



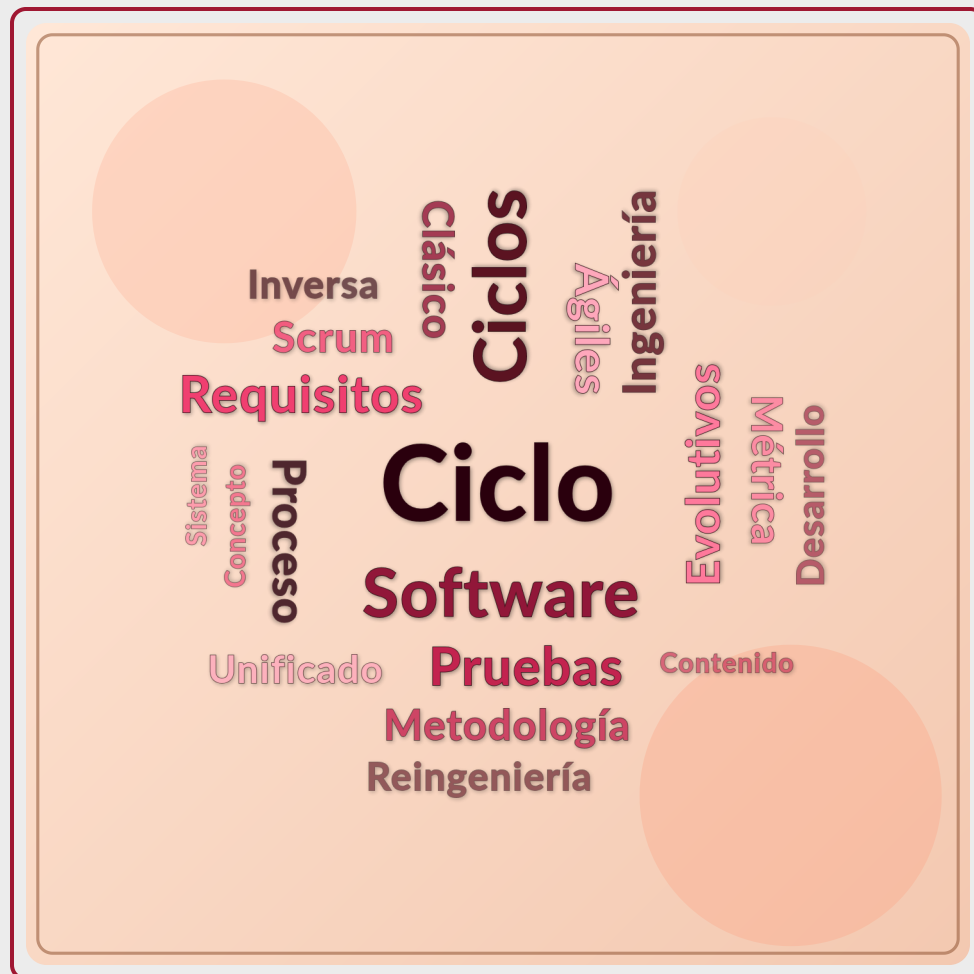
El Ciclo de Vida del Software

Universidad de Sevilla

**Departamento de Lenguajes y Sistemas
Informáticos**

UNIVERSIDAD DE SEVILLA

Objetivos



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

Pruebas en el ciclo de vida

Ingeniería inversa

Reingeniería del software



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

¿Qué es el ciclo de vida del software?

Un **marco** de referencia que contiene **procesos, actividades y tareas** involucradas en el **desarrollo**, la **operación** y el **mantenimiento** de un producto software, abarcando la vida del sistema desde su definición **hasta su retirada** (ISO 12207).

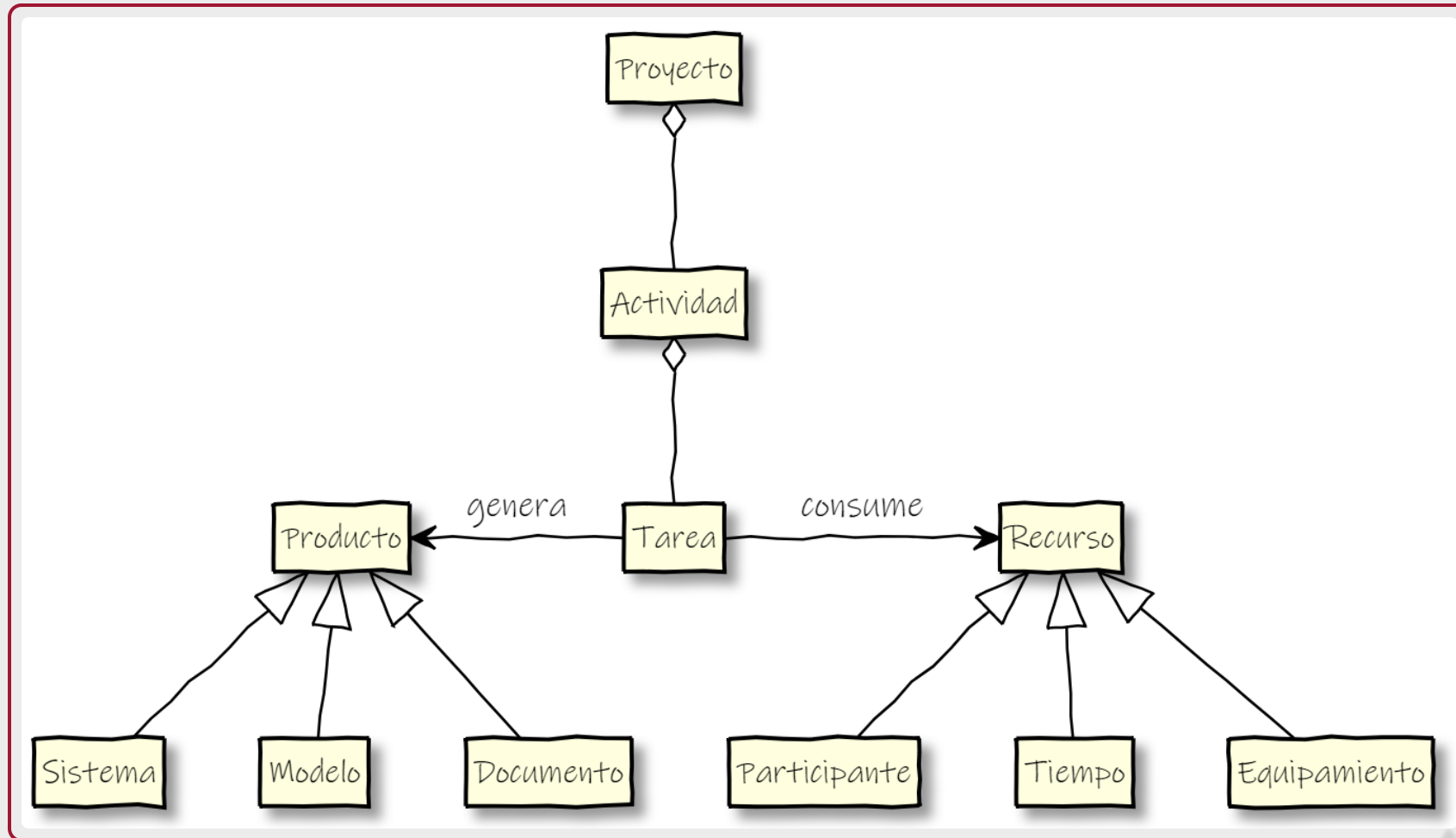
El ciclo de vida de un proyecto **especifica** el enfoque general del desarrollo, indicando procesos, actividades y tareas que se van a realizar y **en qué orden**.

También define qué **productos** se van a generar, cuáles se entregan al cliente y en qué secuencia.

El software tiene **etapas** desde su nacimiento hasta su retirada.



¿Qué es el ciclo de vida del software?



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

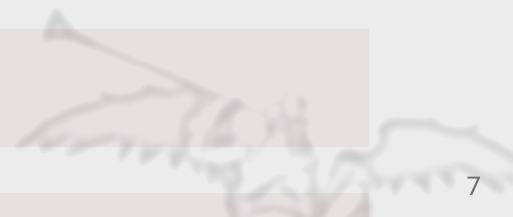
Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

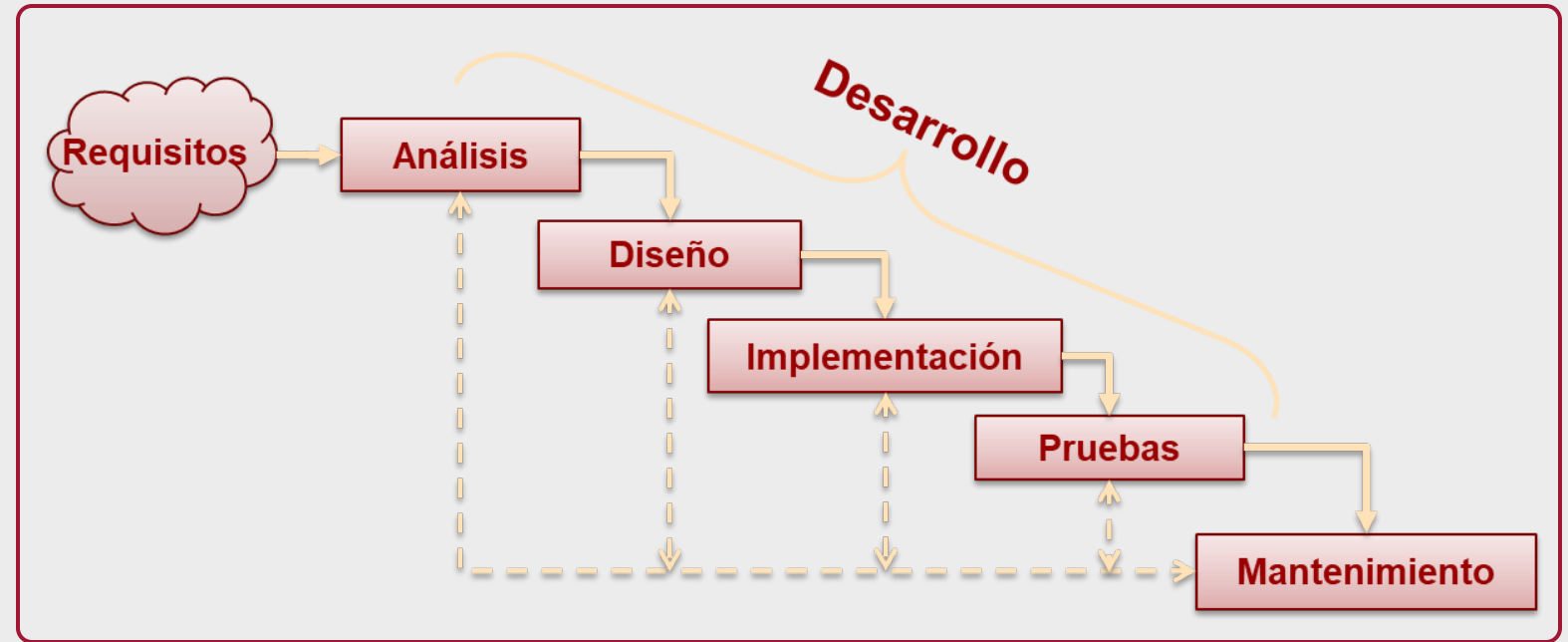
Proceso Unificado

Métrica 3



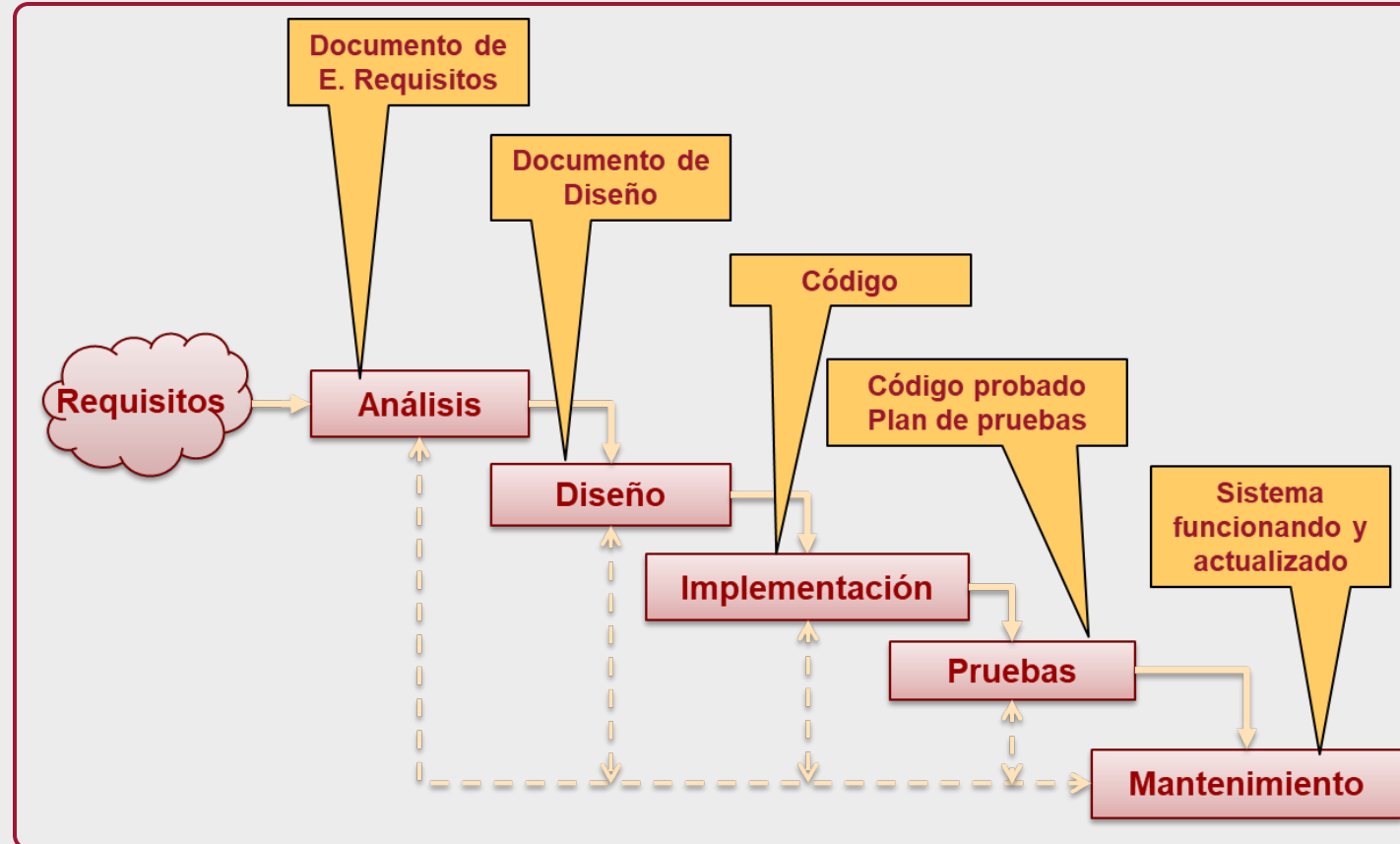
Ciclo de vida clásico (en cascada)

- Cada fase comienza cuando termina la anterior.
- Asume que se conocen todos los requisitos.
- Se tarda mucho en disponer del software.
- Es mejor que no seguir ningún ciclo de vida.
- Es el más fácil de planificar.



Ciclo de vida clásico (en cascada)

Cada fase finaliza con entregables.



Ciclo de vida clásico (en cascada)

Ventajas

- Tiene un lugar destacado en la Ingeniería del Software.
- Proporciona una plantilla para adecuar los métodos.
- Es muy utilizado.
- Es mejor que desarrollar sin guías.

Inconvenientes

- La progresión lineal no refleja cómo se desarrolla realmente el software.
- Es un modelo rígido.
- Exige requisitos completos al principio, con poca tolerancia a la incertidumbre inicial.
- Recorre el ciclo completo antes de disponer de valor operativo.
- Es monolítico y retrasa la detección de errores o incongruencias en requisitos.
- Impone una estructura de gestión que no siempre encaja con la resolución iterativa de problemas.

Construir un puente

Fase de requisitos

Antes de mover una piedra, se recogen requisitos: longitud, peso máximo, materiales, presupuesto y plazo.

Fase de diseño

Se elaboran planos detallados (cálculos estructurales, resistencia de materiales y medidas exactas).

Fase de construcción

Se construye siguiendo planos; ya hay resultado parcial, pero aún no está listo para uso.

Fase de pruebas

Se valida resistencia, seguridad y tráfico. Corregir fallos en esta etapa es costoso.

Entrega y mantenimiento

Se abre al público y se realizan reparaciones o mantenimiento, no rediseños mayores.

Idea clave: En obra civil el cascada encaja bien; en software, los cambios de requisitos durante el proyecto hacen ese enfoque mucho más costoso.

Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

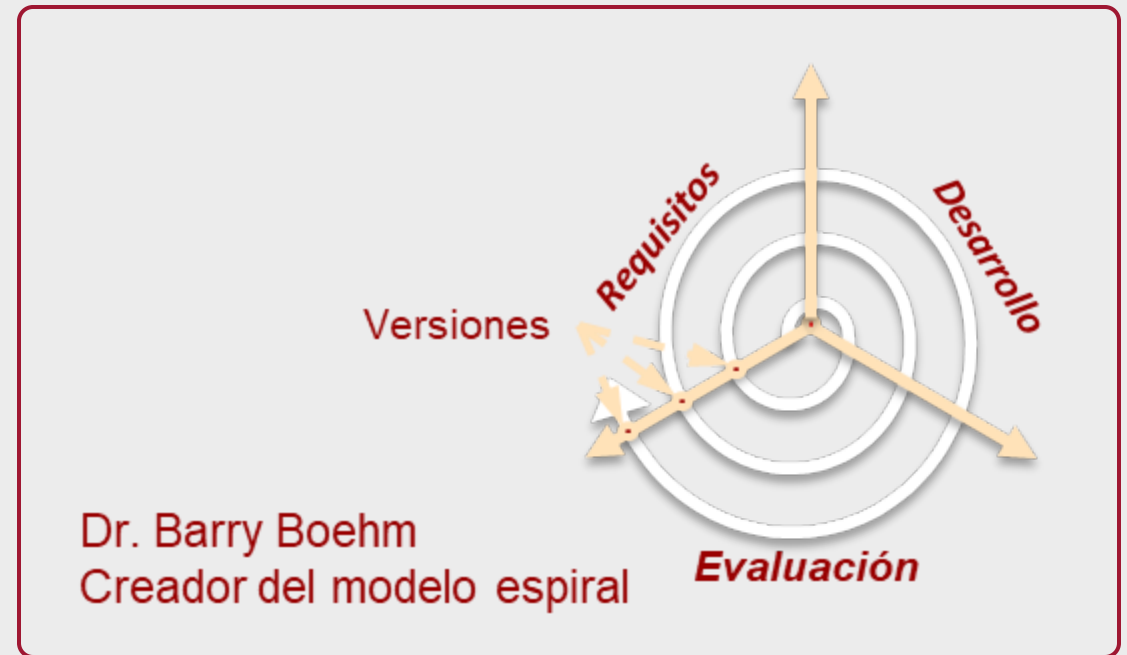
Metodología Scrum

Proceso Unificado

Métrica 3

Ciclos evolutivos

- Cuanto mayor es un proyecto, menor es la probabilidad de éxito (informes CHAOS).
- Obtener todos los requisitos al comienzo suele ser inviable.
- Se trabaja en ciclos de requisitos-desarrollo-evaluación.
- La evaluación permite evolucionar hacia la siguiente versión.
- Se entrega valor antes de finalizar el producto completo.



Ciclos de vida evolutivos

Incremental

- Desarrollo en módulos o partes.
- Cada entrega añade funcionalidad.
- **Ejemplo:** banca (saldo / transferencias / pagos).

Iterativo

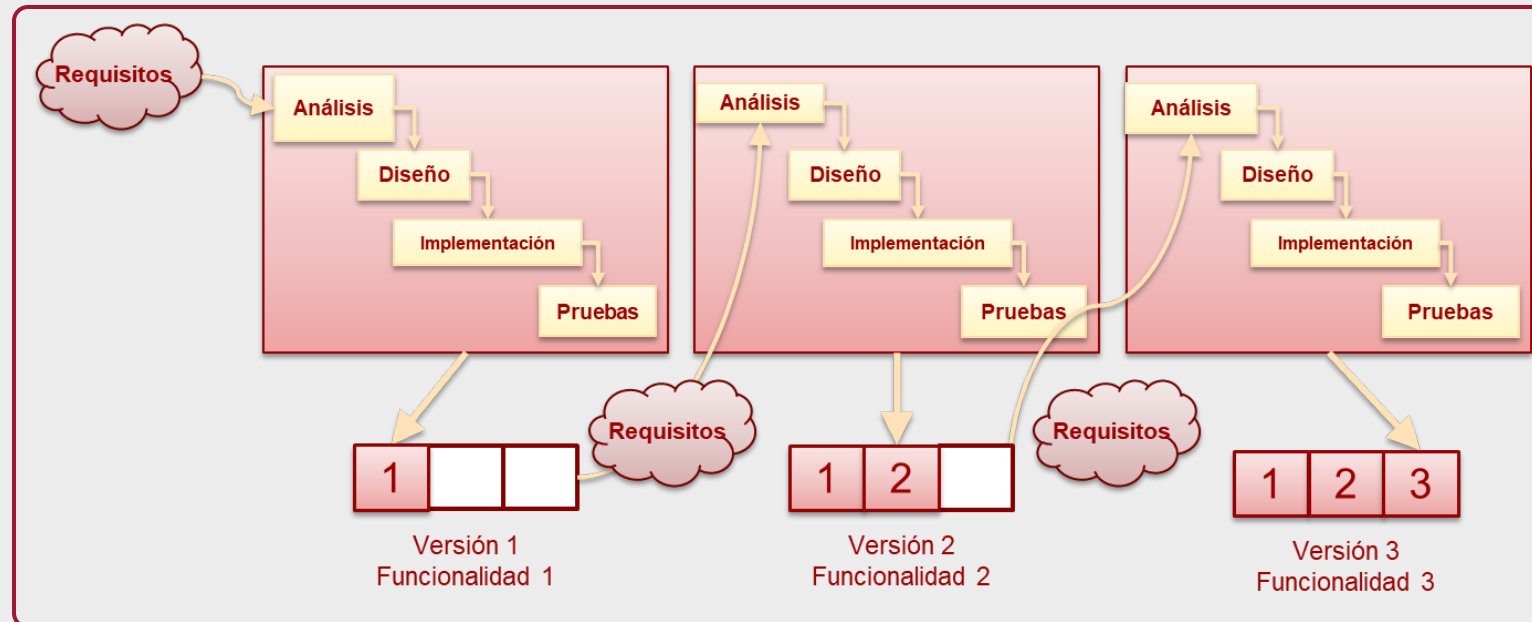
- Versiones sucesivas del producto.
- Cada versión se refina hasta el sistema final.
- **Ejemplo:** videojuego que mejora jugabilidad, gráficos y modos.

Prototipos

- Versiones rápidas para validar requisitos.
- Pueden ser desechables o evolutivos.
- **Ejemplo:** prototipo de interfaz para validar experiencia.

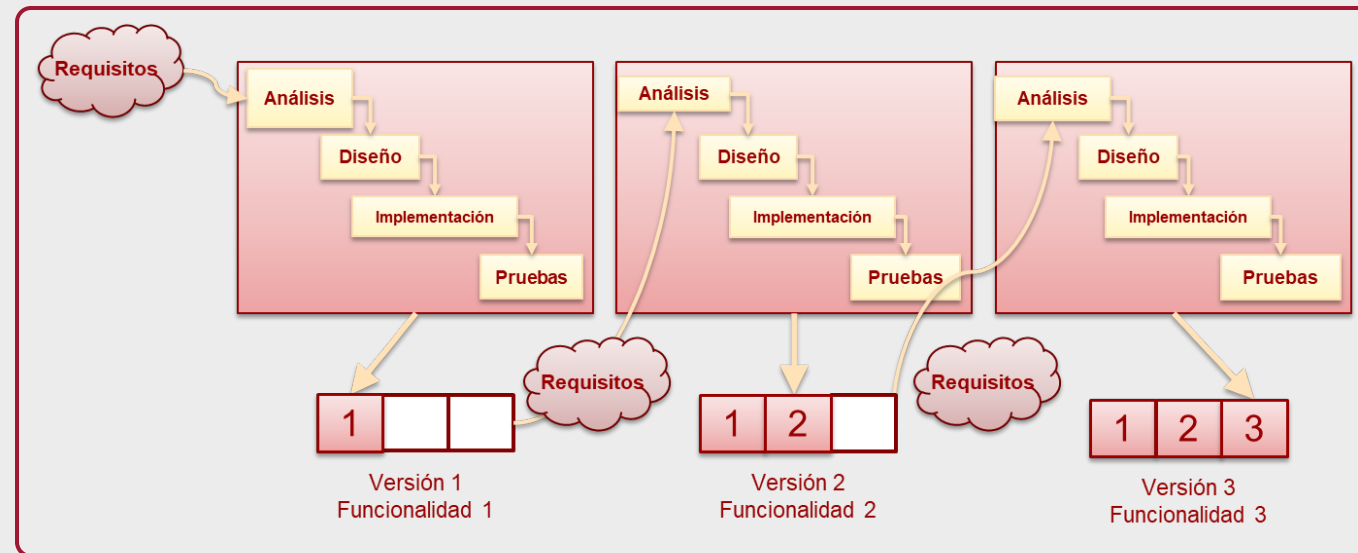
Ciclo de vida incremental

- Repetición de varios ciclos de vida en cascada.
- Se aplica con frecuencia en desarrollos de gran tamaño.
- Al final de cada ciclo se entrega una versión parcial incrementada con nueva funcionalidad.



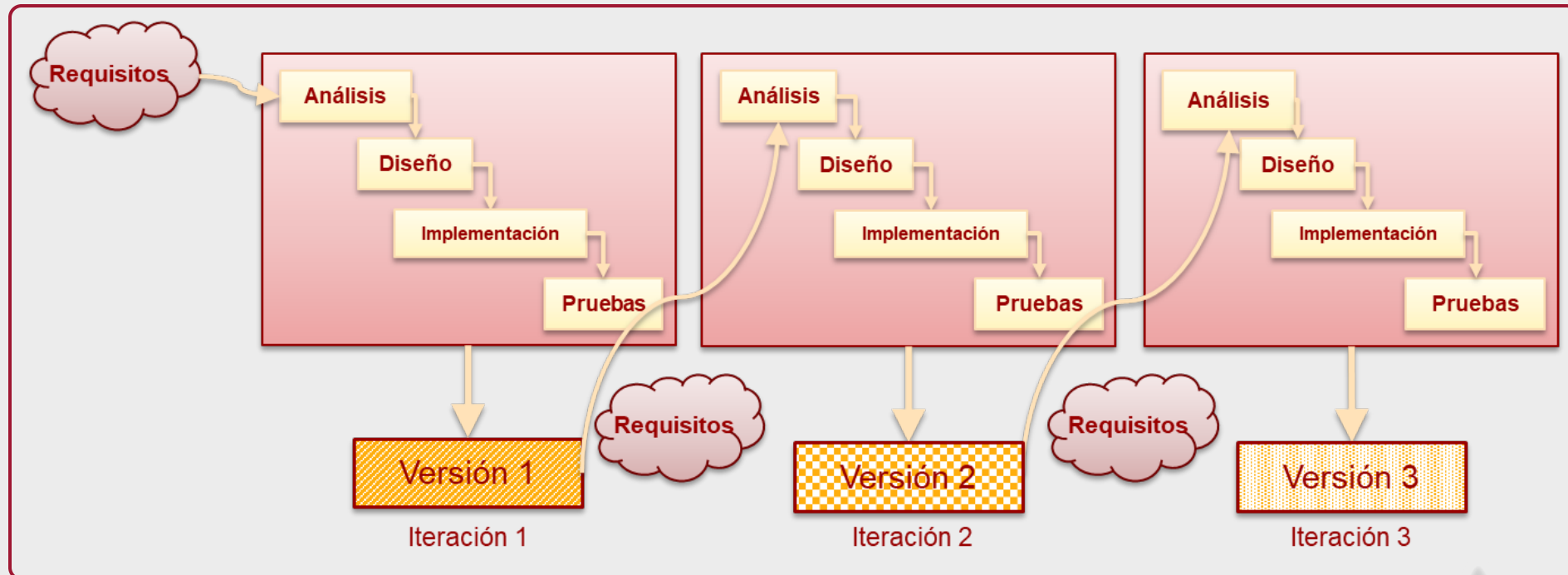
Ciclo de vida incremental

- Los ciclos se repiten hasta obtener un producto completo.
- Aunque no esté completamente terminado, el usuario dispone antes del software.
- Esto permite incorporar sugerencias y nuevos requisitos en incrementos posteriores.



Ciclo de vida iterativo

- Las primeras versiones pueden ser prototipos que luego se desechan.
- Los ciclos se repiten hasta obtener un producto satisfactorio.
- Los usuarios evalúan cada iteración y proponen mejoras.

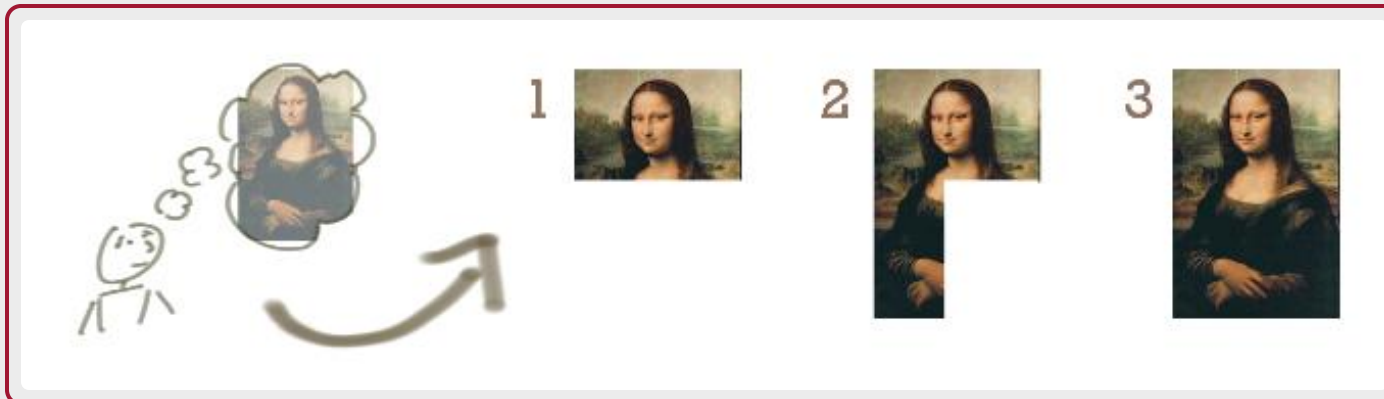


Incremental frente a iterativo

Incremental

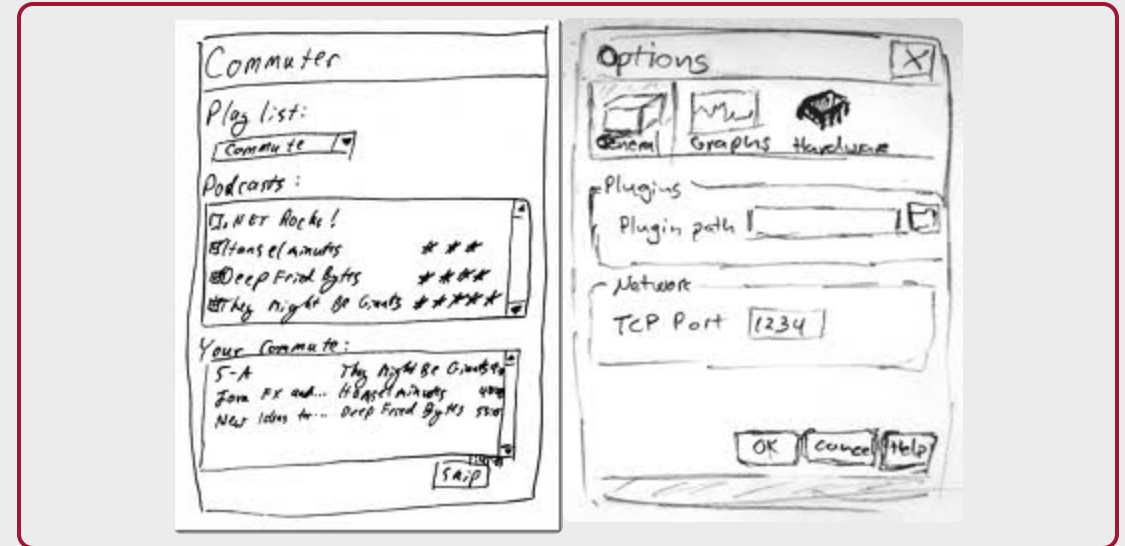


Iterativo



CV basados en prototipos

- No son exclusivos del ciclo iterativo.
- Pueden usarse para obtener y validar requisitos en cualquier ciclo de vida.
- Lo más habitual son prototipos de interfaz: reutilizables (ejecutables) o desechables (mockups).



CV basados en prototipos

Siempre hay que **evaluar** si el **esfuerzo** de construir el prototipo compensa. Es fundamental la **implicación** de usuarios.

Existen distintos enfoques:

- **Desechable**: el prototipo se descarta.
- **Evolutivo**: el prototipo evoluciona hasta sistema final.
- **Mixto**: desechables para requisitos conocidos y evolutivos para requisitos inciertos.

También se usan ***prototipos funcionales*** para evaluar algoritmos antes de tomar decisiones de diseño.



CV basados en prototipos

Inconvenientes:

- El sistema puede deteriorarse si se arrastra el modelo primitivo.
- Se tiende a refinar el prototipo en lugar de rediseñar con base sólida.
- El cliente puede intentar adoptar el prototipo como sistema final.
- Puede haber relajación en las prácticas de desarrollo.
- No siempre reduce el tiempo entre requisitos y entrega.
- Al usuario puede desagradarle que se deseche código.



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

CV en los métodos ágiles

Son CV evolutivos con **iteraciones cortas** (2 semanas a 2 meses)

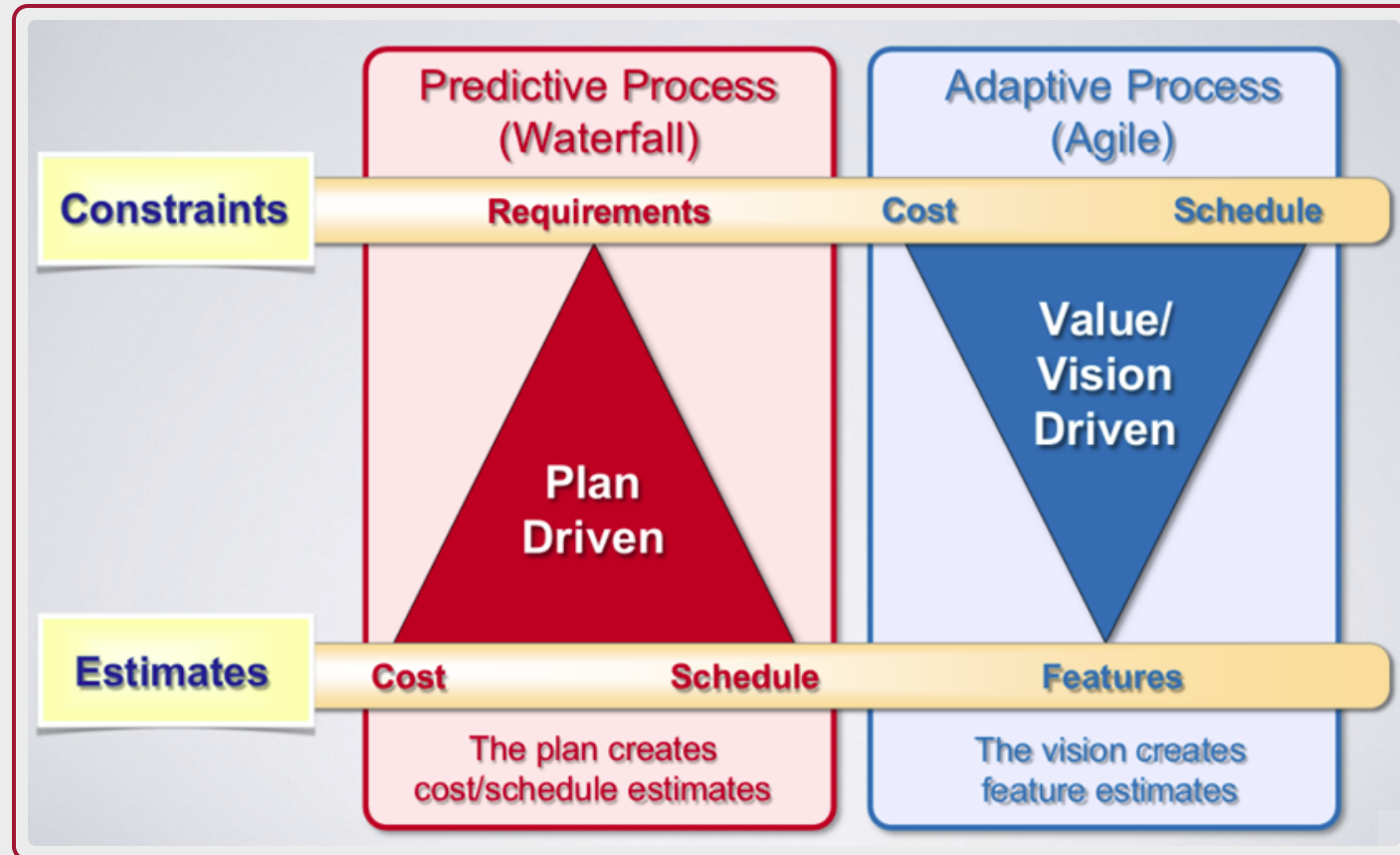
Buscan favorecer la **comunicación** con clientes y usuarios

En cada iteración se incorporan **nuevas peticiones** (requisitos)



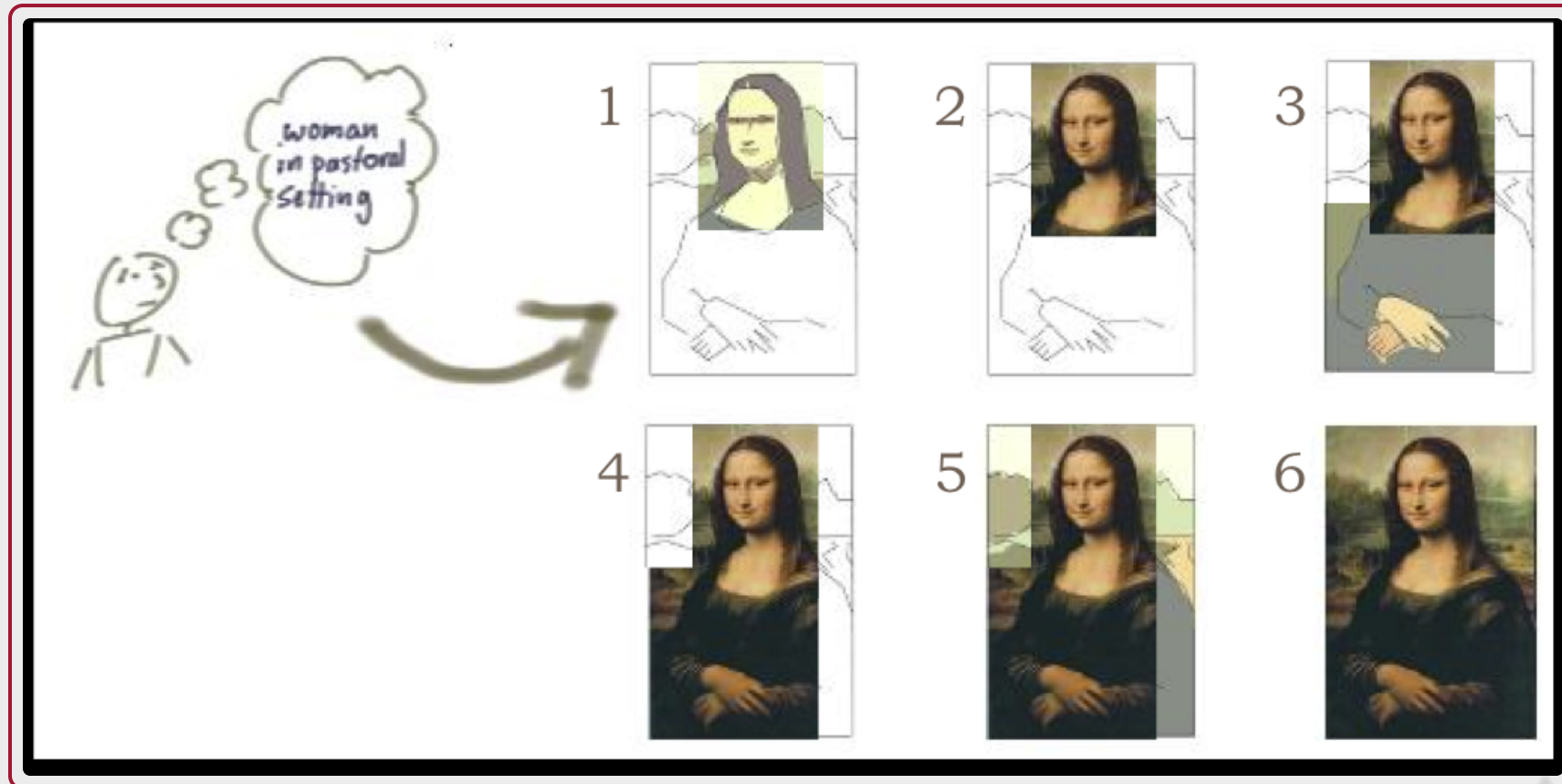
CV en los métodos ágiles

Desarrollo ágil frente a tradicional



CV en los métodos ágiles

Iterativo e incremental a la vez



El manifiesto ágil de 2001...

AGILE

Manifiesto for Agile Software Dev.

- INDIVIDUALS AND INTERACTIONS OVER PROCESSES AND TOOLS
- WORKING SOFTWARE OVER COMPREHENSIVE DOCUMENTATION
- CUSTOMER COLLABORATION OVER CONTRACT NEGOTIATION
- RESPONDING TO CHANGE OVER FOLLOWING A PLAN

El manifiesto ágil de 2001...

Individuos e *interacciones* sobre procesos y herramientas.

Software que funcione sobre documentación detallada.

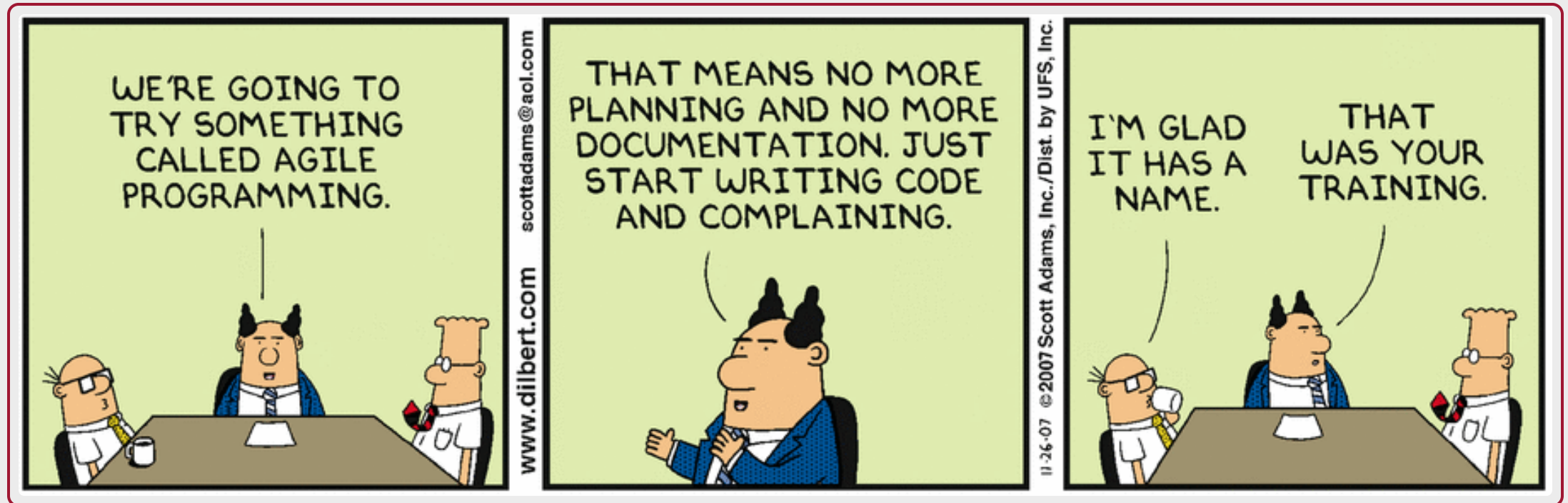
Colaboración con el cliente sobre negociación de contratos.

Respuesta al cambio sobre seguimiento de un plan.



El manifiesto ágil de 2001...

Los métodos ágiles no consisten en escribir código desde el primer momento sin planificar ni documentar.



Fuente: <https://www.agitma.nl/dilbert-saves-the-agile-day/>

El manifiesto ágil de 2001...

Individuos e interacciones sobre procesos y herramientas

- La gente es el principal factor de éxito de un proyecto software.
- Es más importante construir un buen equipo que construir primero el entorno.
- Un error frecuente es montar el entorno y esperar adaptación automática.
- Es preferible formar el equipo y que este configure su entorno según necesidades.

El manifiesto ágil de 2001...

Software que funcione sobre documentación detallada

- Regla práctica: *"no producir documentos salvo que sean necesarios para tomar decisiones importantes"*.
- Cuando se documenten decisiones, hacerlo en artefactos breves y centrados en lo esencial.



El manifiesto ágil de 2001...

Colaboración con el cliente frente a negociación de contratos

- Se promueve una interacción continua entre cliente y equipo de desarrollo.
- Esta colaboración marca la marcha del proyecto y ayuda a asegurar su éxito.



El manifiesto ágil de 2001...

Respuesta al cambio sobre seguimiento de un plan

- La capacidad de responder a cambios en requisitos, tecnología o equipo es clave, determina también éxito o fracaso del proyecto.
- La planificación debe ser flexible y abierta, no rígida.

Desarrollo ágil vs. desarrollo tradicional

Ágil

- Basadas en heurísticas provenientes de prácticas de producción de código
- Especialmente preparados para cambios durante el proyecto
- Impuesta internamente
- Proceso menos controlado, con pocos principios
- El contrato es flexible
- El cliente es parte del equipo de desarrollo
- Equipos pequeños y/o en contacto físico
- Pocos artefactos
- Pocos roles
- Menor énfasis en la arquitectura

Tradicional

- Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
- Presenta cierta resistencia al cambio
- Impuesta externamente
- Proceso muy controlado, con numerosas políticas y normas
- Contrato prefijado
- El cliente interactúa formalmente en reuniones
- Grupos grandes y/o distribuidos
- Numerosos artefactos
- Numerosos roles
- Arquitectura y modelos fundamentales

Técnicas de apoyo para métodos ágiles

Refactorización (refactoring)

- Mejoras sobre el código fuente sin cambiar su funcionalidad.

Pruebas automáticas

- Pruebas programadas en lugar de realizadas a mano.

Integración continua

- Automatización de compilación y ejecución de pruebas automáticas.

Gestión de configuración

- Especialmente orientada a apoyar colaboración e integración continua.

Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

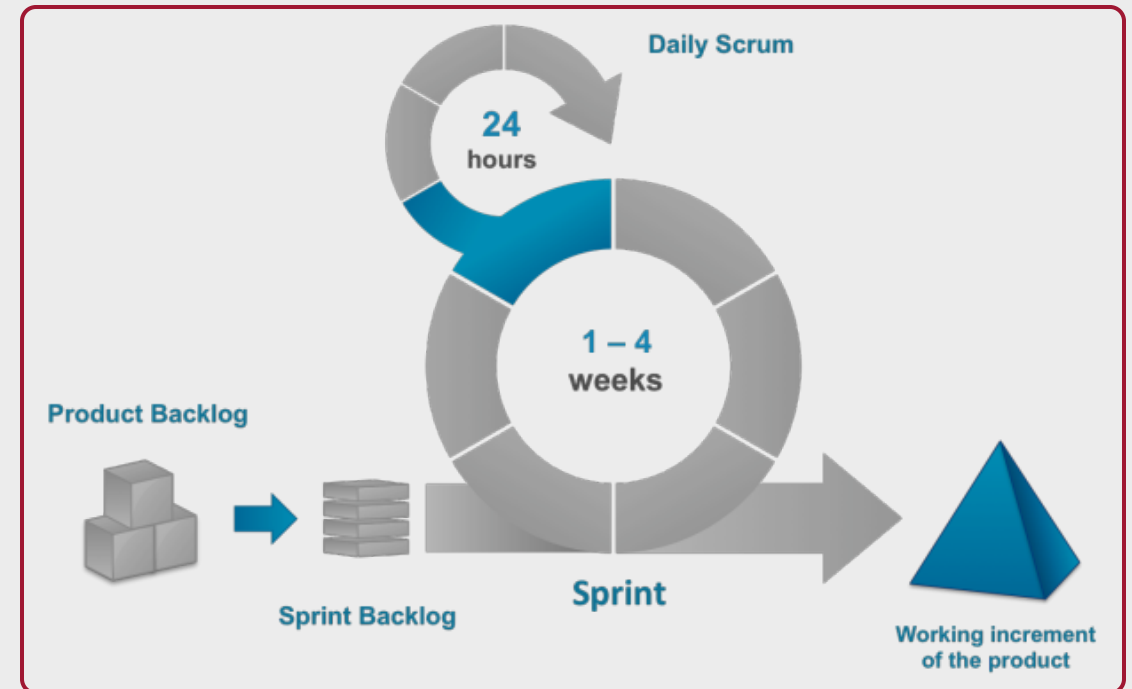
Metodología Scrum

Proceso Unificado

Métrica 3

Metodología Scrum

- Metodología ágil de uso muy extendido.
- Se basa en iteraciones de 1 a 4 semanas (sprints).
- Durante un sprint:
 - Se produce código potencialmente entregable.
 - No se admiten cambios de requisitos ni de miembros del equipo.



Referencia

Metodología Scrum

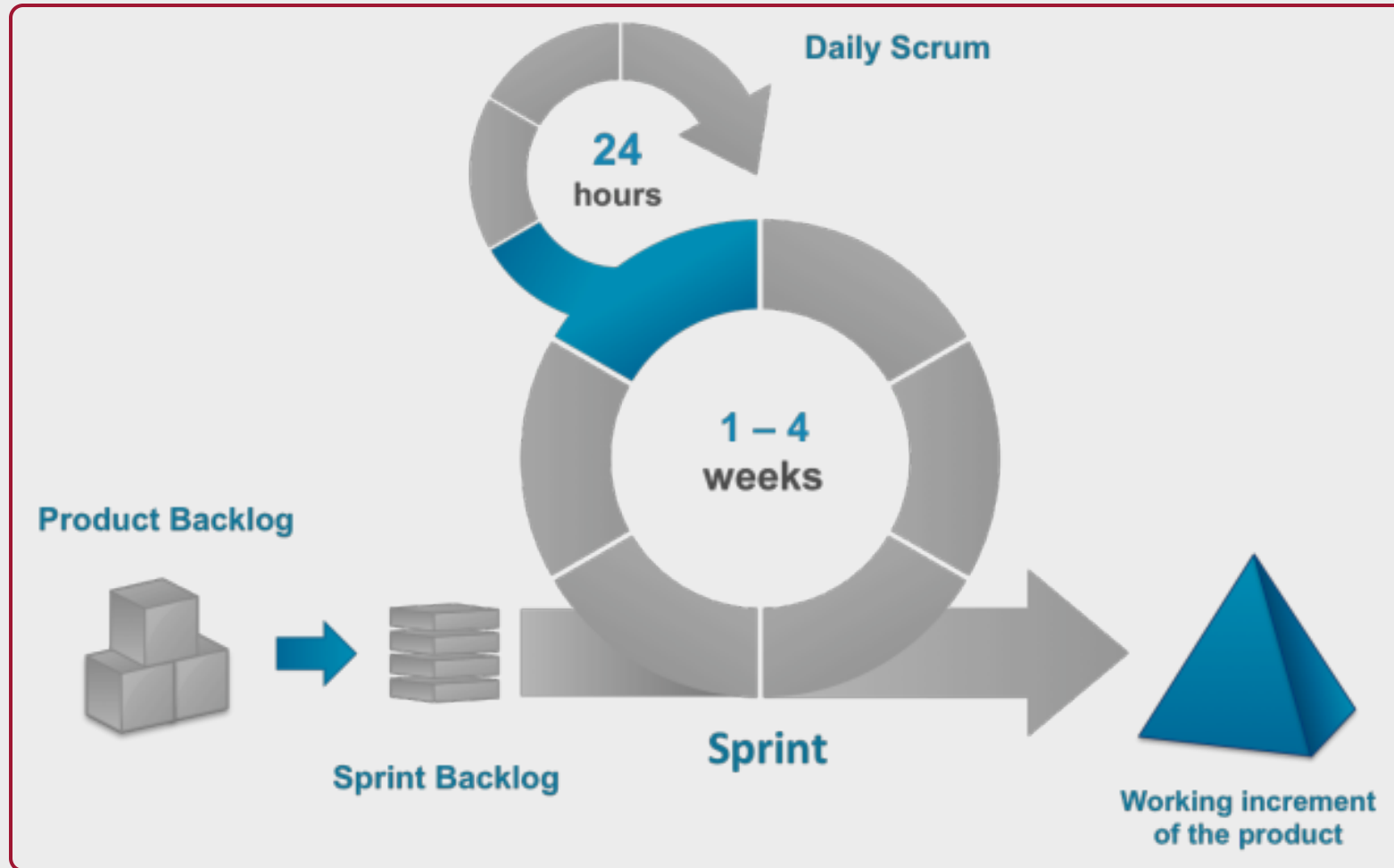
Agile meeting: reuniones cortas y frecuentes donde cada miembro expone:

- Qué ha hecho desde la última reunión.
- Qué problemas ha tenido.
- Qué va a desarrollar hasta la próxima reunión.

Backlog: lista priorizada de tareas.

- Sustituye a diagramas de Gantt en el seguimiento diario.
- Se distinguen backlog de producto y backlog de iteración.

Modelo de procesos de Scrum



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

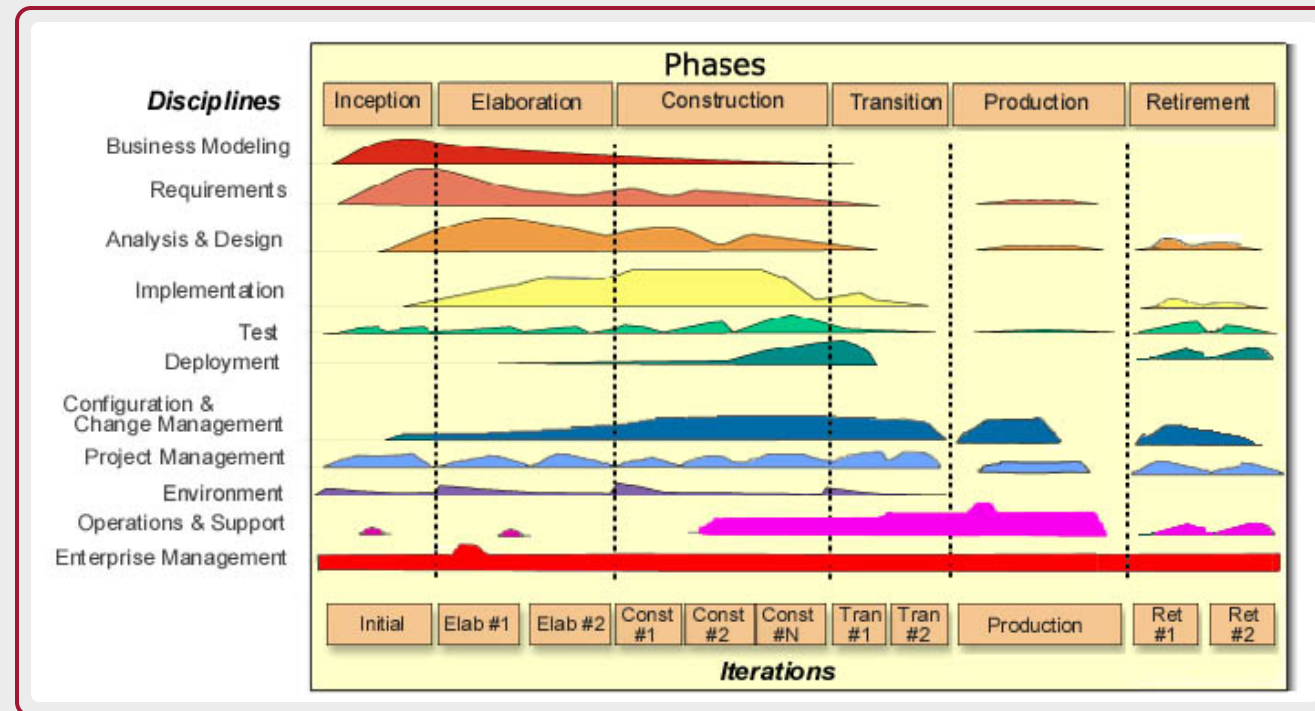
Metodología Scrum

Proceso Unificado

Métrica 3

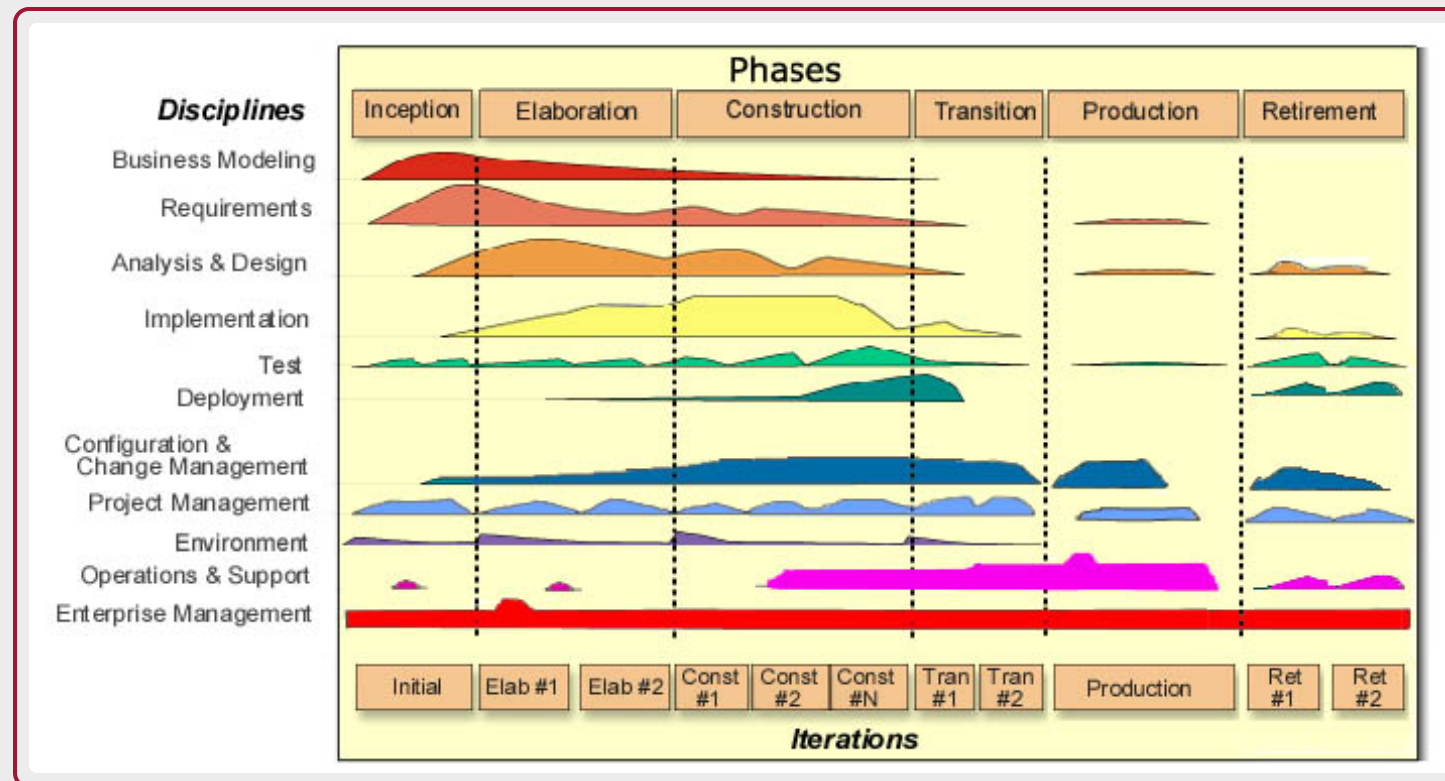
Ciclo de vida del Proceso Unificado

- Proceso iterativo e incremental propuesto por los creadores de UML.
- Define seis fases: *Inicio, Elaboración, Construcción, Transición, Producción, Retirada*



Ciclo de vida del Proceso Unificado

- En cada fase se producen una o más iteraciones.
- Cada iteración genera una versión evaluable del software.



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

Ciclo de vida en Métrica 3

Metodología oficial de las Administraciones Públicas en España.

Métrica 3 permite aplicar diferentes ciclos de vida. Procesos básicos:

- Plan de Sistemas de Información (**PSI**)
- Desarrollo de Sistemas de Información
- Estudio de Viabilidad del Sistema (**EVS**)
- Análisis del Sistema de Información (**ASI**)
- Diseño del Sistema de Información (**DSI**)
- Construcción del Sistema de Información (**CSI**)
- Implantación y Aceptación del Sistema (**IAS**)
- Mantenimiento de Sistemas de Información (**MSI**)

Ciclo de vida en Métrica 3

También incluye procesos de apoyo.

- Gestión de proyectos.
- Seguridad.
- Gestión de la configuración.
- Aseguramiento de la calidad.

Puede descargarse desde: <http://administracionelectronica.gob.es>

Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

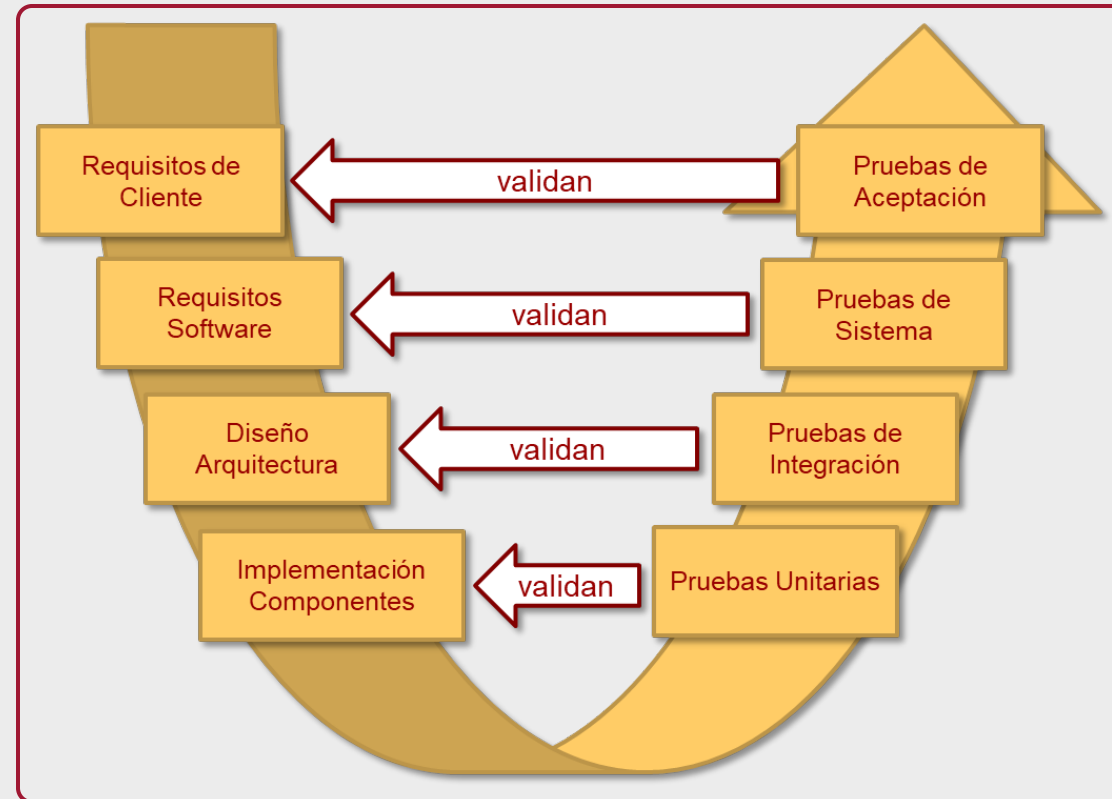
Metodología Scrum

Proceso Unificado

Métrica 3

Pruebas en el ciclo de vida

El modelo en V asocia un tipo de pruebas a cada producto de cada fase según su nivel de abstracción.



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

Ingeniería inversa

- A veces es necesario mantener sistemas heredados (**legacy systems**) sin documentación disponible.
- Consiste en analizar el resultado de una fase para obtener el resultado de la fase anterior.
- El caso más habitual es analizar código para recuperar diseño.



Contenido

Concepto de ciclo de vida

Ciclo de vida clásico

Ciclos de vida evolutivos

Ciclos de vida ágiles

Metodología Scrum

Proceso Unificado

Métrica 3

Reingeniería

- La reingeniería usa la información obtenida mediante ingeniería inversa para aplicar mantenimiento.
- El mantenimiento preventivo del **efecto 2000** fue el mayor esfuerzo de ingeniería inversa y reingeniería de la historia de la Ingeniería del Software hasta la fecha.

Conclusiones

El ciclo de vida del software es el marco de referencia que contiene los procesos, actividades y tareas involucrados en el desarrollo, operación y mantenimiento de un producto software, desde su definición hasta su retirada. Especifica el enfoque general del desarrollo, indicando el orden de ejecución y qué productos se generarán.

El ciclo de vida clásico o en cascada asume que se conocen todos los requisitos al inicio y sigue una progresión lineal: requisitos, diseño, implementación, pruebas y entrega. Aunque es fácil de planificar, es rígido y retrasa la detección de errores. La realidad del software requiere mayor flexibilidad.

Los ciclos evolutivos (incremental, iterativo y prototipos) permiten trabajar en ciclos de requisitos-desarrollo-evaluación, entregando valor antes de completar todas las funcionalidades. Cada iteración refina el producto hacia el sistema final.

Los ciclos ágiles, como Scrum y Kanban, enfatizan la adaptabilidad, colaboración continua y entrega frecuente. Son especialmente efectivos en proyectos con requisitos cambiantes. El Proceso Unificado integra elementos de cascada y agilidad.

Métrica 3 es el estándar para adaptación del ciclo de vida en España, incluyendo procesos de apoyo como gestión de proyectos, seguridad, gestión de configuración y aseguramiento de calidad. Las pruebas acompañan cada fase según el modelo en V, asociando tipos de pruebas a cada nivel de abstracción.



Gracias

Universidad de Sevilla

**Departamento de Lenguajes y Sistemas
Informáticos**

UNIVERSIDAD DE SEVILLA