

INGENIERÍA DEL SOFTWARE



Departamento de Lenguajes
y Sistemas Informáticos

Facultad de Informática



OCW 2013

Competencias

(al terminar el curso, se espera que el estudiante sea capaz de)

- Distinguir las diversas etapas que componen todo proceso de ingeniería del software
- Entender un sistema software con orientación a objetos en el lenguaje UML
- Diseñar un sistema software en una arquitectura de varios niveles
- Implementar un sistema a partir del diseño de la aplicación

Temario

- TEMA 1: Introducción a la Ingeniería del Software
- TEMA 2: Captura de requisitos usando UML
 - 2.1- Modelo de Casos de Uso
 - 2.2.- Modelo del Dominio
- TEMA 3: Diseño
 - 3.1.- Diagramas de secuencia UML
 - 3.2.- Patrones de responsabilidad GRASP
 - 3.3.- Arquitecturas Software de varios niveles
- TEMA 4: Implementación usando Java
 - 4.1.- Interfaces de usuario gráficas: SWING/AWT
 - 4.2.- Persistencia de objetos: db4o
 - 4.3.- Computación distribuida: RMI

Bibliografía

- **Libro general sobre Ingeniería del Software:**
- Ingeniería del Software. Un enfoque práctico. Roger S. Pressman. MacGraw-Hill, 2001. 5ª Edición.
- **Libros sobre UML y proceso unificado de desarrollo de software:**
- Ingeniería El Proceso Unificado de Desarrollo de Software. Jacobson, Booch, Rumbaugh. Editorial Addison Wesley, 1999
- **Libros sobre UML y patrones GRASP:**
- UML y Patrones: introducción al análisis y diseño orientado a objetos. Craig Larman. Prentice-Hall, 2003
- **Libro sobre Orientación a Objetos.**
- Construcción de Software Orientado a Objetos. Bertrand Meyer. Prentice-Hall. 1998.

Tema 1: Introducción a la ingeniería del software



A. Goñi, J. Ibáñez, J. Iturrioz, J.A. Vadillo



OCW
2013



Índice

- Motivación y definición de Ingeniería del Software
- Calidad del Software
- Proceso de ingeniería del producto (software)
- Proceso de desarrollo software
- Proceso de gestión del proyecto
- Ciclo de Vida
- El Proceso Unificado de Desarrollo de Software

Motivación

A las 14:21 de un 15 de enero, hora punta en el tráfico telefónico, el conmutador 4ESS de la red telefónica de Manhattan detectó un pequeño fallo en su hardware. El 4ESS se desconectó de la red tras, amablemente, notificar a los conmutadores vecinos su desconexión. En pocos segundos, el conmutador volvió a funcionar, lo que notificó también a sus vecinos. Pero éstos estaban todavía procesando el primer mensaje cuando recibieron el segundo. Esto les confundió, se dieron cuenta de este desconcierto y se desconectaron de la red para realizar un auto-chequeo y reinicializarse, no sin antes notificar a todos sus vecinos su desconexión. Como una epidemia de gripe, los mensajes se distribuyeron por todo EEUU. Tan pronto como un conmutador se desconectaba y se recuperaba, recibía un aluvión de mensajes de sus compañeros que le hacían volver a fallar. En el centro de operaciones, los técnicos veían cómo las líneas de comunicaciones del mapa se ponían rojas por todo el país partiendo de Manhattan, mientras que los ingenieros se veían negros siguiendo todos los procedimientos estándar para la solución de problemas, los no estándar y los improvisados. Pero nada funcionó. Un minúsculo error en la versión de diciembre del software del conmutador 4ESS causó el caos. AT&T volvió temporalmente a la versión anterior e instaló la versión corregida cinco días después. Pero el daño ya estaba hecho.

[Arthur, L.J. Improving Software Quality: an Insider's Guide to TQM. John Wiley & Sons, 1993

Motivación

Un minúsculo error en la versión de diciembre del software del conmutador 4ESS causó el caos.

1. Definición de Ingeniería del Software

- Bauer 1972: IS trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales.
- Bohem, 1976: IS es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos. Se conoce también como desarrollo de software o producción de software.
- Zelkovitz 1978: IS es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software.

1. Definición de Ingeniería del Software

- Mills, 1980: la IS tiene como uno de sus principales objetivos la producción de programas que cumplan las especificaciones, y que se demuestren correctos, producidos en el plazo y coste adecuados.
- Meyer 1988: la IS es la producción de software de calidad.
- Ford, 1990: IS es una forma de ingeniería que aplica los principios de la ciencia de los computadores y matemáticas para conseguir soluciones a los problemas del software de forma efectiva y económica.
- IEEE 1993: IS es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de ingeniería al software.

1. Definición de Ingeniería del Software

Ingeniería del Software es la **INGENIERÍA** que trata de construir software de **ALTA CALIDAD** a **BAJO COSTO**

A) Qué es **CALIDAD** del **SOFTWARE**

B) Cómo construir el software: **PROCESO SOFTWARE**

- **PROCESO** de **INGENIERÍA** del **PRODUCTO**
(**SOFTWARE**)

- **MEJORA** del **PROCESO** **SOFTWARE**

2. Calidad de software

ES UNA COMBINACIÓN DE FACTORES

Factores Externos ← **Los que importan**

- ⇒ Corrección
- ⇒ Robustez
- ⇒ Extensibilidad
- ⇒ Reusabilidad
- ⇒ Compatibilidad

- ⇒ Eficiencia
- ⇒ Portabilidad
- ⇒ Facilidad de Uso
- ⇒ Funcionalidad
- ⇒ Oportunidad

Factores Internos: modularidad y legibilidad.

↑
La clave para conseguirlos

3. Proceso de ingeniería del producto (software)



3. Proceso de ingeniería del producto (software)

- Objetivo del proceso de ingeniería del producto:

conseguir el producto deseado

- Dentro de este, se distinguen 3 procesos:
 - Proceso de desarrollo software
 - Proceso de gestión del proyecto
 - Proceso de control de configuraciones

4. Proceso de desarrollo software

- Objetivo: desarrollo de un producto que satisfaga al cliente



4. Proceso de desarrollo

software (también llamado “Metodología”)

- OFRECE UN MARCO DE TRABAJO GENÉRICO:

REPRESENTAR DATOS Y ARQUITECTURA

PARTE
ESTÁTICA

REPRESENTAR PROCESOS/FUNCIONALIDAD
DEL SISTEMA

PARTE
DINÁMICA

REPRESENTAR RELACIÓN USUARIOS CON SISTEMA

INTERFAZ

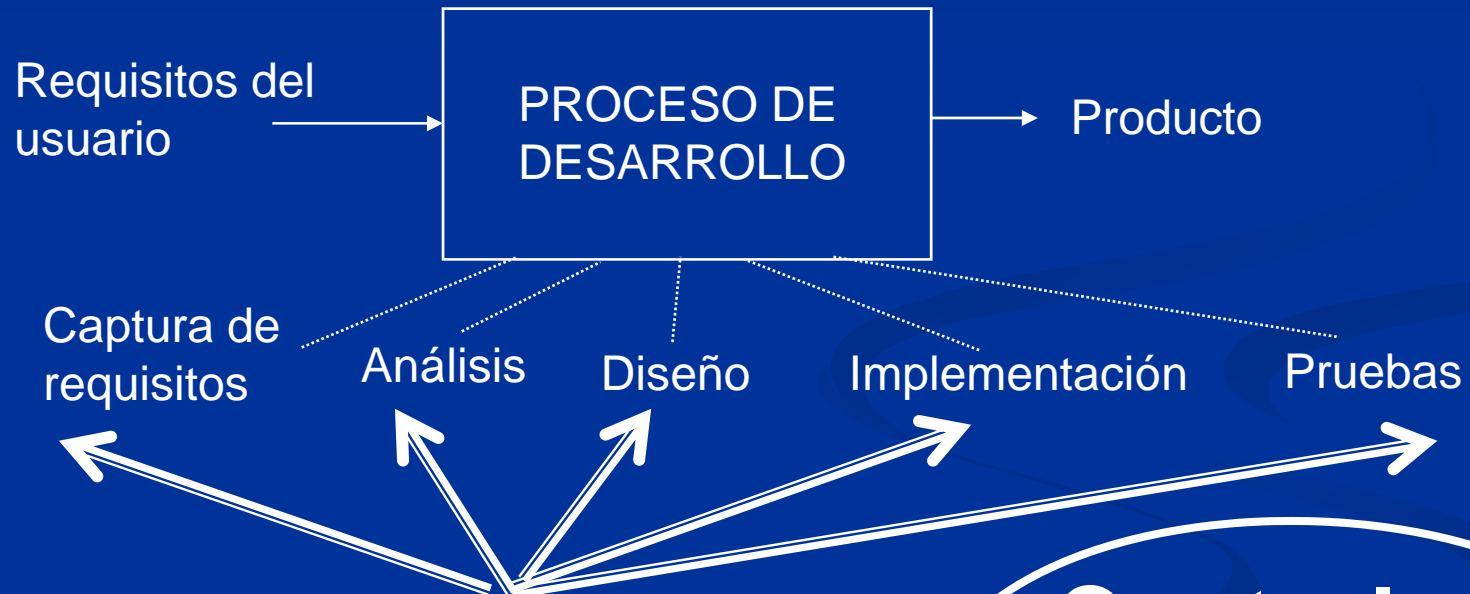
MODELO DE REFERENCIA

CICLO DE
VIDA

- Ejemplos de Procesos de Desarrollo / Metodologías:
Merise, SSADM, Métrica, XP (programación extrema),
Proceso Unificado de Desarrollo de Software

5. Proceso de gestión del proyecto

- Control y planificación de las actividades para alcanzar el objetivo



**PROCESO DE GESTIÓN
DEL PROYECTO**

**Controla y
Planifica**

5. Proceso de gestión del proyecto

- Se trata de realizar una serie de actividades de planificación y control:
 - del alcance del proyecto
 - del tiempo
 - del costo
 - de recursos humanos
 - de calidad
 - de riesgos
 - ...

6.- Ciclo de Vida (CV)

CV es MODELO de REFERENCIA para construir software

Tipos de Ciclos de Vida:

- Sin prototipos
 - En cascada (Waterfall)
- Con prototipos
 - desechables
 - no desechables
- En espiral
- Iterativo e incremental

7.- El proceso unificado de desarrollo de software

- Es un proceso de desarrollo **ORIENTADO A OBJETOS**
 - La construcción de software se basa en **componentes Orientados a Objetos**
- Usa el lenguaje unificado de modelado (UML)
 - 1. Guiado por *casos de uso*
 - 2. Centrado en la *arquitectura*
 - 3. Con un ciclo de vida *iterativo e incremental*

7.- El ciclo de vida del Proceso Unificado de Desarrollo de Software

- El CV consta de cuatro fases:
 - iniciación, elaboración, construcción, transición
- Cada fase está dividida en varias iteraciones
- En cada iteración se realizan flujos de trabajo
- Tras el ciclo de CV se obtiene un nuevo producto (o versión) para el cliente

7.- El CV del proceso unificado

Flujos de trabajo:
Actividades



Requisitos

Análisis

Diseño

Implementación

Prueba

Fases

