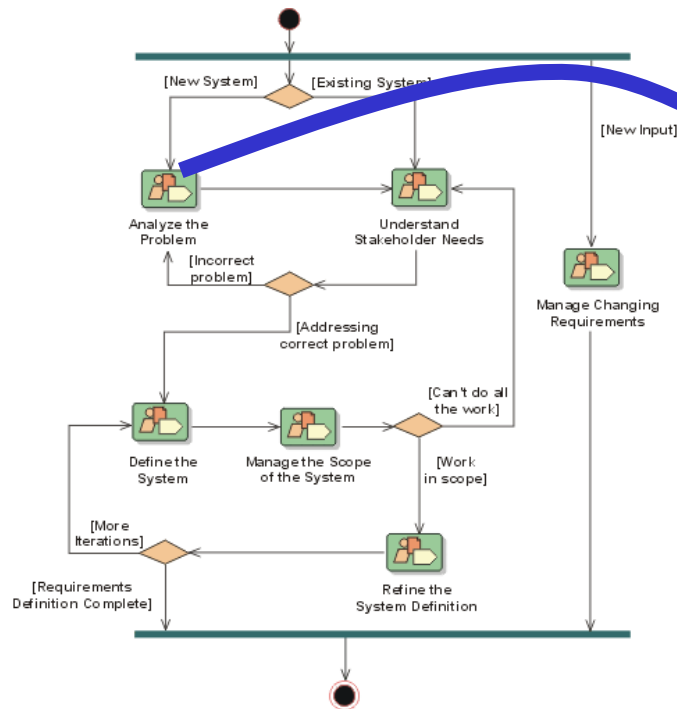
- Environment (Entorno)
- Project Management (Gestión del Proyecto)
- Configuration & Change Management (Gestión de Configuración y Cambios)

... Elementos en RUP

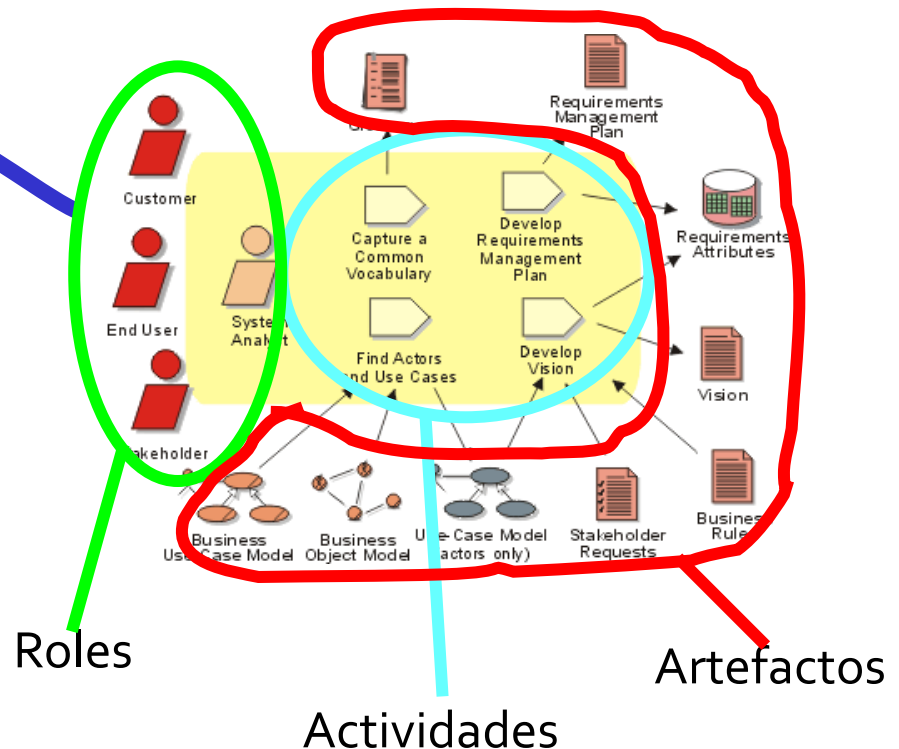
Workflow (Disciplina), Workflow Detail, Roles, Actividades y Artefactos

Ejemplo

Workflow: Requirements



Workflow Detail: Analyse the Problem



Roles

Actividades

Artefactos

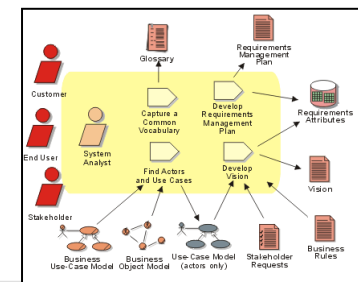
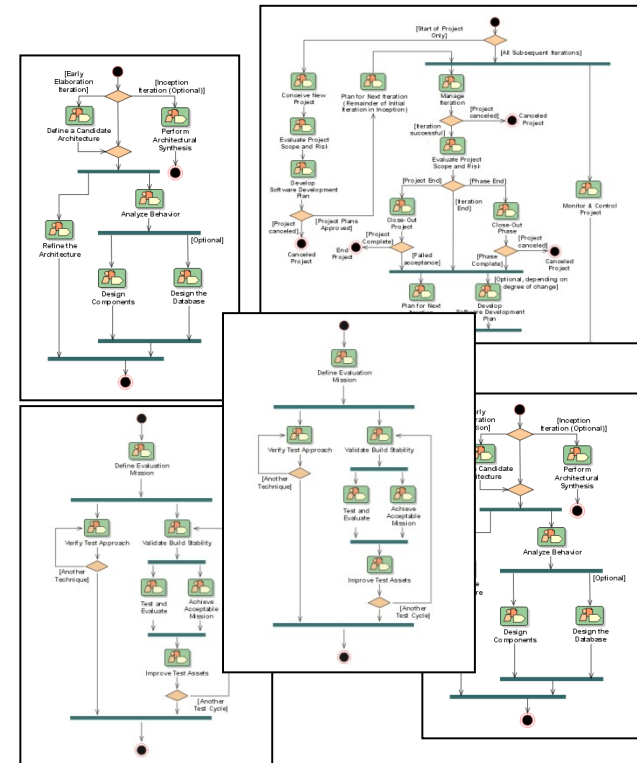
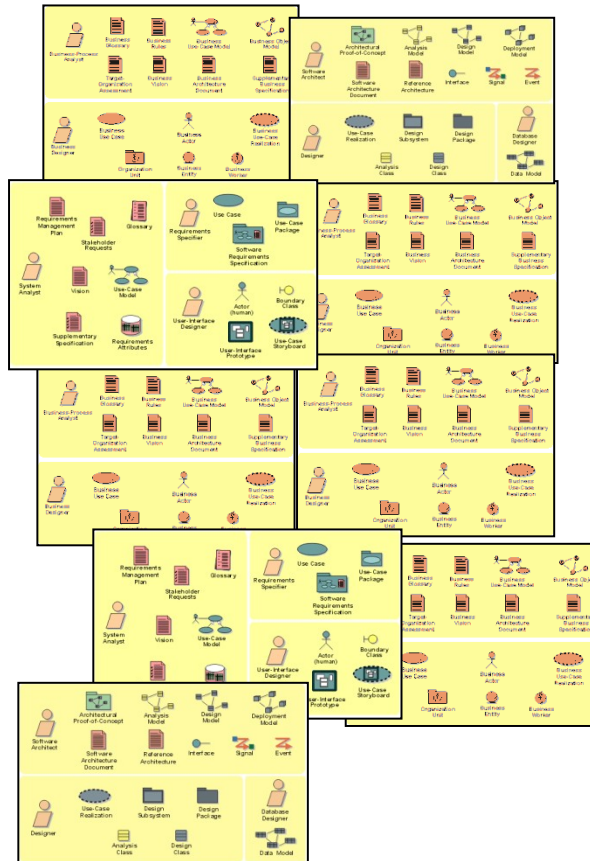
Roles, Artefactos y Actividades de RUP

Analyst

- Business-Process Analyst
- Business Designer
- Business-Model Reviewer
- Requirements Reviewer
- System Analyst
- Use-Case Specifier
- User-Interface Designer

Developer

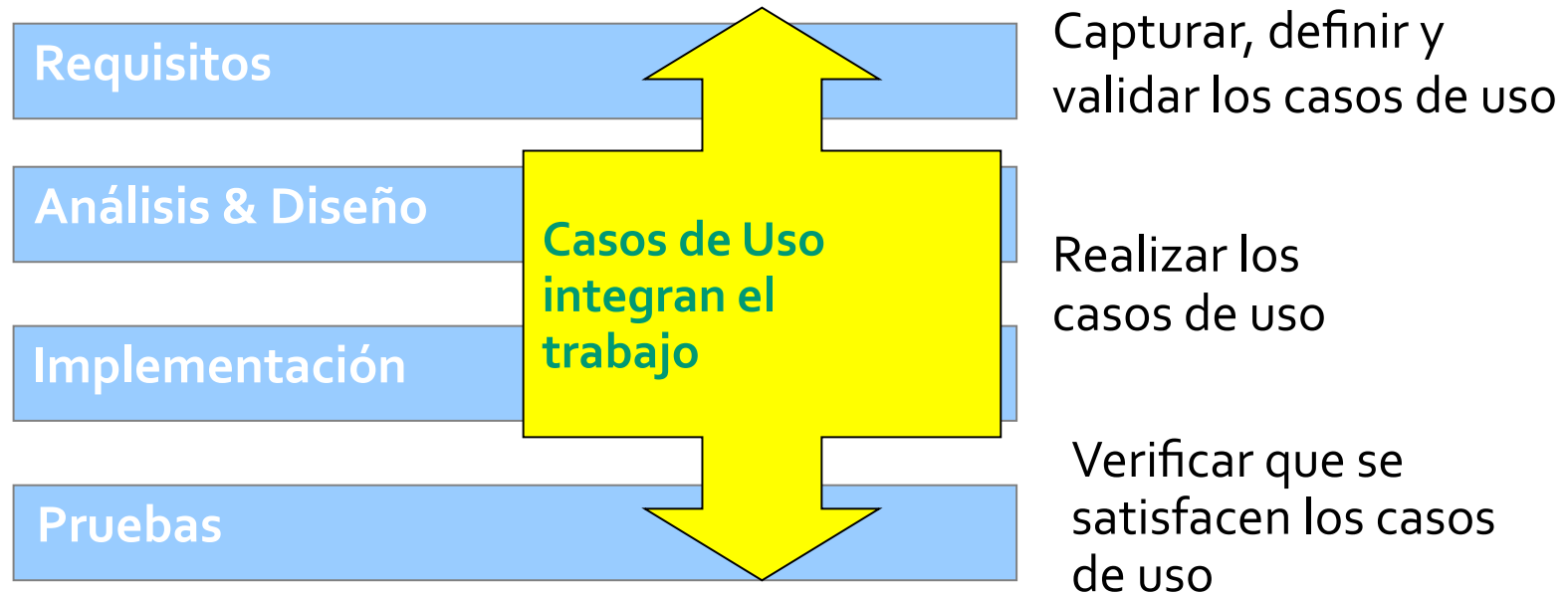
- Architect
- Architecture Reviewer
- Capsule Designer
- Code Reviewer
- Database Designer
- Design Reviewer
- Designer
- Implementer
- Integrator



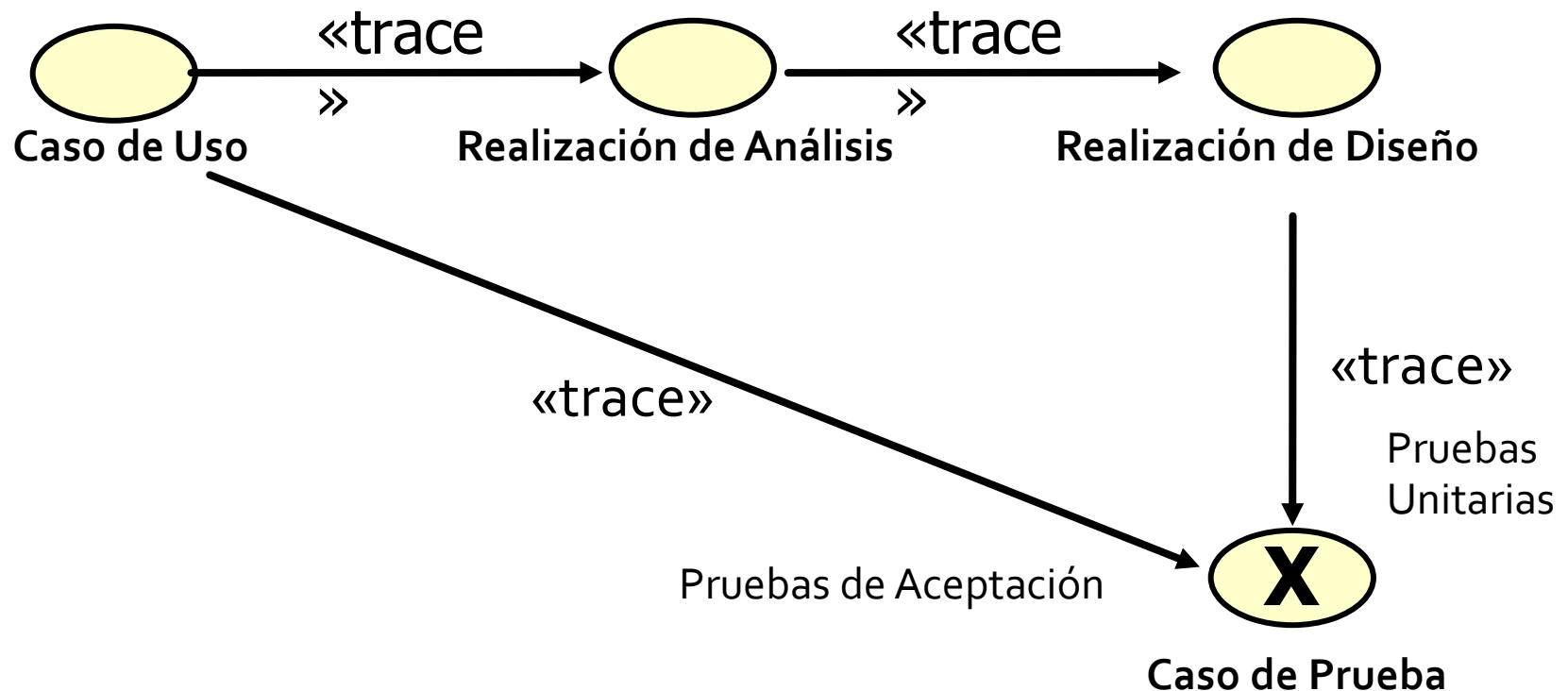
Características Esenciales de RUP

- Proceso Dirigido por los Casos de Uso
- Proceso Iterativo e Incremental
- Proceso Centrado en la Arquitectura

Proceso dirigido por los Casos de Uso



... Proceso dirigido por los Casos de Uso



[The Unified Software Development Process. I. Jacobson, G. Booch and J. Rumbaugh. Addison-Wesley, 1999]

... Proceso dirigido por los Casos de Uso

Estado de aspectos de los Casos de Uso al finalizar cada fase

	Modelo de Negocio Terminado	Casos de Uso Identificados	Casos de Uso Descritos	Casos de Uso Analizados	Casos de Uso Diseñados, Implementados y Probados
Fase de Concepción	50% - 70%	50%	10%	5%	Muy poco, puede que sólo algo relativo a un prototipo para probar conceptos
Fase de Elaboración	Casi el 100%	80% o más	40% - 80%	20% - 40%	Menos del 10%
Fase de Construcción	100%	100%	100%	100%	100%
Fase de Transición					

The Unified Software Development Process. I. Jacobson, G. Booch y J. Rumbaugh. página 358. Addison-Wesley, 1999.

Proceso Iterativo e Incremental

- El ciclo de vida iterativo se basa en la evolución de prototipos ejecutables que se muestran a los usuarios y clientes
- En el ciclo de vida iterativo a cada iteración se reproduce el ciclo de vida en cascada a menor escala
- Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes

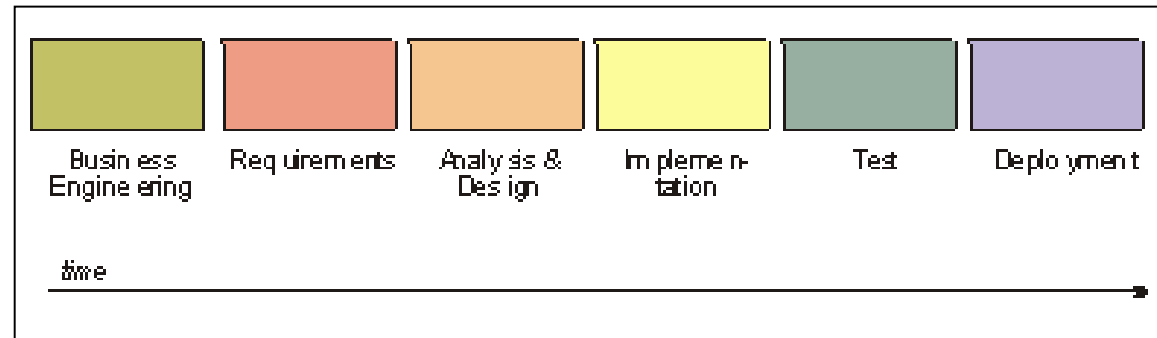
... Proceso Iterativo e Incremental

- Para cada Caso de Uso implementado en la iteración, sus actividades se encadenan en un micro-ciclo (mini-cascada)



... Proceso Iterativo e Incremental

Enfoque
Cascada



Enfoque
Iterativo e
Incremental

