

Metodologías de diseño y programación

Universidad de Chile

Departamento de Ciencias de la Computación

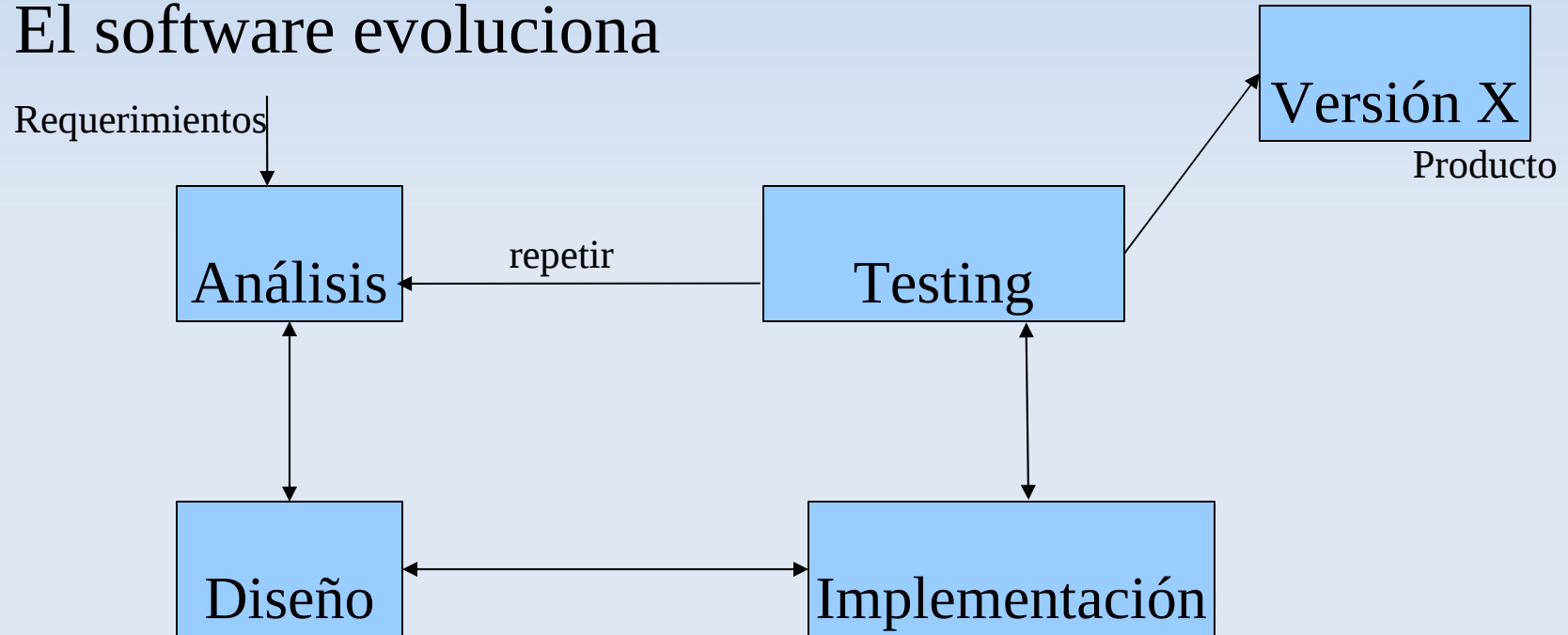
Prof.: Nancy Hitschfeld Kahler

Introducción

- ¿Por qué la construcción de software es tan compleja? [1]
 - Complejidad del dominio del problema
 - ◆ Requerimientos cambian en el desarrollo
 - ◆ Requerimientos complejos y contradictorios
 - ◆ Sistemas evolucionan
 - Complejidad del proceso de desarrollo
 - ◆ Sistemas extensos
 - ◆ Desarrollo requiere de una alta coordinación de equipo

Introducción

El software evoluciona



Metodologías de diseño y programación

Introducción

- **Análisis:**
 - Analizar la documentación entregada por el cliente y entrevistar a usuarios potenciales para entender los requerimientos y determinar las funcionalidades de la aplicación
- **Diseño:**
 - Proponer una posible solución, especificando algoritmos, interfaces, etc
- **Implementación:**
 - Escribir la solución en algún lenguaje de programación (programa)
- **Testing (etapa de prueba):**
 - Ejecutar primero cada una de las partes del programa para determinar si funciona adecuadamente y luego el programa como un todo para verificar el sistema completo
- **Mantenición:**
 - Corregir errores encontrados por los usuarios
 - Modificar requerimientos

Introducción

- Metodologías de programación más comunes:
 - Programación estructurada (clásica antes de 1995 aprox.)
 - Descomposición o división del problema basado en procedimientos
 - Rol principal: procedimientos (funciones)
 - Rol secundario: datos
 - Programación orientada a objetos
 - Descomposición orientada a objetos: problema se resuelve a través de la interacción entre objetos
 - Rol principal: objetos (incluye datos y procedimientos o funciones)
 - Otras: funcional (Lisp) y lógica (Prolog)

Introducción

- Estadística sobre desarrollos usando programación estructurada (procedural): evaluados 487 proyectos
 - Del costo total de los proyectos, cuánto se destina a mantención? 70%
 - Por qué?
 - Cambios en requerimientos del usuario (41,8%)
 - Problema: carencia de extensibilidad
 - Solución: usar orientación a objetos
 - Cambio en formatos de entrada y salida de datos (27,6%)
 - Problema: representación de datos no encapsulada
 - Solución: usar orientación a objetos

Introducción

- Modificaciones de emergencia (12,4%)
 - Problemas: errores catastróficos
 - Solución: orientación objetos, contratos
- Documentación (5,5%)
 - Problema: separada del proceso de desarrollo
 - Solución: orientación a objetos, Eiffel
- Otros: Cambios al hardware (6,2%), eficiencia (4%), etc.

Fechas controles y evaluaciones

- Control 1: lunes 21 noviembre
- Control 2: miércoles 4 enero
- Evaluaciones en grupos (tres personas)
 - Miércoles 26 octubre (14:00 a 15:15)
 - Miércoles 16 noviembre (14:00 a 15:15)
 - Miércoles 7 diciembre (14:00 a 15:15)
 - Miércoles 28 diciembre (14:00 a 15:15)
 - Miércoles 11 enero (14:00 a 15:15)
 - Control 3: promedio de 4 mejores evaluaciones

Tareas

- Individual
- C++ o java
- Incremental
- Uso de herramientas de modelamiento (argoUML, Umbrello, Rational, etc)
- 3 o 4 de programación y una escrita de análisis de la tarea
- Todas obligatorias