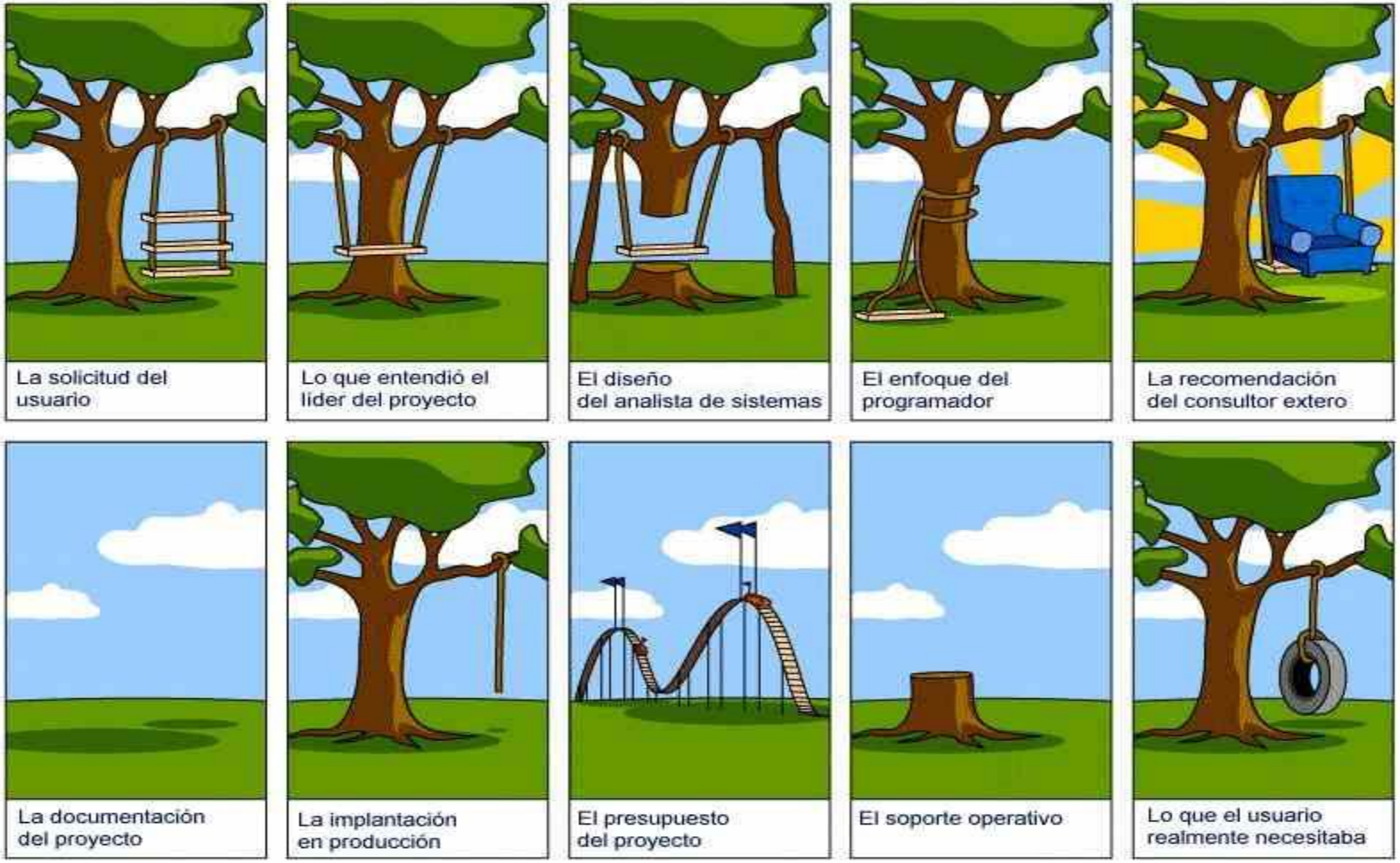
Administración de proyectos de software

MODELOS DE DESARROLLO DE SOFTWARE

Lo que no debería pasar



Tipos de modelos

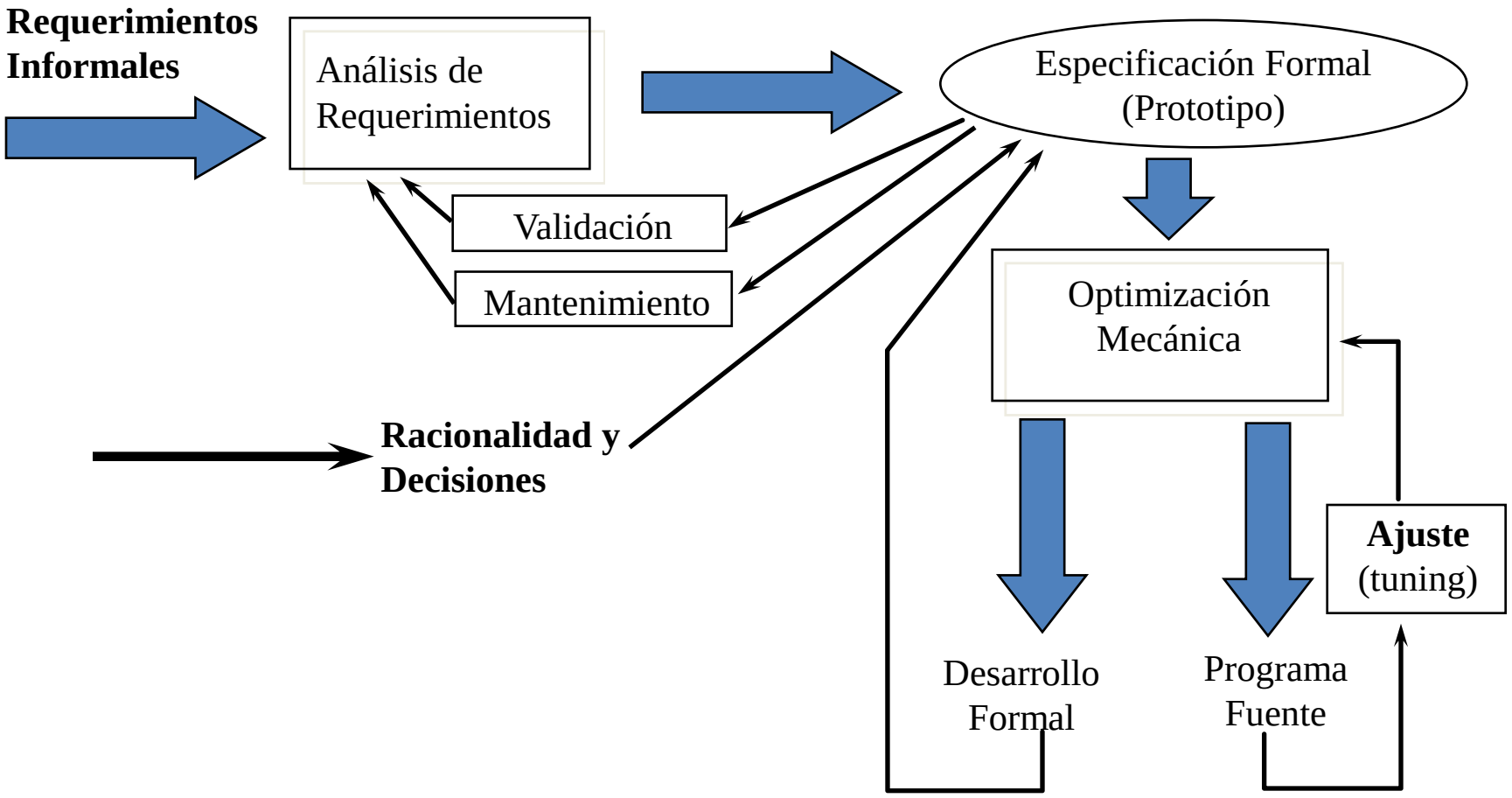
- Modelo lineal secuencial (Cascada)
- Modelo de construcción de prototipos
- Modelo DRA (Desarrollo rápido de aplicaciones.)
- Modelo evolutivos de proceso del software
  - Modelo incremental
  - Modelo espiral
  - Modelo espiral (ganar-ganar)
  - Modelo de desarrollo concurrente (V)
- Modelo de compra de software (Cots)
- Programación Extrema.
- Otras metodologías ágiles
- RUP

Modelo cascada

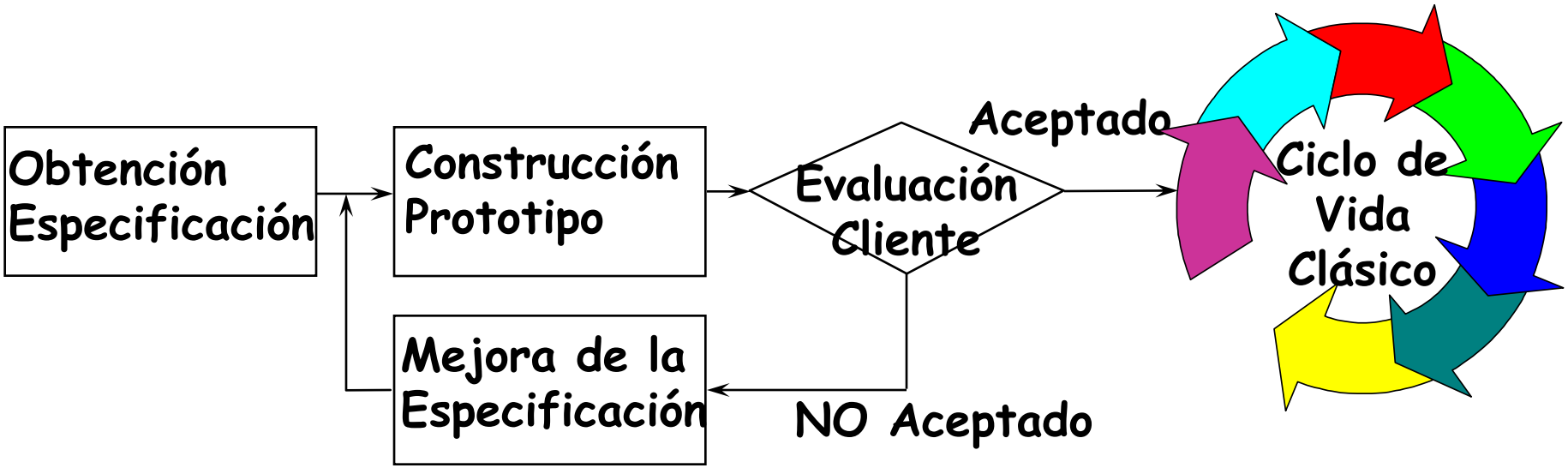
Tipo cascada

- Ventajas:
  - La definición de productos es estable.
  - Cuando se entiende bien las tecnologías.
  - Es la cualidad de tener un buen enfoque de los costos y tiempos.
  - Tener un buen personal técnico
    - Tener un buen equipo
    - Trabaje de forma eficiente.
- Desventajas
  - No es flexible entre su inicio y terminación.
  - Dificultad de definir todos los requerimiento de una sola.
  - Puede producir excesiva documentación.
  - Pobre visibilidad de signos de progreso hasta el final.

Síntesis automática de software



Modelo de construcción de Prototipos



De interfase: Usualmente un modelo sobre papel o ppt con pantallas o listados

De comportamiento: Cubre menús, ciertos procesos, de baja calidad.

Cuidado!!!
El cliente vea lo que parece ser?
Compromisos de implantación rápida

Construcción de prototipos

- Ventajas
  - Diseño, más claro, usualmente de vía visual.
  - Es interesante usarlo cuando:
    - Cambios rápidos de requerimientos
    - No hay compromiso del cliente
    - Cuando el dominio del problema no es óptimo.
  - Presentar firmes progresos en la comprensión
- Desventajas
  - Dificultad en la estimación de tiempos
  - La complejidad del proyecto puede ser desconocida.
  - Si solo se usa este modelo, puede pasar errores graves.

Modelo DRA (Desarrollo de aplicaciones rápidas)

- Es utilizado para periodos de desarrollo corto (60 a 90 días), generalmente para aplicaciones puntuales de sistemas de información. Los pasos a seguir son:

Modelo DRA

- Ventajas
  - Automatiza el proceso manual
  - Se recomienda su uso para pequeños proyectos.
  - Generalmente el usuario es el cliente.
  - Se necesita de personal conocedor de la herramienta.
- Desventajas
  - Es un modelo de inicio y fin, sin retroalimentación.
  - No ofrece el uso de “mejores prácticas”
  - No funciona para proyectos grandes

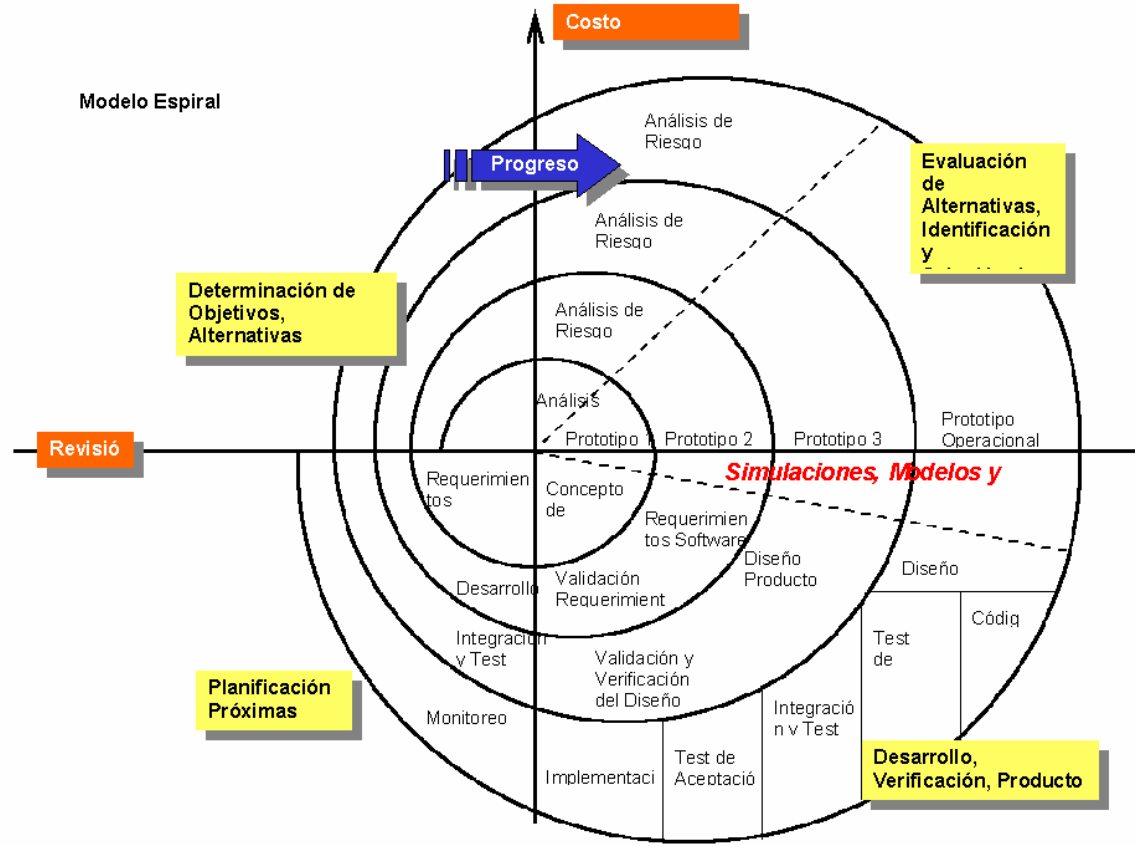
Modelos evolutivos de construcción de software
Modelo Incremental

- Combina elementos del modelo lineal aplicados repetitivamente.

Modelo incremental

- Ventajas
  - Construir un sistema pequeño es menos riesgoso que construir un sistema grande.
  - Ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
  - Reduciendo el tiempo de desarrollo de un sistema por incrementos, decrecen las probabilidades que esos requerimientos de usuario puedan cambiar durante el desarrollo.
  - Si un error importante es detectado, sólo la última iteración necesita ser descartada.
  - Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento.
- Desventajas
  - Serios problemas de integridad, tanto a nivel de código como a nivel de datos, peor cuando parte del proceso es reemplazar a sistemas funcionando.

Modelo Espiral



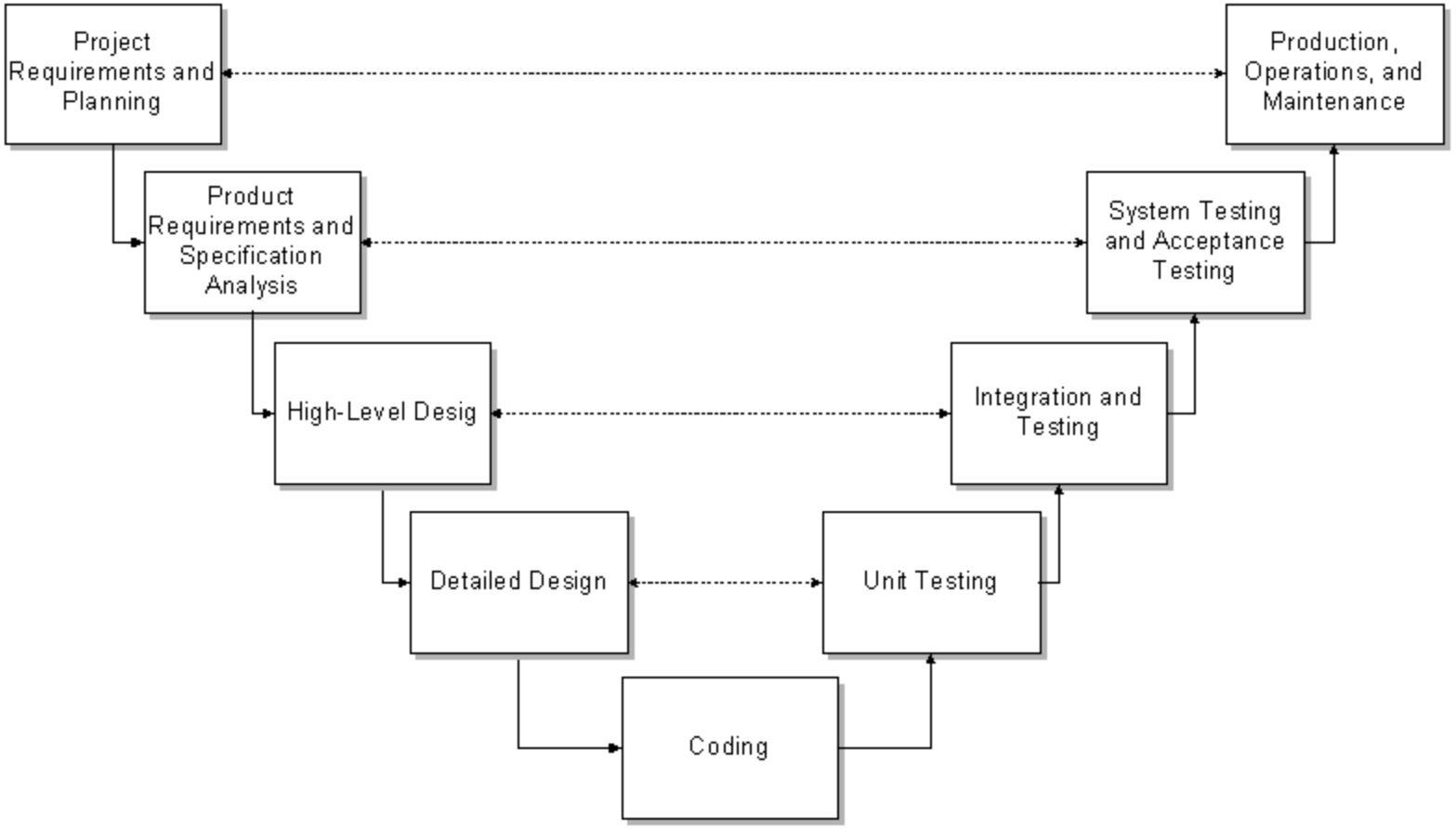
Espiral

- Ventajas
  - Decidir qué problema se quiere resolver antes de viajar a resolverlo.
  - Examinar las múltiples alternativas de acción y elegir la más conveniente.
  - Examinar lo que tienes hecho y lo que te falta por hacer
  - Conocer los riesgos del proyecto en cada vuelta
  - Incremento en el costo y decremento en el riesgo
- Desventajas
  - Más complejo
  - Requiere más administración

Modelos evolutivos de construcción de software
Modelo Espiral ganar-ganar

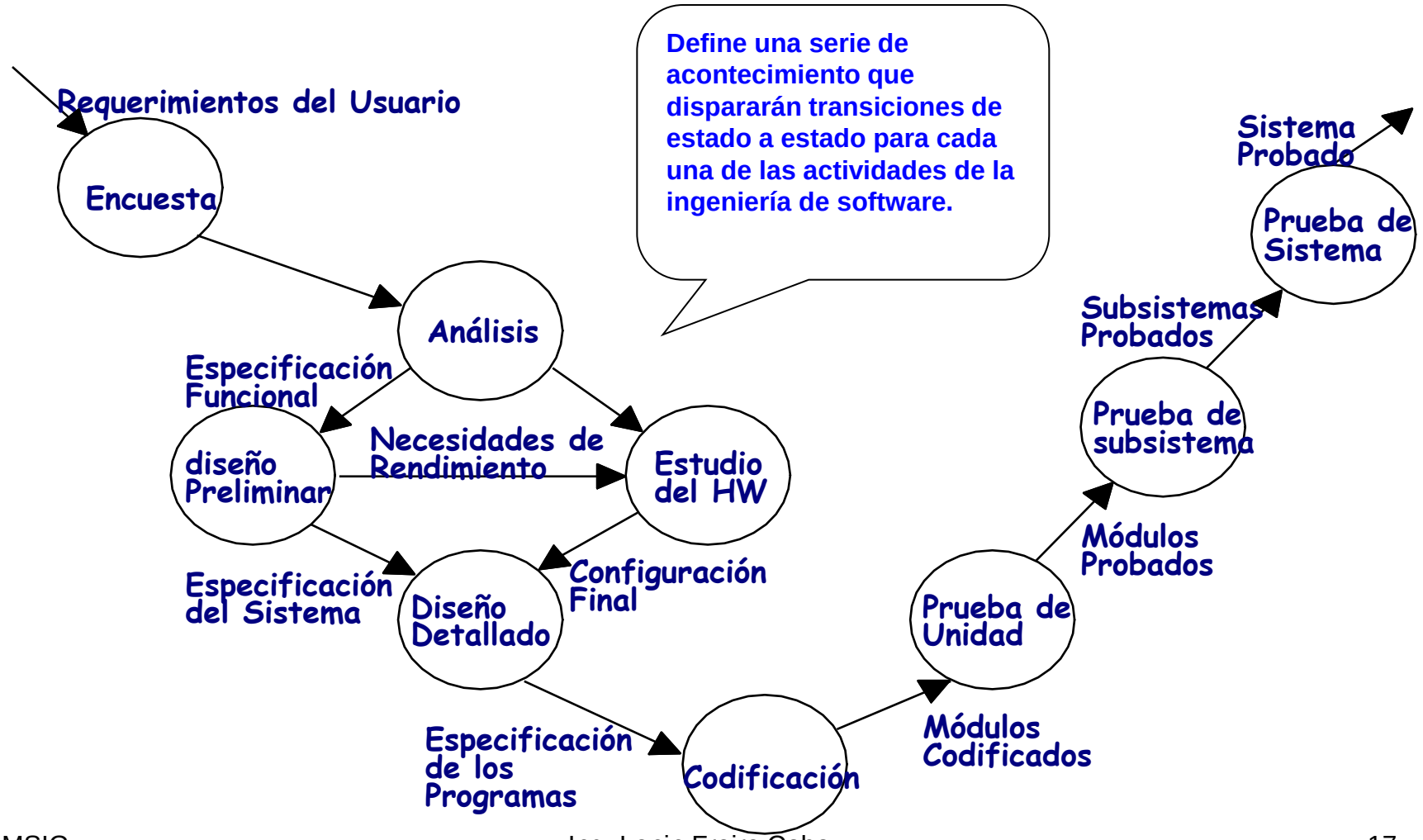
- En cuando el cliente y el desarrollador están en un proceso de negociación permanente sopesando la funcionalidad, rendimiento, tiempo y costo.
- En la práctica, es muy raro que se dé, porque se hacen a través de contratos.

V Process Model



Modelos evolutivos de construcción de software

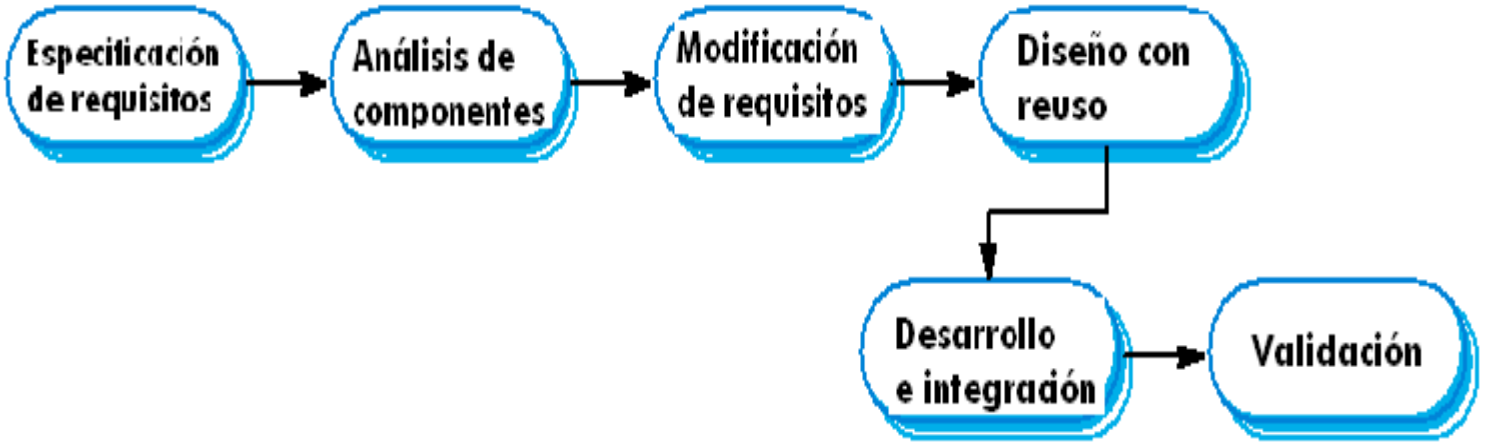
Modelo V



V Process Model

- Ventajas
  - Diseñado para pruebas.
  - Enfatiza verificación y validación en todas las fases.
  - Es una buena opción para sistemas que necesitan alta confiabilidad, como los sistemas de control.
- Desventajas
  - No maneja interacciones
  - Los cambios pueden ser manejados con mucha dificultad

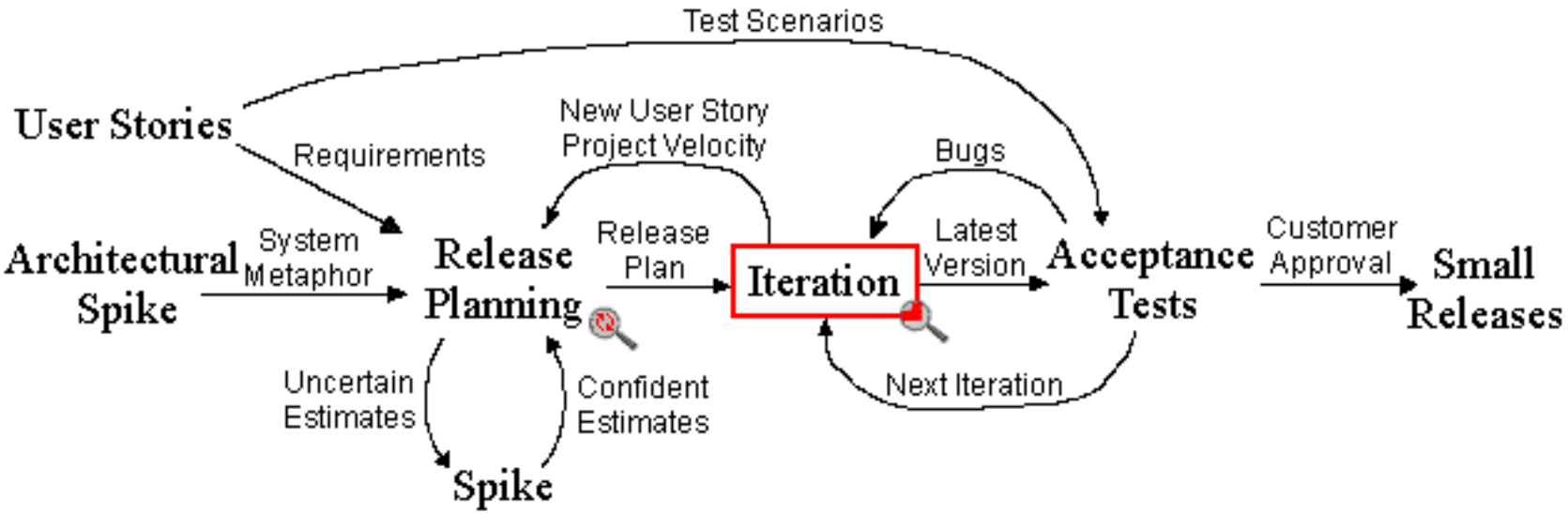
COTS “Commercial-off-the-shelf systems”)



COTS

- Ventajas
  - Se pueden compran componentes de software.
  - Cada vez hay más estandarización.
  - Disponibilidad inmediatamente
  - Potencialmente de bajo costo
- Desventajas
  - No esta hecho a la medida de los requerimientos

eXtreme Programming



XP: eXtreme Programming

- Ventajas
  - No es un producto de Microsoft.
  - Es parte de un movimiento llamado “Agile Development”.
  - Es una metodología ligera.
  - Es una corriente en boga.
  - Utiliza un modelo incrementar/Interactivo.
  - Trabaja con el cliente en sitio.
  - Control de cambios, es incremental.
  - Se programa entre dos personas.
  - Los requerimientos se construyen a través de experiencias.
  - Una buena técnica para un buen desarrollador.
- Desventaja
  - No funciona como industria.
  - No funciona cuando se trabaja con equipos de trabajo grandes.

Otras “Agile” Metodologías

- SCRUM
  - Features 30-day “Sprint” cycles
- Feature Driven Development (FDD)
  - XP with more emphasis on docs and process
- Adaptive Software Development (ASD)
  - Book, site
- Dynamic System Development Method (DSDM)
  - Popular in Europe
- http://www.agile-spain.com

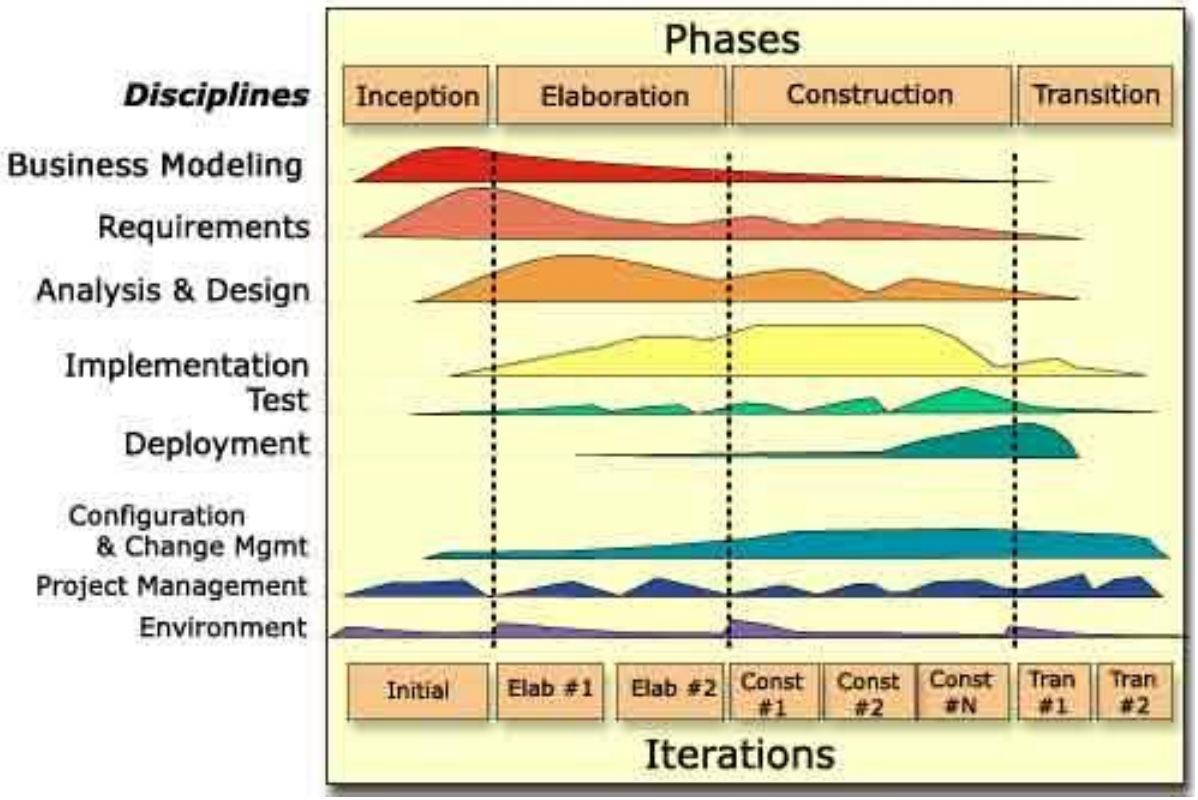
Sobre “Ágiles” Metodologías

- Ventajas
  - Similar al XP, puede reducir procesos
  - Sensible a la retroalimentación con el usuario
  - Receptivo a los cambios
- Desventajas
  - Se requiere cerrar en algún momento los requerimientos.
  - Puede que no haya fin en proyectos largos
  - Frecuentemente requiere calidad de los desarrolladores.
  - Y de una contratación continua.

RUP - Rational Unified Process

- From Rational Corporation
- “Generic” version is the Unified Process
- Commercial
- Extensive tool support (expensive)
- Object-oriented
- Incremental
- Newer

Rational Unified Process



Rational Unified Process

- Ventajas
  - Desarrollo iterativo.
  - Utilizan la metodologíaUML (Unified Modeling Language).
  - Produce artefactos (Componentes).
  - Modelo de software visual.
  - Proceso complejo.
  - Conveniente para sistemas grandes.
- Desventaja
  - No tener personal preparado en la metodología.
  - Deber ir en consistencia con la gestión del proyecto.

Elejir el ciclo de vida

- Dependerá del proyecto?
- Cómo son los requerimientos?
- Cuales son los riesgos?
- Son fijos los tiempos?
- Cual es la experiencia del equipo y del cliente?

Calificación del ciclo de vida

Capacidades del modelo de ciclo de vida	Entrega por etapas	Entrega evolutiva	Diseño por planificación	Diseño por herramientas	Software comercial disponible
Trabaja con poca identificación de los requerimientos	Malo	Medio a excelente	Malo a medio	Medio	Excelente
Trabaja con poca comprensión sobre la arquitectura	Malo	Malo	Malo	Malo a excelente	Malo a excelente
Genera un sistema altamente fiable	Excelente	Medio a excelente	Medio	Malo a excelente	Malo a excelente
Genera un sistema con amplio desarrollo	Excelente	Excelente	Medio a excelente	Malo	N/A
Gestiona riesgos	Medio	Medio	Medio a excelente	Malo a medio	N/A
Estar sometido a una planificación predefinida	Medio	Medio	Excelente	Excelente	Excelente
Requiere poco tiempo de gestión	Medio	Medio	Medio	Medio a excelente	Excelente
Permite modificaciones a medio camino	Malo	Medio a a excelente	Malo a medio	Excelente	Malo
Ofrece a los clientes signos visibles de progreso	Medio	Excelente	Medio	Excelente	N/A
Ofrece a la directiva signos visibles de progreso	Excelente	Excelente	Excelente	Excelente	N/A
Requiere poca sofisticación para los directivos y desarrolladores	Medio	Medio	Malo	Medio	Medio

Calificación del ciclo de vida

Capacidades del modelo de ciclo de vida	Cascada pura	Codificar y corregir	Espiral	Cascadas modificadas	Prototipado evolutivo
Trabaja con poca identificación de los requerimientos	Malo	Malo	Excelente	Medio a excelente	Excelente
Trabaja con poca comprensión sobre la arquitectura	Malo	Malo	Excelente	Medio a excelente	Malo a medio
Genera un sistema altamente fiable	Excelente	Malo	Excelente	Excelente	Medio
Genera un sistema con amplio desarrollo	Excelente	Malo a medio	Excelente	Excelente	Excelente
Gestionar riesgos	Malo	Malo	Excelente	Medio	Medio
Estar sometido a una planificación predefinida	Medio	Malo	Medio	Medio	Malo
Requiere poco tiempo de gestión	Malo	Excelente	Medio	Excelente	Medio
Permite modificaciones a medio camino	Malo	Malo a excelente	Medio	Medio	Excelente
Ofrece a los clientes signos visibles de progreso	Malo	Medio	Excelente	Medio	Excelente
Ofrece a la directiva signos visibles de progreso	Medio	Malo	Excelente	Medio a excelente	Medio
Requiere poca sofisticación para los directivos y desarrolladores	Medio	Excelente	Malo	Malo a medio	Malo

IEEE 1074

- Investigar el proceso de desarrollo de software IEEE 1074