Ingeniería de Software

Agustín J. González

ElO329: Diseño y Programación Orientados a Objeto

Adaptado de: **http://www.dsic.upv.es/~uml**
http://inst.eecs.berkeley.edu/~cs169/ entre otras fuentes.

Definiciones

- (1993) La aplicación de mecanismos sistemáticos, disciplinados, y cuantificables para el desarrollo, operación y mantención de software; esto es la aplicación de la ingeniería al software.
- Establecimiento y uso de principios con caracteres de ingeniería apropiados para obtener, eficientemente, software confiable, que opere eficaz y eficientemente en máquinas reales
- La aplicación del arte del desarrollo software junto con las ciencias matemáticas y computadores para diseñar, construir, y mantener programas computacionales eficientes y económicos que logran sus objetivos.
- Wikipedia: “Software engineering is a profession and field of study dedicated to designing, implementing, and modifying software so that it is of higher quality, more affordable, maintainable, and faster to build.”
- Se busca: Resolver el problema en costo y tiempo controlados.

Estado del arte en Ing. de Software

- ¿Es una ciencia rigurosa con fuertes fundamentos matemáticos?
- ¿Es una campo técnico bien desarrollado con mucho de disciplina de ingeniería?
- O está realmente en un estado primitivo...
  - A lo más una serie de “mejores prácticas”, desarrolladores de software construyen software y si éstos funcionan entonces nosotros estudiamos cómo ellos lo hicieron.
  - Si éstos funcionan por un largo tiempo entonces estudiamos sus procesos de software aun más cuidadosamente.

Construcción de una casa para “fido”

Puede hacerlo una sola persona
Requiere:
- Modelado mínimo
- Proceso simple
- Herramientas simples

Construcción de una casa

Construida eficientemente y en un tiempo razonable por un equipo
Requiere:
- Modelado
- Proceso bien definido
- Herramientas más sofisticadas

Construcción de un rascacielos

Claves en Desarrollo de IS

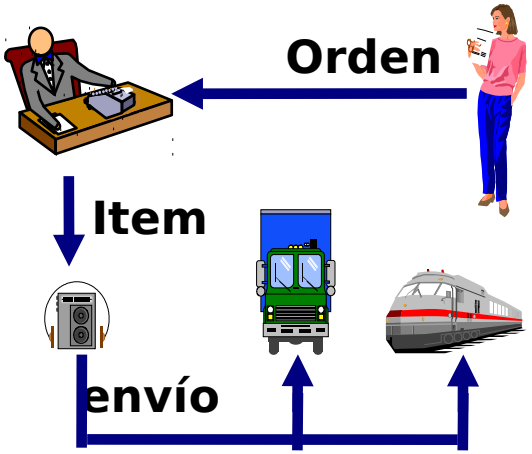
Notación (UML)

Herramientas
(Ej: Rational Rose, Umbrello, IDEs)

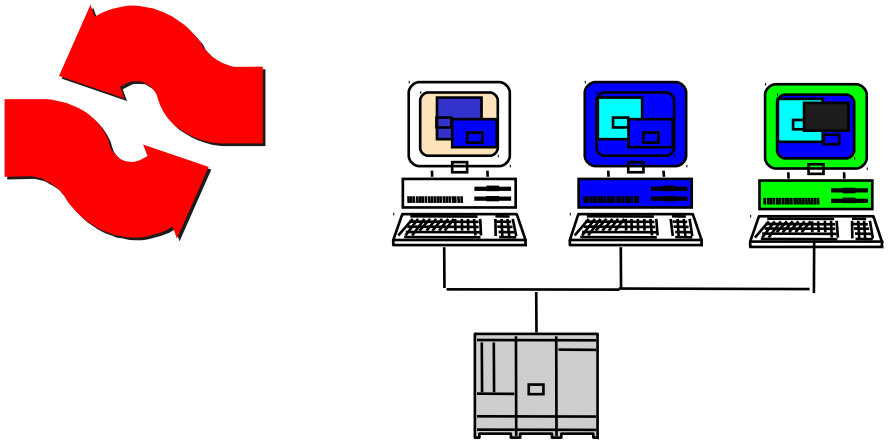
Proceso
(Metodologías
Ej: ITIL, Extreme Programming, RUP: Rational Unified Process, Personal Software Process)

Abstracción - Modelado Visual (MV)

"El modelado captura las partes esenciales del sistema"



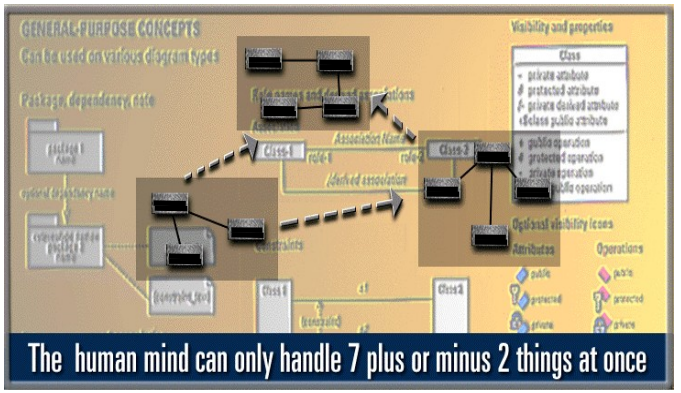
Proceso de Negocios

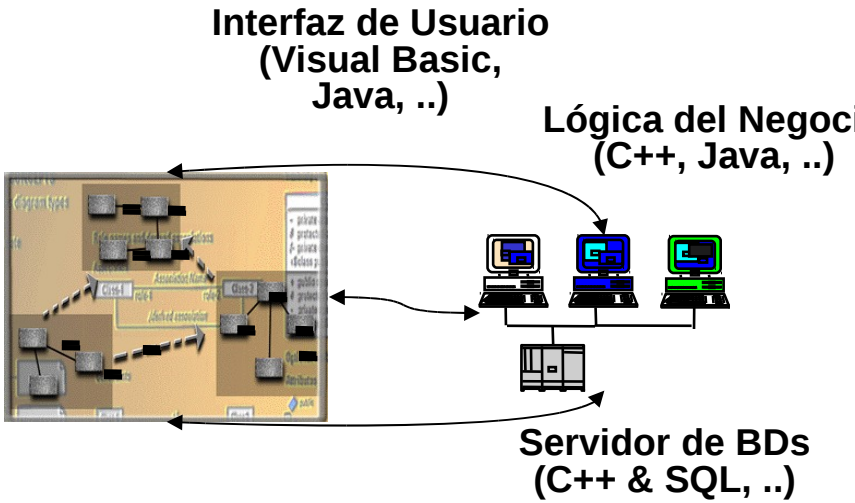


Sistema Computacional

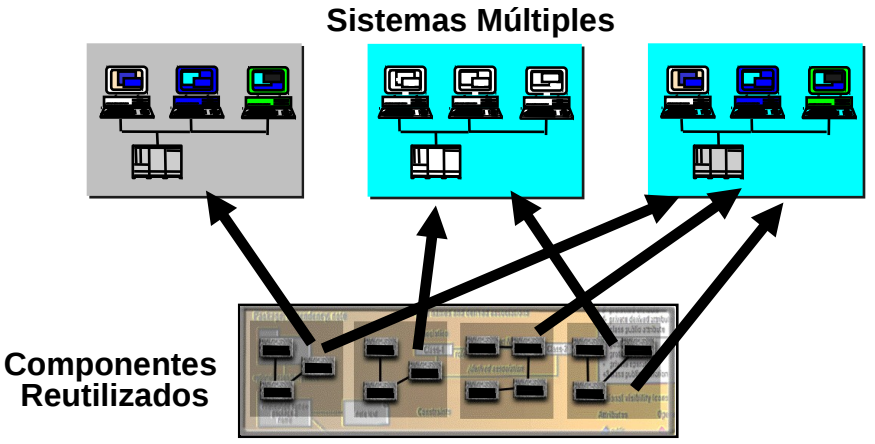
Notación (Visual) - Beneficios

Manejar la complejidad





"Modelar el sistema independientemente del lenguaje de implementación"



Promover la Reutilización

¿Por qué la Orientación a Objetos?

- Por su proximidad de los conceptos de modelado respecto de las entidades del mundo real
  - Mejora la captura y validación de requisitos
  - Acerca el “espacio del problema” al “espacio de la solución”
- Modelado integrado de propiedades estáticas y dinámicas del ámbito del problema
  - Facilita construcción, mantenimiento y reutilización
- Su diseño facilita:
  - la creación de Abstracciones (Ignorar detalles)
  - la Modularización (separación en módulos)
  - Ocultar información (separar la implementación del uso)
- Podríamos dar muchas razones pero hay problemas.

Problemas en OO

"... Los conceptos básicos de la OO se conocen desde hace dos décadas, pero su aceptación todavía no está tan extendida como los beneficios que esta tecnología puede sugerir"

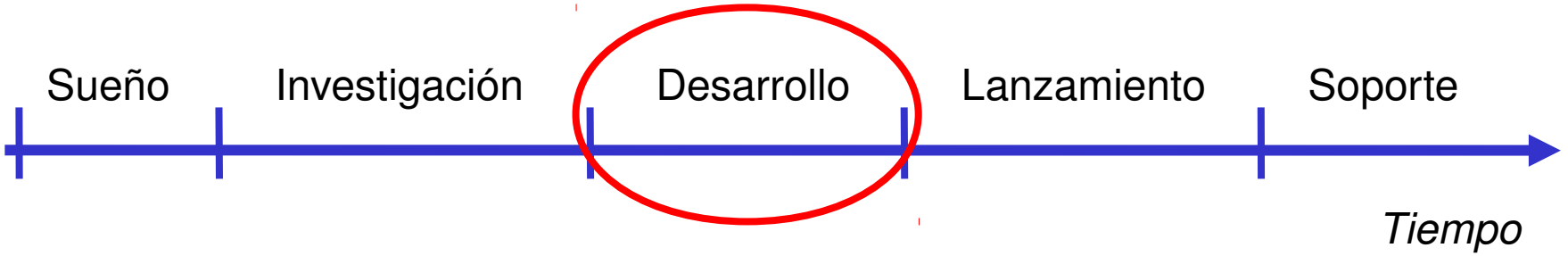
… Problemas en OO

- Un objeto contiene datos y operaciones que manipulan los datos, pero ...
- Podemos distinguir dos tipos de objetos degenerados:
  - Un objeto sin datos (que sería lo mismo que una biblioteca de funciones). Si los métodos son estáticos, “peor” aún.
  - Un objeto sin “operaciones”, con sólo atributos lo que permitiría crear, recuperar, actualizar y borrar su estado (que se correspondería con las estructuras de datos tradicionales)
- Un sistema construido con objetos degenerados no es un sistema verdaderamente orientado a objetos.

Proceso de Desarrollo de SW

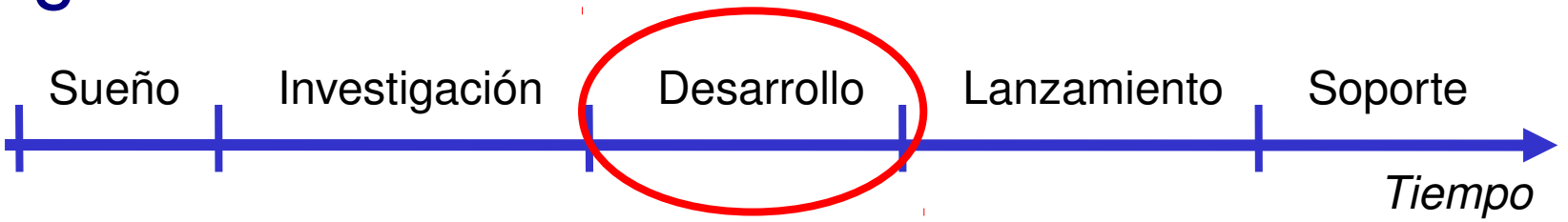
El proceso de desarrollo “Completo”

- Se da en un contexto y dependiendo el texto o investigador destaca más o menos etapas.
- El más completo que he visto incluye:



- Al desarrollo normalmente se le da más énfasis en la literatura.

¿Qué es un Proceso de Desarrollo de SW?



- Define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo

- No existe un proceso de software universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Áreas cubiertas por la Ing. de Solftware

- La ingeniería de Software estudia todos los aspectos involucrados en el desarrollo de software, entre los cuales se encuentran:
  - Etapas de un Proyecto de Software
  - Paradigmas/enfoques para enfrentar el desarrollo de proyectos de software (Cascada, incremental e iterativo)
  - Tiempos y costos
  - Aspectos tecnológicos
  - Gestión de Proyectos

Etapas de un Proyecto de Software

- Etapas/Hitos esenciales requeridos en un proyecto de desarrollo de software:
- Oportunidad de Negocio
- Levantamiento inicial de Requerimientos (Domino del problema)
  - Entendimiento inicial del problema a resolver
  - Reuniones con clientes
  - Especificación inicial de requerimientos
- Confección de la propuesta
  - Estimación de esfuerzo y planificación del proyecto. Uso de técnicas y estadísticas de proyectos similares, entrevistas a personas con experiencia en proyectos similares.
  - Definición del alcance del proyecto
  - Costos

Etapas de un Proyecto de Software (cont)

- Análisis OO (Dominio del problema)
  - Casos de Uso : Análisis dinámico
  - Modelos Conceptuales: Análisis estático
  - Diseño preliminar de Interfaces Gráficas (WEB, etc)
- Arquitectura de la solución
  - Especificación de la tecnología a utilizar (J2SE, J2EE)
  - Especificación de Patrones de Diseño
- Diseño (Dominio de la solución)
  - Diseño de interfaces gráficas (WEB, GUI, comandos, voz)
  - Realización de diagramas de secuencia
  - Realización de diagramas de clases

Etapas de un proyecto de Software (cont)

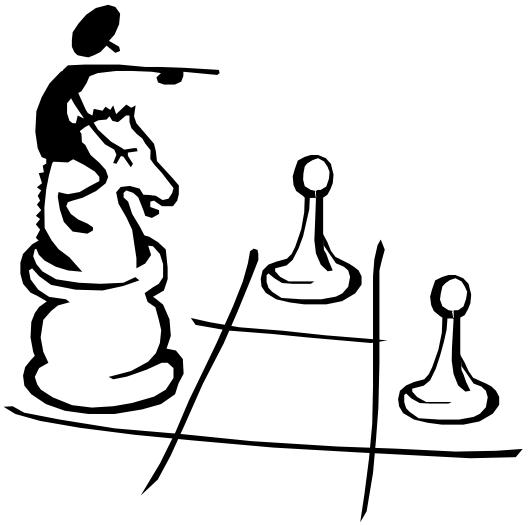
- Construcción / Codificación / Implementación
  - Entendimiento real del diseño (solución) a programar
  - Programación eficaz (y eficiente) del código fuente.
  - Comentar en forma entendible y razonable el código fuente.
  - Construcción de programas de pruebas “Tests”
  - Prueba unitaria del desarrollador mediante un programa de “Test”
- Pruebas Unitarias: Realizar las pruebas unitarias
- Pruebas de Integración: Realizar las pruebas de integración
- Capacitar a Usuarios
  - Confección de manuales de Usuario
  - Reuniones con usuario final
- Puesta en Producción
  - Confección de plan de puesta en producción
- Marcha Blanca
- Garantía: Resolución de Incidencias

Planificar y Evaluar Proyectos ...

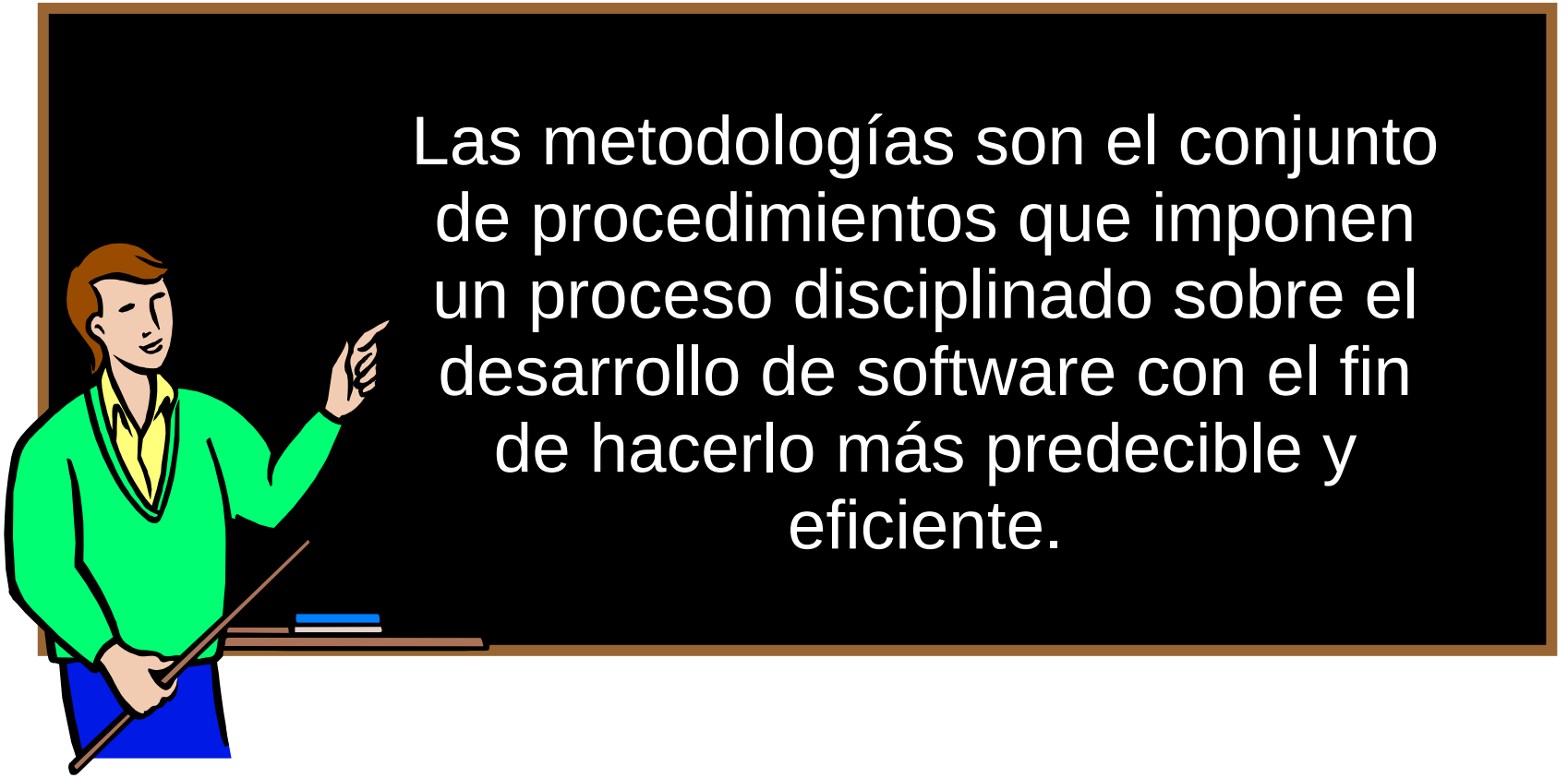
- ¿Podré cumplir con los plazos?
- ¿Estaré dentro de lo presupuestado?
- ¿El “cliente” quedará satisfecho?

Las Metodologías pueden ser la ayuda que necesitamos, si podemos usarlas correctamente !!

Procesos, Metodologías



¿Qué es una Metodología ...



Algunas Metodologías ...

- Personal Software Process y Team software Process
- XP (Programación Extrema)
- RUP (Rational Unified Process)
- Hay muchas otras.

- En este curso veremos algunas ideas de RUP