

# Planificación y Gestión de Proyectos de Software

Pablo de la Fuente

Departamento de Informática

*pfuente@infor.uva.es*

Curso 2015-2016

# Horarios

- **Teoría:** lunes de 10:00 a 11:00 y miércoles de 9:00 a 10:00.
- **Laboratorios, seminarios, etc.:** Miércoles de 18:00 a 20:00.
- **Horario de tutorías:**  
Lunes de 11:00 a 13:00 y de 18:00 a 20:00 y miércoles de 11:00 a 13:00.

# Evaluación

- Existirá un examen de teoría que contará el 40% de dicha calificación.
- La valoración del trabajo práctico supondrá un 40% del total de la calificación final. Será necesaria la defensa de la práctica, con asistencia de todos los miembros del grupo.
- Un 20% corresponderá a la valoración de las actividades a realizar, de la participación de los alumnos en las discusiones planteadas a lo largo del curso y de las pruebas tipo test que se realizarán a lo largo del cuatrimestre (3 o 4).
- Para poder considerar superada tanto la parte teórica como la parte práctica será necesario haber alcanzado, al menos, un 4,5 en la calificación de cada prueba. En el caso de la parte práctica para la calificación se tendrán en cuenta, tanto los aspectos de funcionalidad, como la calidad de la documentación aportada y la defensa realizada de la práctica.

# Evaluación de la convocatoria extraordinaria

- Si la práctica ya está superada se realizará un examen con el mismo formato que en la convocatoria ordinaria. Si, por el contrario, la práctica no estuviera superada deberá volver a presentar la práctica (en una única entrega) y realizar un examen de teoría con el mismo formato que en la convocatoria ordinaria. En ambos casos las pruebas se considerarán con similar peso y fórmula de cálculo para la nota final que en la convocatoria ordinaria.

# Referencias Generales

- Sobre gestión de proyectos:  
*A Guide to the Project Management Body of Knowledge*. Project Management Institute, 2008  
*Software Project Management*. Bob Hughes and Mike Cotterell. McGraw-Hill, 2002  
*Managing Risk*. Elaine M. Hall. Addison-Wesley. 1998
- Sobre Proceso Unificado:  
*El proceso unificado de desarrollo de software*. Ivar Jacobson, Grady Booch, James Rumbaugh. Addison-Wesley 2000
- Sobre métricas:  
*Software Metrics: A rigorous & practical approach*. Norman E. Fenton. Shari Lawrence Pfleeger. International Thomson Computer Press, 1996  
*Metrics and Models in Software Quality Engineering*. Stephen H. Kan. Addison Wesley, 2003

En cada módulo se indicarán las referencias específicas.

# Definiciones de Ingeniería

DRAE. Proporciona dos definiciones para ingeniería. Como la segunda se refiere a la profesión o ejercicio del ingeniero, nos centramos en la primera. *Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía.*

Real Academia de Ciencias Exactas, Físicas y Naturales de España. *Conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones u otras realizaciones provechosas para el hombre.*

Javier Aracil en un trabajo de 1999<sup>1</sup> indica que la ingeniería hay que entenderla como *combinación de procesos tendentes a la solución de problemas*

Se diferencia de la ciencia, ya que se refiere a artefactos que se producen teniendo en cuenta criterios como efectividad, eficiencia y oportunidad.

---

<sup>1</sup> Javier Aracil. ¿Es menester que los ingenieros filosofen?. Argumentos de Razón Técnica. (2), 20-49, 1999. ▶

# Repaso de Conceptos

En ninguna de las definiciones anteriores se habla de ingeniería del software, quizá por que su uso es reciente. No obstante, la bibliografía en la que aparece el uso de dicho término no es nueva. Algunas definiciones son:

- Ingeniería de Software trata del establecimiento de los principios y métodos de la ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje con máquinas reales. Bauer 1972 (chairman of the *1968 NATO Software Engineering Conference*)
- Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada para desarrollar, operar y mantener dichos programas. Se conoce también como desarrollo de software o producción de software. Boehm 1976.
- Ingeniería de Software es el estudio de los principios y metodología para desarrollo y mantenimiento de sistemas software. Zelkovitz 1978.

# Repaso de Conceptos

- Ingeniería de Software es la profesión que crea y mantiene aplicaciones informáticas utilizando tecnologías y prácticas de informática (computer science), gestión de proyectos, ingeniería, dominios de aplicación y otros campos. Wikipedia 2005.
- Ingeniería de Software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software; es decir, la aplicación de la ingeniería al software. IEEE 1993. En el glosario también se considera la acepción “El estudio de las aproximaciones dadas anteriormente”.<sup>2</sup>
- La ingeniería de software es fundamentalmente una disciplina de ingeniería y como tal tiene un objetivo último: la entrega de artefactos que satisfagan las necesidades de aquellos que los usen. Clemens Szyperski, ICSE 2005

---

<sup>2</sup>En la wikipedia aparecen todas estas definiciones



# Problemas con el software

El software ha planteado siempre una serie de problemas, muchos de los cuales aún continúan abiertos:

- A pesar de los avances en el software, se está muy atrás respecto a las posibilidades del hardware.
- La demanda de nuevos productos sigue creciendo a un ritmo superior a la capacidad para producirlos.
- La utilización cada vez mas extensa de las computadoras ha hecho que la sociedad precise de un software fiable. Un fallo en determinado software puede producir considerables pérdidas en las empresas y, en algunos casos, hasta daño a las personas.
- Un objetivo fundamental es la construcción de software que tenga gran calidad y fiabilidad.
- Cuanto menos adecuado es el diseño del software mayor es el coste de mantener y mejorar los programas existentes.

# Problemas con el software

Desde hace unos años, el software, y no el hardware, es el elemento fundamental del coste en el desarrollo de sistemas informáticos. Por ello, y a pesar de los avances manifiestos que se han producido, los responsables se siguen planteando preguntas como:

- ¿Por qué lleva tanto tiempo terminar las aplicaciones?
- ¿Por qué es tan elevado su coste?
- ¿Por qué es tan difícil localizar todos los errores antes de entregar el producto al cliente?
- ¿Por qué es difícil para personas ajenas constatar el progreso del desarrollo a medida que este se va realizando?

Estas preguntas y otras que pueden surgir fácilmente son las que justifican la necesidad de aplicar técnicas de ingeniería en el desarrollo del software. Un dato: en 2012 fallaron el 18% de los proyectos y otro 43% superaron el tiempo o coste estimados.

# Problemas con el software. Algunos datos

Table 1			
Standish project benchmarks over the years			
Year	Successful (%)	Challenged (%)	Failed (%)
1994	16	53	31
1996	27	33	40
1998	26	46	28
2000	28	49	23
2004	29	53	18
2006	35	46	19
2009	32	44	24

Resolución Tipo 1, o proyecto con éxito. El proyecto ha finalizado, la relación entre las estimaciones de tiempo y coste y la realidad ha sido  $=1$ , y con respecto a la cantidad de funcionalidad conseguida es  $=1$ .

Resolución Tipo 2, o proyecto deficiente. El proyecto ha finalizado y está operativo pero la relación obtenida con respecto al tiempo y coste es  $< 1$  y para la cantidad de funcionalidad conseguida es  $> 1$

<sup>3</sup>Fuente: Standish Group, 2009

# Problemas con el software. Algunos datos

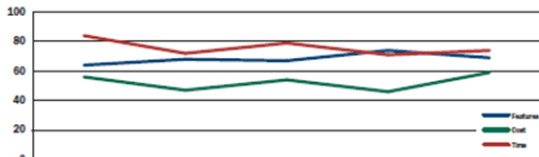
## RESOLUTION

	2004	2006	2008	2010	2012
Successful	29%	35%	32%	37%	39%
Failed	18%	19%	24%	21%	18%
Challenged	53%	46%	44%	42%	43%

Project resolution results from CHAOS research for years 2004 to 2012.

## OVERRUNS AND FEATURES

Time and cost overruns, plus percentage of features delivered from CHAOS research for the years 2004 to 2012.



	2004	2006	2008	2010	2012
TIME	84%	72%	79%	71%	74%
COST	56%	47%	54%	46%	59%
FEATURES	64%	68%	67%	74%	69%

# Características del software

El software presenta algunas características a considerar:

- El software se desarrolla, no se fabrica en el sentido clásico
- El software no se “estropea”
- Aunque la industria tradicional tiende a ensamblar componentes (actualmente servicios) la mayoría del software se construye a medida
- El cambio: *Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder, el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.* Kent Beck 2000.

# Proceso de desarrollo del software

## Aspectos del proceso del software:

- Aspectos básicos del proceso del software
- Aspectos de la ingeniería del software
- Aspectos de la gestión del proceso del software
- Aspectos de calidad en el proceso del software

# Marco General

La Guía para el Cuerpo de Conocimiento de Ingeniería de Software (SWEBOK) establece los objetivos siguientes:

- 1 Promover una visión consistente de la ingeniería de software en todo el mundo.
- 2 Clarificar la posición, y establecer los límites, de la ingeniería de software con respecto a otras disciplinas como ciencia de computadoras, gestión de proyectos, ingeniería de computadoras y matemáticas.
- 3 Caracterizar el contenido de la disciplina de Ingeniería de Software.
- 4 Proporcionar un acceso actualizado al Cuerpo de Conocimiento de Ingeniería de Software (SWEBOK).
- 5 Proporcionar un fundamento para el desarrollo curricular y la certificación particular.

# Objetivos del curso

Algunos de los objetivos del curso son:

- Conocer los roles específicos de una organización de software relacionados con la gestión de proyectos, garantía de calidad y gestión de procesos.
- Comprender las etapas fundamentales de la planificación y gestión de proyectos.
- Comprender las etapas fundamentales de la garantía de la calidad.
- Conocer los pasos fundamentales de gestión de procesos de software y la calidad de los mismos.



# Áreas del SWEBOK consideradas

Áreas especificadas en SWEBOK (SoftWare Engineering Body of Knowledge)

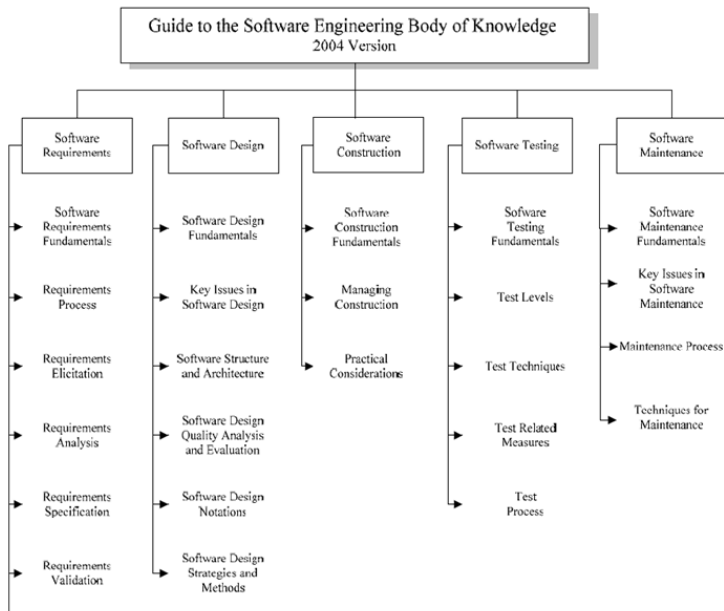
- Requisitos.
- Diseño.
- Construcción de software.
- Pruebas.
- Mantenimiento.
- ...

# Áreas del SWEBOK consideradas

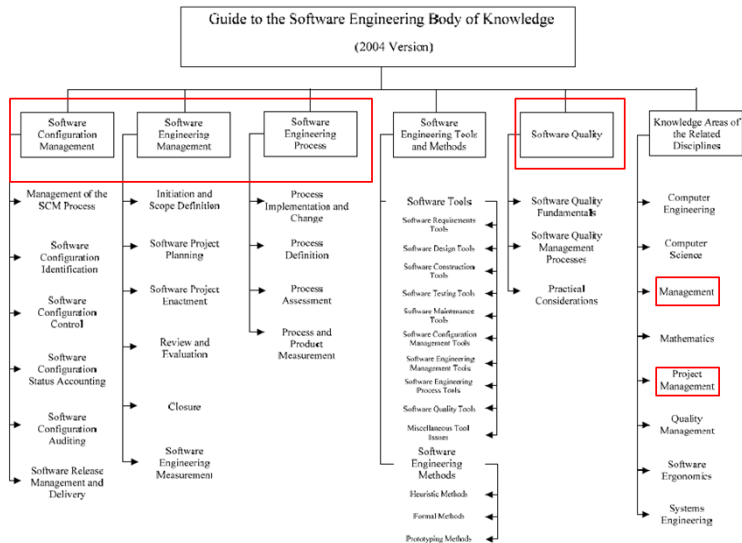
... continua

- Gestión de configuraciones.
- Gestión de ingeniería.
- Proceso de ingeniería.
- Herramientas y métodos de ingeniería.
- Calidad.

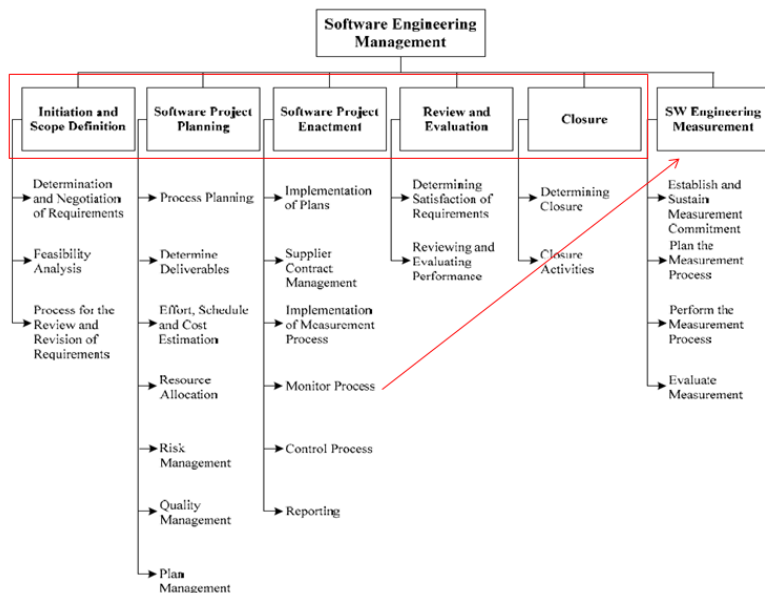
# Áreas del SWEBOK



# Áreas del SWEBOK



# Áreas del SWEBOK



## Áreas especificadas en el PMBOK (Project Management Body of Knowledge)

- Gestión de la Integración (del Proyecto)
- Gestión del Alcance
- Gestión del Tiempo
- Gestión de Costes
- Gestión de la Calidad
- Gestión de Recursos Humanos
- Gestión de la Comunicación
- Gestión de Riesgos
- Gestión de Adquisiciones
- Gestión de los Interesados

# Material adicional

SWEBOK, Software Engineering Body of Knowledge (2004) disponible desde IEEE Xplore en: <http://almena.uva.es>

Lista consolidada de referencias para la preparación del CSDP.

[http://www.computer.org/portal/web/swebok/consolidated\\_reference\\_list](http://www.computer.org/portal/web/swebok/consolidated_reference_list)

El borrador (*draft*) de la versión 3 del SWEBOK está en:

<http://www.computer.org/portal/web/swebok/v3guide/>

*Existen varias direcciones donde se puede localizar la versión 4 de PMBOK en pdf. No se proporciona una concreta porque a veces cambian de ubicación.*