

CC31B: Desarrollo de Software de Aplicaciones

2007/2

Prof. M. Cecilia Bastarrica
cecilia@dcc.uchile.cl

Generalidades del Curso

- Clases:
 - Cátedra: Martes/4 y Jueves/4.
 - Auxiliar: Viernes/2.
- Las transparencias se publicarán regularmente en la página web del curso en UCursos.
- Tareas grupales aproximadamente cada dos semanas.
- Dos controles de lecturas, dos controles de materia y un examen final.

2

Conformación de Grupos

- Los grupos de trabajo deberán:
 - tener entre dos y tres personas;
 - decidirse antes del martes 31 de julio;
 - mandarle un mail al auxiliar del curso informándole la conformación de los grupos;
 - quien para el martes 31 no tenga grupo será asignado arbitrariamente.

3

Calificación

- | | |
|------------------------|------|
| • Tareas | 30 % |
| • Controles | 30 % |
| • Controles de lectura | 10 % |
| • Examen | 30 % |
- Con 5.5 de promedio en tareas y 5.5 en controles se puede eximir.

4

Contenidos del Curso

- Definición de Ingeniería de Software.
- Principios y Cualidades del Software y su Desarrollo.
- Ciclo de Vida del Software:
 - especificación,
 - diseño,
 - verificación y validación.
- Proceso de Desarrollo de Software.

5

Ingeniería de Software

¿Qué es la Ingeniería de Software?

6

Ingeniería, Tecnología y Ciencia

- ¿Qué es la tecnología?
 - Aplicación de la ciencia para la resolución de problemas prácticos.
- ¿Qué es la ingeniería?
 - Aplicación de la tecnología en presencia de recursos limitados.
- Pero, ¿qué es la ciencia y cuándo estamos aplicando la ciencia?
 - Cuerpo de doctrina metódicamente formado y ordenado, que constituye una rama particular del saber.
 - Modelos de la realidad cuyo comportamiento es explicable y que permiten predecir el comportamiento de situaciones similares.

7

Definición (I)

- Entonces:
 - La ingeniería de software es la aplicación de la ciencia para la construcción de software en presencia de recursos limitados.
- Pero, ¿qué es el software? y ¿qué ciencia se aplica en su construcción?

8

Software

- El software es una serie de programas, conjuntamente con todos los documentos asociados y la configuración de datos que se necesitan para que los programas operen de manera correcta.
 - Programas
 - ejecutables, de instalación, de reconfiguración, etc.
 - Documentación
 - requisitos, diseño, pruebas, código fuente, manuales de usuario, etc.
 - Datos
 - datos de la aplicación, parámetros de configuración, etc.

9

Ciencia de la Computación

- La ciencia de la computación se ocupa de las teorías y métodos subyacentes a las computadoras y los sistemas de software.
- La ingeniería de software se refiere a los problemas prácticos de producir software.
 - Los ingenieros de software requieren conocimiento de ciencias de la computación para realizar su labor, pero su objetivo es pragmático.



10

Definición (II)

- La ingeniería de software es la aplicación de la ciencia de la computación para la construcción de programas de software y todos sus documentos y datos asociados.
- Pero, ¿es así como se desarrolla habitualmente la práctica profesional? y ¿por qué sería conveniente que así fuera?

11

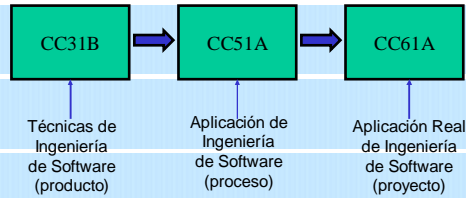
Otras Definiciones

- La Ingeniería de Software es el área de la ciencia de la computación que trata con la construcción de sistemas de software que son tan grandes o complejos que son construidos por un equipo o varios equipos de ingenieros. [Ghezzi91]
- Construcción de un software con múltiples versiones por múltiples personas. [Parnas87]

12

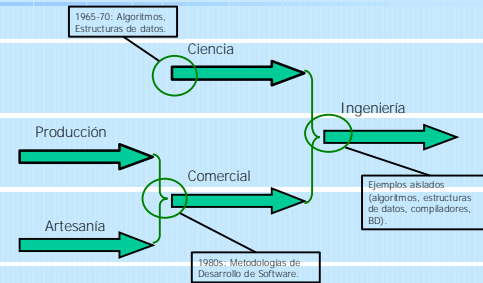
Producto y Proceso

- La IS trata de organizar un buen proceso de desarrollo para lograr un buen producto de software.



13

Evolución de la IS



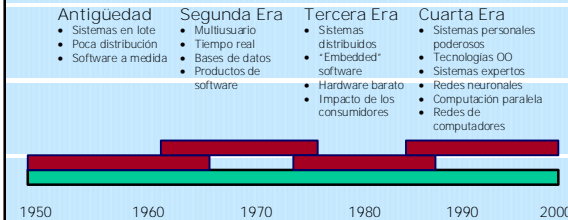
14

Historia

- 1950:
 - los programadores escribían programas.
- Principios de los '60:
 - expertos desarrollaron grandes proyectos.
- Fines de los '60:
 - aparecen los grandes sistemas comerciales de software,
 - el trabajo va más allá de la programación,
 - se crea el término "ingeniería de software".

15

Eras de la IS



16

Programación



- Programación:
 - un solo programador escribe todo el programa,
 - "programming in the small",
 - los programadores aprecian el sistema completo,
 - se requiere más tiempo para comunicación que para codificación,
 - cambios de personal afectan la productividad.

17

La Ingeniería de Software

- ¿Para qué la Ingeniería de Software?
 - ayuda a construir grandes sistemas de software,
 - ayuda a definir claramente los problemas,
 - "programming in the large",
 - administración, organización, herramientas, teorías, metodologías y técnicas como base para resolver los problemas computacionales.

18

Influencia de la IS

- Los costos de desarrollo de software crecen en comparación con los costos del hardware.
- Se requiere una producción de software más eficiente.
- La complejidad de los sistemas grandes de software hace que el desarrollo sea más que la codificación:
 - concepción, diseño, desarrollo, documentación, mantenimiento, evolución.
- La ingeniería de software continuará creciendo:
 - economía,
 - rol creciente del software en todos los ámbitos.

19

Rol del Ingeniero de Software

- Programador:
 - trabaja solo,
 - conocimiento de estructuras de datos y algoritmos,
 - hábil con varios lenguajes de programación.
- Ingeniero de software:
 - parte de un equipo de trabajo,
 - conoce estrategias de diseño,
 - traduce necesidades en especificaciones,
 - conocimiento de varias áreas de aplicación,
 - visión del sistema con distintos niveles de abstracción,
 - comunicación e interacción con otras personas.

20

Producto y Proceso



- El objetivo de la IS es desarrollar productos de software de calidad.
- Dos tipos de productos:
 - genéricos - desarrollados para vender masivamente.
 - a medida - desarrollados para un cliente particular.
- El desafío es desarrollar software de alta calidad con una cantidad finita de recursos y dentro del plazo planeado.
- El proceso de desarrollo tiene una incidencia importante en la calidad del producto.



La IS se ocupa tanto del producto como del proceso

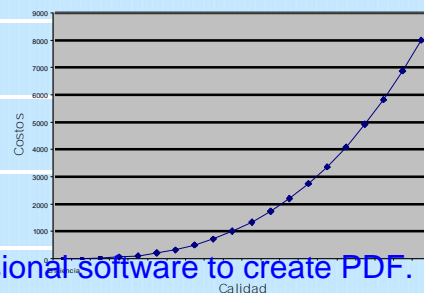
21

Proceso de Software

- Conjunto de actividades y resultados asociados que producen un producto de software.
- Actividades fundamentales:
 - especificación de requisitos,
 - diseño, codificación y test,
 - verificación y validación,
 - instalación y evolución.
- Distintos tipos de procesos pueden aplicarse para desarrollar un mismo producto.
- En general algunos procesos son más apropiados para desarrollar ciertos productos.

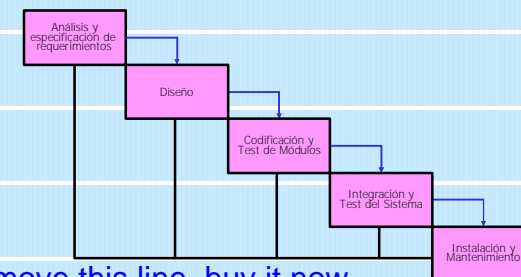
22

Más Calidad es Más Caro



23

Modelo de Cascada



24