

INGENIERÍA DEL SOFTWARE II

TEMA 1 INTRODUCCIÓN

INTRODUCCIÓN

- Retos de la ingeniería del software
- Definición de software
- Definiciones, elementos, y beneficio de la ingeniería de software
- Proceso y Modelo de Proceso
- Estrategias, Actividades, Métodos y Tareas
- Modelo de Cascada y Modelo de Espiral

Retos de la ingeniería del software

- Un poco de *historia*
 - *Primeras* décadas:
 - desarrollar el hardware
 - reducir costes de procesamiento y almacenamiento
 - Década de los **80's**:
 - desarrollo de la microelectrónica
 - mayor potencia de cálculo y reducción de costes
 - Objetivo *actual*:
 - mejorar la calidad de las soluciones software

Retos de la ingeniería del software

- **Problemas** del desarrollo de software:
 - Baja explotación del potencial del hardware
 - Incapacidad de atender a la demanda o a las expectativas
 - Incapacidad de mantener el software existente (sistemas heredados)
- La “crisis del software”:
 - problemas que aparecen en el desarrollo del software al desarrollar, mantener y atender la demanda de nuevas aplicaciones

Retos de la ingeniería del software

- Un enfoque informal o “heroico” demostró no ser buena solución
 - Años de retraso
 - Presupuesto sobrepasado
 - Sistemas difíciles de mantener
 - Desempeño pobre
 - Y esto cuando se terminaban los proyectos...
- Pero no caigamos en el desánimo, ¡los sistemas software complejos existen y funcionan!



Retos de la ingeniería del software

- Evitar los **desastres** por causa del software:
 - El **Therac-25**, una máquina de terapia por radiación para el cáncer, provoca la muerte de varias personas entre los años 1985 y 1987, por sobredosis de radiaciones
 - El cohete **Ariane 5**, lanzado por la Agencia Espacial Europea, explota durante el lanzamiento, por desborde en el sistema de referencia inercial
 - Guerra del Golfo: un misil Scud iraquí alcanza un cuartel norteamericano en Dharan, Arabia Saudí, tras fallar los dispositivos antiaéreos **Patriot**, por fallo en el reloj de sistema
 - **Y otros...**



Retos de la ingeniería del software

- Evitar el uso injustificado del término “fallo informático”:
 - Un Airbus chino se estrella en Nagoya el 16 de Abril de 1994, mueren 264 personas, por fallo del piloto
 - El prototipo español del Eurofighter 2000 se estrella en Toledo, por fallo de los motores
 - El Mars Climate Orbiters se estrella al aterrizar en Marte, por una confusión entre millas y kilómetros, por fallo de comunicación
 - Y otros...
- Minidesastres: por culpa del efecto del año 2000 una señora de 104 años de Minnesota recibió una invitación para acudir a la guardería



Definición de software

- Para muchos sólo son programas de ordenador
- En realidad deberíamos considerar que es **mucho más** y darle su importancia:
 - + programas
 - + archivos de configuración
 - + documentación de la estructura del sistema
 - + manuales de instalación y uso
 - + sitios web con información y actualizaciones

Definiciones de ingeniería del software

1. Establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico, fiable, eficiente y que satisfaga las necesidades del usuario

Definiciones de ingeniería del software

2. Disciplina de ingeniería que comprende todos los aspectos de la producción de software, desde las etapas iniciales hasta el mantenimiento:
 - “*disciplina de ingeniería*”: aplicación de teorías, métodos y herramientas para solucionar problemas, y teniendo en cuenta restricciones financieras y organizativas
 - “*todos los aspectos de producción*”: comprende tanto procesos técnicos del desarrollo y actividades como la administración de proyectos, desarrollo de herramientas, métodos y teorías

Elementos de la ingeniería del software

- Trata de ser la **respuesta** a la “crisis del software” mediante una combinación de elementos:
 - Mejores elementos de programación
 - Métodos completos para todas las fases
 - Mejores técnicas de control de calidad
 - Herramientas para automatizar los métodos
 - Filosofía de coordinación, control y buena gestión
 - Además recomienda el uso del **modelado**

Elementos de la ingeniería del software

- *Modelado*
 - Representación gráfica y abstracta de algo real
 - Los *modelos de sistema* intentan expresar visualmente lo esencial del trabajo sin proporcionar detalles excesivos
 - Es más barato construir modelos
 - Permite verificar el sistema
 - Facilita la tarea del usuario
 - Para comprender el sistema se necesita hacer uso de múltiples modelos que se enfocan a aspectos diferentes (diferentes *vistas del sistema*)

Ingeniería del software frente a desarrollo “tradicional”

- El desarrollo tradicional era una mezcla de informalidad, urgencia, intuición y arte
- Si las cosas marchaban bien, los desarrolladores eran considerados unos **héroes**
- Pero las cosas no siempre marchan bien, y cuando esto pasa los héroes se **estrellan** y se **queman** (*Pressman & Lowe, “Web Engineering”*)
- La ingeniería de software propone un entorno de trabajo **disciplinado** para el desarrollo de aplicaciones de calidad
- Reduce el riesgo (pero no lo elimina) y aumenta las probabilidades de **éxito**

Proceso

- Según la *RAE*, acciones y operaciones que conducen a un fin
- En el software, abarca desde la creación a la entrega de un producto software de calidad, verificado y validado (y su mantenimiento), esto es, a lo largo del *ciclo de vida* del software

Modelo de Proceso

- Define un orden o secuencia para efectuar las distintas operaciones del *proceso*
- Es un grupo de *estrategias*, *actividades*, *métodos* y *tareas* para lograr una meta

Modelo de Proceso

- Abarca *aspectos* como:



- **Planificación:** cómo se llevarán a cabo las diferentes etapas del proceso



- **Autoridad:** cómo se puede influir para llegar al objetivo



- **Predicción:** a dónde se va a llegar



- **Evaluación:** cuál es el estado del proceso actualmente



- **Rastreabilidad:** cómo se ha llegado a un resultado

Proceso

- En particular *el proceso* es considerado como un conjunto de:
 - Personas
 - Estructuras organizacionales
 - Reglas
 - Políticas
 - Actividades
 - Componentes de software
 - Metodologías
 - Herramientas para automatizar el proceso de generación de software

Modelo de Proceso

- Existen unas **creencias generales** (opiniones, máximas, experiencia):
 - Comprender antes de desarrollar
 - Resultado predecible, sin importar individuo
 - Planificar y calcular el proceso con gran precisión
 - Evaluar y administrar el riesgo importante para el éxito
 - Etapas bien definidas con entregas intermedias aumentan la confianza
- En general, estas creencias actúan como base para definir las **estrategias, actividades, métodos, y tareas** del ***modelo de proceso***

Estrategias

- **Plan** para llevar a cabo un objetivo, en nuestro caso el desarrollo de software de calidad
 - **Tecnología** (tipo de programación, lenguajes)
 - **Arquitectura** (estructura general del sistema, componentes, niveles)
 - **Desarrollo** (cómo se llevan a cabo las actividades, iterativo, incremental)
 - **Prototipo** (versión incompleta del sistema)
 - De Requisitos, Análisis, Diseño, Vertical, Factibilidad
 - **Reutilización**
 - Entre proyectos, en un mismo proyecto

Actividades

- Unidad o paso organizacional para llevar a cabo cierto aspecto de un proceso (*flujo de trabajo, tipo de trabajo*)
 - **Requisitos:** necesidades del sistema; todo se prueba frente al modelo de requisitos
 - **Análisis:** comprender los requisitos del sistema logrando la estructura lógica, *qué* debe hacer el sistema, modelo más formal de lo que debe hacer el sistema
 - **Diseño:** se implementa el análisis en la tecnología adecuada; entran en juego rendimiento, concurrencia y particularidades del lenguaje; se responde a *cómo* se debe hacer el sistema

Actividades

- **Implementación:** se expresa el sistema en una forma aceptable para la computadora, o sea el código
- **Pruebas:** se verifica y valida a nivel de componentes y de integración, se *verifica* si el sistema es correcto y se *valida* si es lo que el cliente quiere
- **Integración:** se unen los distintos componentes
- **Documentación:** se describen los aspectos sobresalientes, distintos manuales
- **Mantenimiento:** corregir errores y ampliar funcionalidad, mejoras

Actividades

- Otras actividades:
 - **Comunicación**
 - intercambio de modelos y documentos, informes de evolución y calidad, negociaciones, comunicación de decisiones,...
 - **Gestión de la configuración**
 - proceso que supervisa y controla los cambios en los productos de trabajo
 - **Administración del proyecto**
 - asegurar la entrega de un sistema de alta calidad a tiempo y dentro del presupuesto
 - planificación y presupuesto del proyecto
 - contratación de desarrolladores y coordinación de equipos
 - vigilancia de la evolución del proyecto
 - detección de desviaciones e intervención

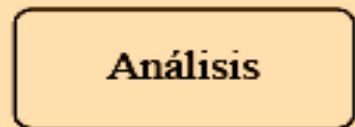
Métodos, Tareas y Metodologías

- *Métodos*: definen las *tareas* que deben llevarse a cabo para satisfacer una *actividad*, los métodos proveen técnicas para la construcción de modelos
- *Tareas*: acciones que contribuyen a acciones mayores para lograr un objetivo, cada *método* define un conjunto de tareas
- *Metodologías*: se refieren al estudio de los métodos, conjuntos de métodos
 - cuerpo de conocimientos y técnicas que abarcan el ciclo de vida del desarrollo de software
 - distintas metodologías, distinta forma de llevar a cabo las *actividades* de desarrollo de software
 - asociadas con un lenguaje preciso (*notación*)

Actividad, Método y Notación

Modelo de Proceso

(Actividades)



Análisis

Diseño

Implementación

Método

Métodos de Análisis

- OBA
- Fusion
- Objectory

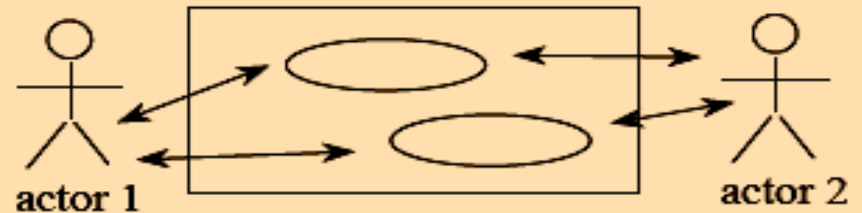
Métodos de Diseño

- OMT
- Booch
- Fusion
- RDD

Método de Codificación

- Técnicas de Java
- Técnicas de C++
- Técnicas de Smalltalk

Notación



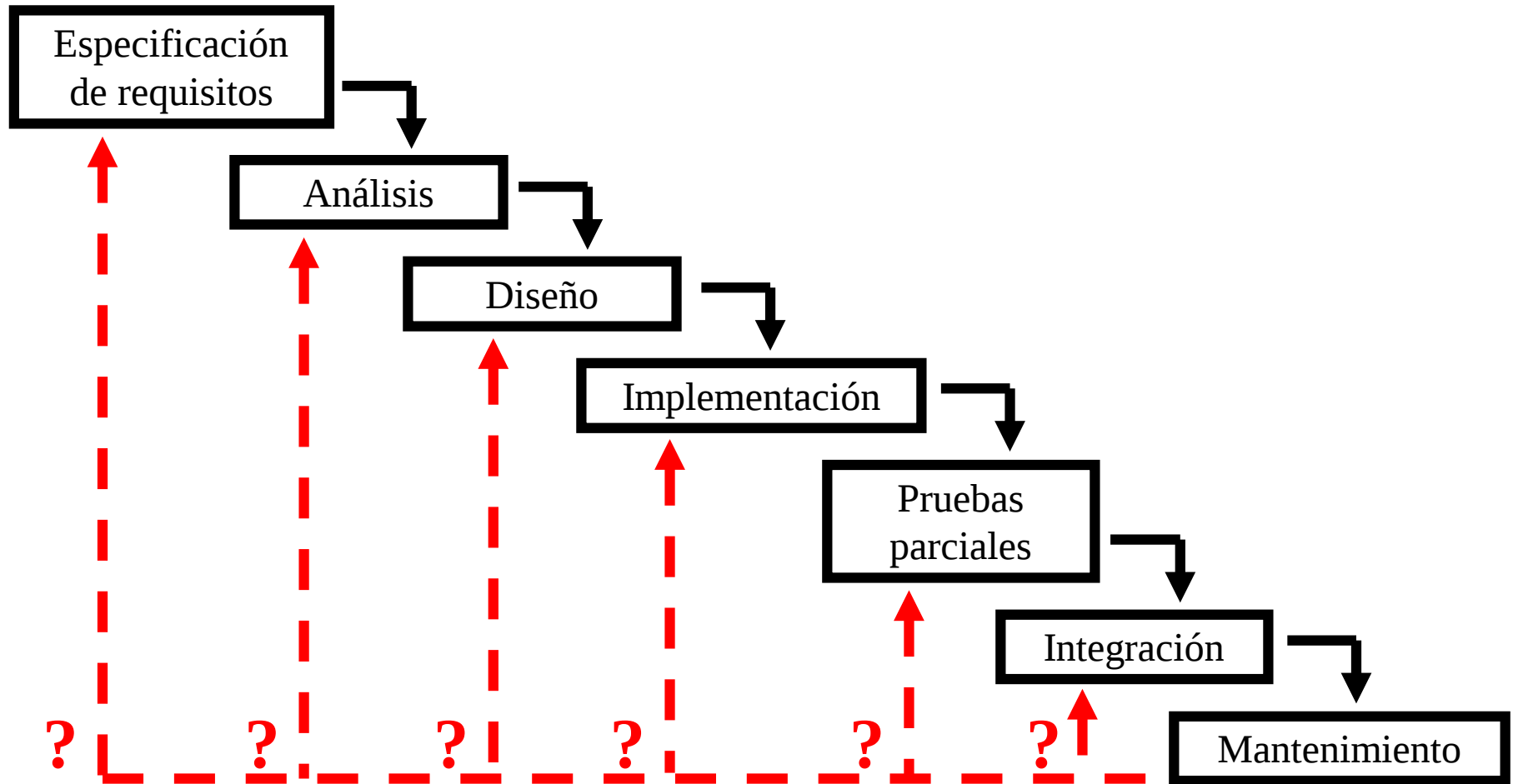
Casos de Uso en Objectory



Asociación de clases en OMT

```
class Cuadrado extends Figura {  
    ....  
}
```

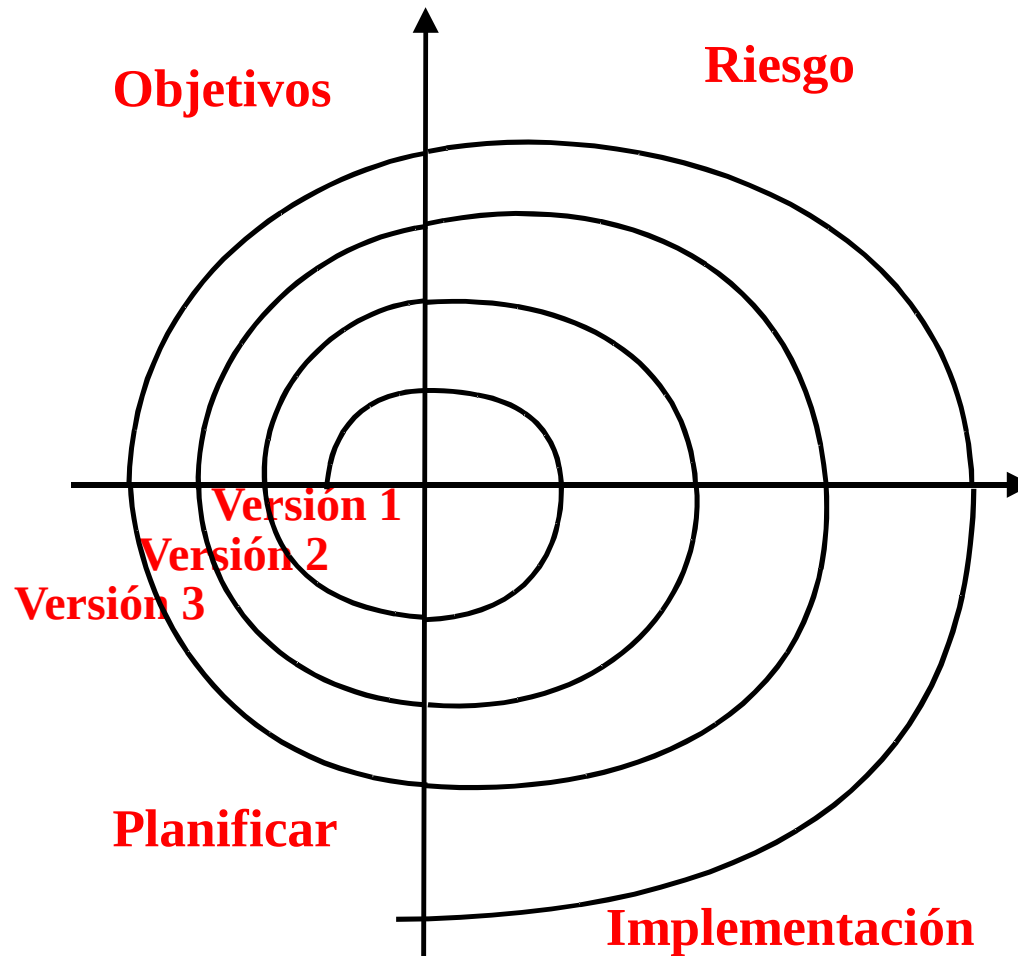
Modelo de Proceso: Modelo de Cascada



Modelo de Proceso: Modelo de Cascada

- *Máximas (principios o reglas)* del Modelo de Cascada
 - Actividades secuenciales con puntos de revisión definidos
 - Documentación comprensible para no-técnicos
 - Cada requisito puede conocerse y es estable
 - Se pueden llevar a cabo pruebas y evaluaciones
- *Debilidades*
 - ☒ Los documentos a entregar son los que rigen el proceso
 - ☒ Demasiado tiempo para ver resultados
 - ☒ Depende de requisitos estables
 - ☒ Difícil rastrear requisitos
 - ☒ Retrasa la detección de errores hasta el final
 - ☒ No favorece la reutilización
 - ☒ No promueve el uso de prototipos

Modelo de Proceso: Modelo de Espiral



Modelo de Proceso: Modelo de Espiral

- *Máximas* del Modelo de Espiral
 - Cada ciclo comienza con un buen entendimiento de los objetivos y riesgos involucrados
 - Basado en la evaluación de soluciones alternas, reducir riesgos
 - Revisión y Planificación
 - Desarrollo en incrementos
- *Debilidades*
 - ☒ Sólo para productos locales gran escala
 - ☒ Requiere gran habilidad para la evaluación del riesgo
- Modelo muy utilizado con algunas variantes

Bibliografía

- **Sommerville**, “Ingeniería de Software,” Novena Edición, Pearson Educación, 2011 (parte de los capítulos 1 y 2)
- **Pressman**, “Ingeniería del Software,” Séptima, Edición, McGraw-Hill, 2010 (parte de los capítulos 1 y 2)