

Tema 1: Introducción a la Ingeniería del Software

BLOQUE I: INTRODUCCIÓN Y PARADIGMAS DE DESARROLLO EN INGENIERÍA DEL SOFTWARE

Ingeniería del Software

Grado en Ingeniería Informática

Curso 2024/2025



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. Naturaleza y Problemas del Desarrollo de Software
4. La Ingeniería del Software
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. Naturaleza y Problemas del Desarrollo de Software
4. La Ingeniería del Software
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



¿Qué es el Software?

Software

Denominamos *software* a todo aquello intangible (no físico) que hay en un ordenador, incluyendo el conjunto de programas informáticos que indican la secuencia de instrucciones que un ordenador debe ejecutar durante su funcionamiento (también denominado código) y el resto de los datos que este ordenador manipula y almacena

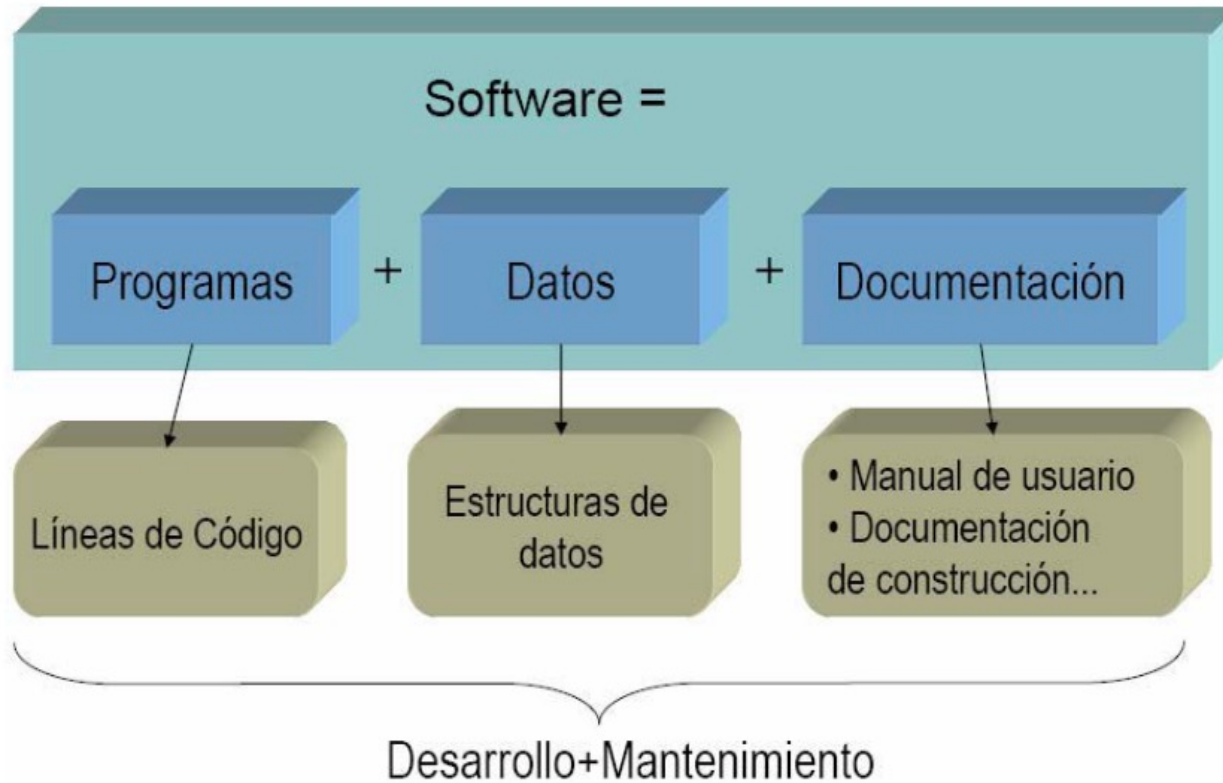
Hardware

Denominamos *hardware* al conjunto de componentes físicos de un ordenador. Este hardware ofrece una serie de instrucciones que el ordenador es capaz de ejecutar cuando ejecuta un programa

Fuente: J. Pradel i Miquel and J. A. R. Martos. Introducción a la ingeniería del software. Asignatura Ingeniería del Software. Universitat Oberta de Catalunya, 2010. FUOC PID_00171152.



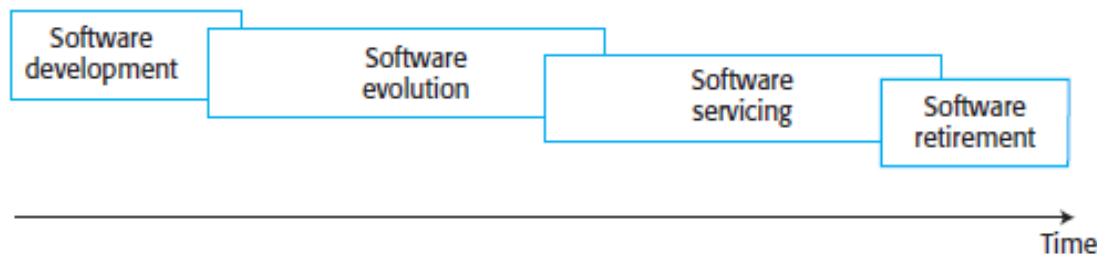
¿Qué es el Software?



¿Qué es el Software?

El software presenta una serie de peculiaridades:

- El producto software es enteramente conceptual
- No tiene propiedades físicas como peso, color o voltaje, y, en consecuencia, no está sujeto a leyes físicas o eléctricas
- Su naturaleza conceptual crea una distancia intelectual entre el software y el problema que el software resuelve
- Difícil para una persona que entiende el problema entender el sistema software que lo resuelve
- Para probar es necesario disponer de un sistema físico
- El mantenimiento no es sólo una substitución de componentes



Fuente: I. Sommerville, "Introduction to Software Engineering". Pearson. 2016.



¿Qué es el Software?

Categorías del software

- **Software de sistemas:** Dan servicio a otros programas, como los sistemas operativos o los compiladores. No tienen un propósito específico
- **Software de aplicación:** Son programas independientes que resuelven una necesidad específica, normalmente de una organización
- **Software científico y de ingeniería.** Muy enfocados al cálculo y a la simulación, se caracterizan por la utilización de algoritmos y modelos matemáticos complejos
- **Software empotrado:** Forma parte de algún dispositivo
- **Aplicaciones web:** Se caracterizan por unificar fuentes de datos y diferentes servicios en entornos altamente distribuidos
- **Software de inteligencia artificial:** Es un programa que realiza tareas que requieren inteligencia humana, como el aprendizaje, la toma de decisiones y el procesamiento del lenguaje



Índice

1. ¿Qué es el Software?
2. **Evolución Histórica**
3. Naturaleza y Problemas del Desarrollo de Software
4. La Ingeniería del Software
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



Evolución histórica

- Décadas 60-70:
 - Pocas computadoras:
 - Grandes computadoras o *mainframes*
 - Muy caros
 - Poca variedad de software → muy específico y desarrollado por el fabricante.
 - Se desarrolla software artesanal:
 - El negocio estaba en el hardware
 - Se dispone del código fuente y los desarrolladores de software compartían libremente sus programas unos con otros con ánimo constructivo
 - El desarrollo de software no se gestionaba según una planificación y era prácticamente imposible predecir los costes y el tiempo de desarrollo.
 - El software era un producto gratuito que se incluía al comprar hardware
 - No existía el concepto de software empaquetado, como producto



Evolución histórica

- Década de los 80:
 - Desarrollo de la microelectrónica
 - Mayor potencia de cálculo y reducción de costes
 - Aparecen lenguajes con nociones de programación estructurada
 - Se comienza a dar importancia al diseño (herramientas CASE) y los costes
 - Aparece el movimiento del software libre

AUMENTAN los problemas del desarrollo de software:

- Sobreexplotación del potencial del hardware
- Incapacidad de atender a la demanda
- Incapacidad de mantener el software existente



Evolución histórica

- Década de los 90:
 - Se impone el desarrollo orientado a objetos (nace Java)
 - Creciente atención a la arquitectura del software (componentes)
 - Creciente uso de arquitecturas distribuidas (cliente/servidor)
 - Primeros sistemas de control de versiones (CVS)
- Década de los 2000:
 - Aparición de los entornos de desarrollo integrados (IDEs)
 - Aunque se impone UML, decaen las herramientas CASE
 - Nuevos lenguajes de programación: C#, Scala, Go...
- Década de 2010:
 - Auge de los sistemas web y el software como servicio
 - Metodologías ágiles de desarrollo
 - Irrumpe la industria móvil
 - Crecimiento de los sistemas empresariales y gubernamentales



Evolución histórica

La ingeniería del software en la actualidad:

- Heterogeneidad de sistemas: web, apps, sistemas ciber-físicos, videojuegos, etc.
- Volatilidad de las tecnologías (mayor competencia)
- La reutilización cobra especial importancia
- Evolución constante de los lenguajes (se han creado cerca de 9000)
- Escalabilidad y disponibilidad global
- Herramientas de soporte y automatización en muchas de sus fases:
 - *Frameworks* de desarrollo
 - Desarrollo continuo: CI/CD (*Continuous integration / Continuous deployment*)
 - Automatización de pruebas (*Selenium, Cypress*)
 - Asistentes inteligentes de programación (*Copilot, Ponicode*)

<https://sourceforge.net/software/ai-coding-assistants/>



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. **Naturaleza y Problemas del Desarrollo de Software**
4. La Ingeniería del Software
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



Naturaleza y Problemas del Desarrollo de Software

- El software como elemento lógico
 - Se desarrolla, no se fabrica:
 - Calidad del diseño
 - Costes más importantes en la ingeniería
 - Gestión especial de los proyectos
 - Se “deteriora” con el mantenimiento
 - En los comienzos era un desarrollo a medida
- La “crisis” del software: *problemas que aparecen en el desarrollo del software al desarrollar, mantener y atender la demanda de nuevas aplicaciones.*



Naturaleza y Problemas del Desarrollo de Software

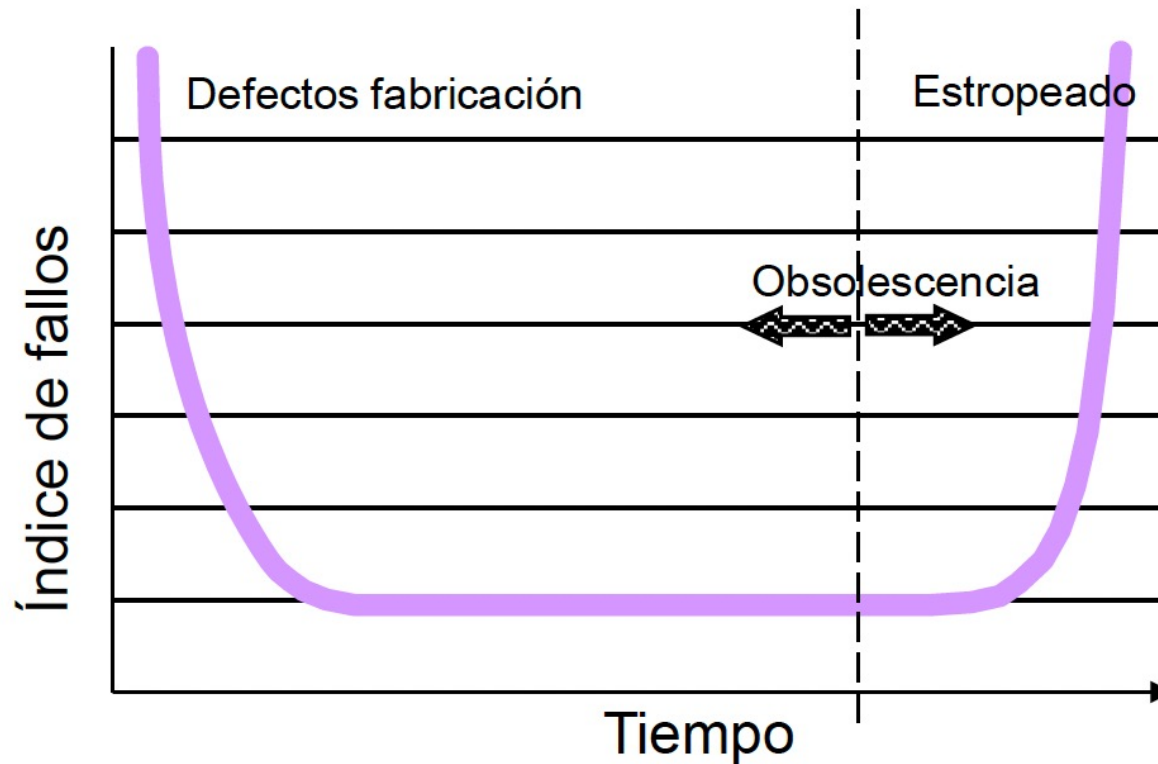
Crisis del software

- Se identificó por primera vez en 1968
- Síntomas de crisis en el desarrollo de software
 - **Productividad:** de los desarrolladores baja en relación a la demanda
 - **Expectativas:** los sistemas no responden a las expectativas
 - **Fiabilidad:** Los programas fallan a menudo
 - **Calidad:** No es adecuada
 - **Costes:** Difíciles de predecir, a menudo sobrepasan lo esperado
 - **Mantenimiento:** Modificación del software costosa y compleja
 - **Plazos:** no se cumplen
 - **Portabilidad:** Difícil cambiar de plataforma
 - **Eficiencia:** No hay aprovechamiento óptimo de recursos
- Consecuencias:
 - Baja productividad
 - Baja calidad
- Solución:
 - Aplicar principios de ingeniería a la construcción de sistemas informáticos



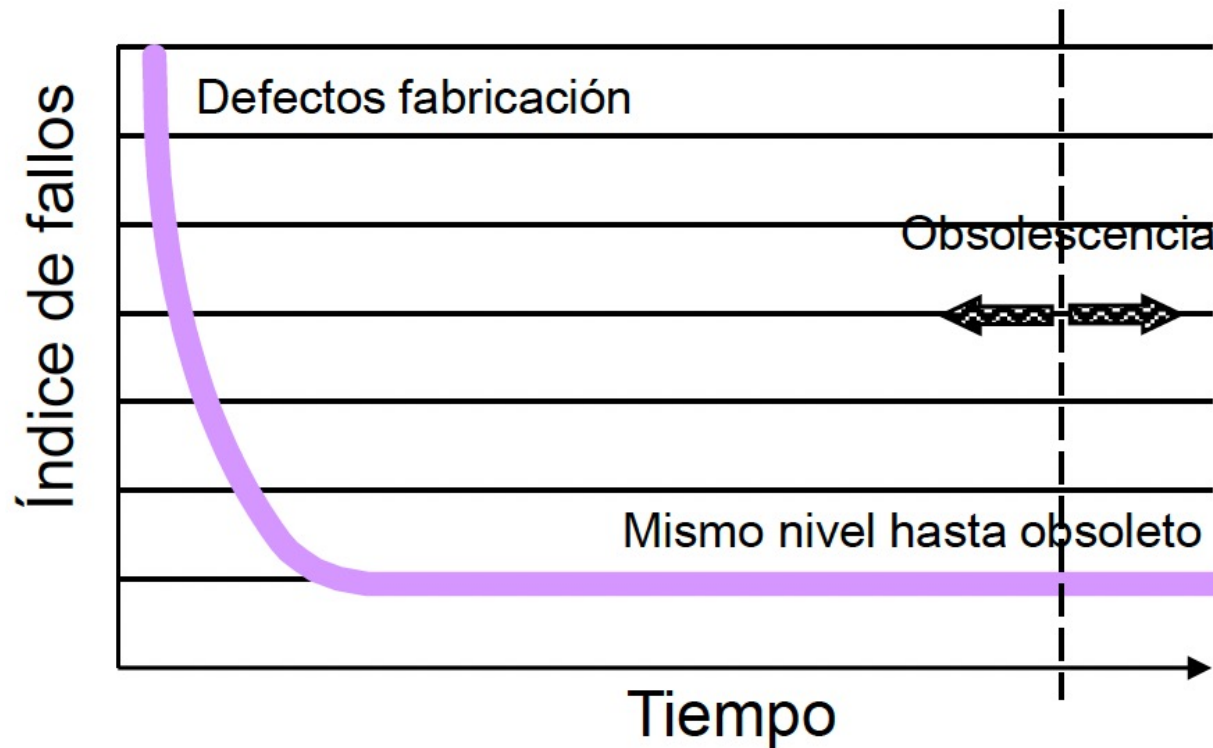
Naturaleza y Problemas del Desarrollo de Software

Curva de fallos del Hardware



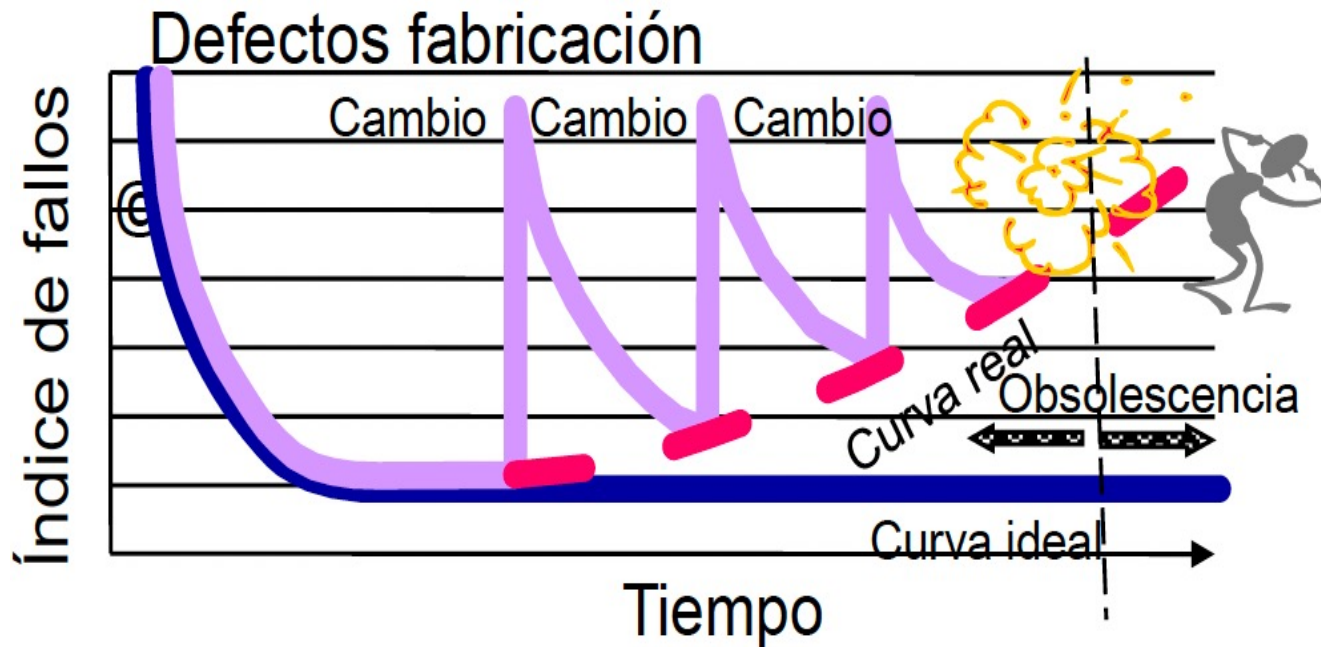
Naturaleza y Problemas del Desarrollo de Software

Curva de fallos del Software



Naturaleza y Problemas del Desarrollo de Software

Curva real de fallos del Software



Naturaleza y Problemas del Desarrollo de Software

Mitos del software

MITOS DE LOS DESARROLLADORES

- Programa funcionando = fin del trabajo
- Calidad = el programa se ejecuta sin errores
- Entrega al cliente = programa funcionando

MITOS DE GESTIÓN

- Uso de estándares
- Uso de herramientas
- Mala planificación: aumento de programadores

MITOS DEL CLIENTE

- Requisitos establecidos como una declaración general de objetivos
- Flexibilidad del software ante los cambios



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. Naturaleza y Problemas del Desarrollo de Software
- 4. La Ingeniería del Software**
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



La Ingeniería del Software

Definición original

“Establecimiento y uso de principios de ingeniería para obtener software económico que trabaje de forma eficiente en máquinas reales”.

Fritz Bayer, 1968 (conferencia OTAN)

Otras definiciones

“Disciplina para producir software de calidad desarrollado sobre las agendas y costes previstos y satisfaciendo los requisitos”.

S. Schach 1990, Software Engineering

“(1) La aplicación de métodos sistemáticos, disciplinados y cuantificables para el desarrollo, operación y mantenimiento de software; esto es, la aplicación de la ingeniería al software. (2) El estudio de (1)”.

IEEE 1993

“Disciplina de ingeniería que se ocupa de todos los aspectos de la producción de software, desde las primeras fases de especificación del sistema hasta el mantenimiento del sistema después de su uso”.

I. Sommerville. “Introduction to Software Engineering”. 10th ed. 2016.



La Ingeniería del Software

Aspectos esenciales

- Establecimiento y uso de principios de ingeniería robustos, orientados a obtener software económico, fiable, eficiente y que satisfaga las necesidades del usuario
- Disciplina que comprende todos los aspectos de la producción de software, desde las etapas iniciales hasta el mantenimiento:
- No solo es programar... Engloba:
 - ✓ Planificar, modelar, diseñar, probar, validar, corregir, mejorar, desplegar...
 - ✓ Orientación a la solución de problemas
 - ✓ Adquisición de conocimientos



La Ingeniería del Software

- Desde 1968 hasta la fecha han sido muchos los esfuerzos realizados por los departamentos de informática de las universidades, y por organismos de estandarización para identificar las causas del problema y definir pautas estándar para la producción y mantenimiento del software
- Los esfuerzos se han encaminado en tres direcciones principales:
 - Identificación de los factores clave que determinan la calidad del software
 - Identificación de los procesos necesarios para producir y mantener software
 - Acotación, estructuración y desarrollo de la base de conocimiento necesaria para la producción y mantenimiento de software
- El resultado ha sido la **necesidad de profesionalizar el desarrollo, mantenimiento y operación de los sistemas de software**



La Ingeniería del Software



International
Organization for
Standardization

- Organización Internacional para la Estandarización. Fundada en 1947.
- Son miembros 172 países.
- En 1987 la Organización Internacional para la Estandarización (ISO) y la Comisión Internacional Electrotécnica (IEC), establecieron un Comité Internacional (JTC1) para las Tecnologías de la Información con la misión de “estandarización en el campo de los sistemas de tecnologías de la información”.



La Ingeniería del Software



- IEEE es el Instituto de Ingenieros en Electricidad y Electrónica (Institute of Electrical and Electronics Engineers).
- Su misión es preservar, investigar y promover la información de las tecnologías eléctricas y electrónicas.
- Surgió en 1963 con la fusión del AIEE (Instituto Americano de Ingenieros Eléctricos) y el Instituto de Ingenieros de Radio (IRE).
- La IEEE Computer Society (www.computer.org) está formada en la actualidad por más de 100.000 miembros en todo el mundo
- Su finalidad es avanzar en la teoría, práctica y aplicación de las tecnologías de la información. Realiza conferencias, publicaciones, cursos de formación, y desarrolla estándares
- IEEE ha desarrollado estándares para todas las áreas de Ingeniería del Software



La Ingeniería del Software



Software Engineering Institute

- Instituto de Ingeniería del Software (SEI <http://www.sei.cmu.edu/>)
- Integrado en la Universidad Carnegie Mellon
- Los trabajos y aportaciones realizadas por el SEI son también referente mundial de primer orden
- Su aportación más significativa son los modelos de madurez de las capacidades: **CMM** y **CMMI**



La Ingeniería del Software

Definirse a sí misma: ¿Cuáles son las áreas de conocimiento que la comprenden?



SWEBOK: Software Engineering Body of knowledge

Definir los procesos que intervienen en el desarrollo, mantenimiento y operación del software



ISO/IEC 12207: Procesos del ciclo de vida del software

De las mejores prácticas, extraer modelos de cómo ejecutar esos procesos para evitar los problemas de la “crisis del software”



CMM / CMMI
ISO/IEC TR 15504

Definir estándares menores para dibujar criterios unificadores en requisitos, pruebas, gestión de la configuración, etc.



IEEE 830 – IEEE 1362 – ISO/IEC 14764 ...



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. Naturaleza y Problemas del Desarrollo de Software
4. La Ingeniería del Software
5. **Conceptos de la Ingeniería del Software**
6. Principios de la Ingeniería del Software



Conceptos de la Ingeniería del Software

- **Papel o Rol**
 - Conjunto de responsabilidades en el proyecto o en el sistema
 - Asociado con un conjunto de tareas y se asigna a un participante
 - Un mismo participante puede cumplir varios papeles
- **Participantes:** todas las personas involucradas en el proyecto
 - **Cliente:** encarga y paga el sistema
 - **Equipo de desarrollo:** construyen el sistema (analistas de negocio, arquitectos software, programadores, probadores...)
 - **Gerente o director del proyecto:** planifica y calcula el presupuesto, coordina a los desarrolladores y cliente
 - **Usuarios finales:** los que van a utilizar el sistema
- ¿Qué hace un ingeniero de software?
- ¿Qué habilidades necesita un ingeniero de software?



Conceptos de la Ingeniería del Software

- **Sistemas y modelos**
 - Sistema: realidad subyacente
 - Modelo: cualquier abstracción de la realidad
- **Productos de trabajo o Entregable**
 - Elemento que se produce durante el desarrollo
 - Dos tipos:
 - Producto de trabajo interno: producto para el consumo interno del proyecto
 - Entrega: producto de trabajo para un cliente
- **Actividad** (o fase): conjunto de tareas que se realiza con un propósito específico que pueden componerse de otras actividades
- **Tarea**: unidad elemental de trabajo que puede ser administrada
- **Recursos**: bienes que se utilizan para realizar el trabajo:
 - Tiempo, equipamiento y recursos humanos
 - Al planificar, el gerente divide el trabajo en tareas y les asigna recursos



Conceptos de la Ingeniería del Software

- **Objetivos**

- Principios de alto nivel que se utilizan para guiar el proyecto
- Definen los atributos realmente importantes del sistema
- A veces hay conflicto entre objetivos que aumentan la complejidad del proyecto

- **Requerimientos**

- Características que debe tener el sistema
- **Requerimiento funcional:** área de funcionalidad que debe soportar el sistema
- **Requerimiento no funcional:** restricción que se establece sobre el funcionamiento del sistema

- **Otras restricciones:** por ejemplo, utilización de un determinado lenguaje, de una determinada plataforma o de un sistema antiguo que el cliente no quiere retirar

- **Notación:** conjunto de reglas gráficas o de texto para representar un modelo (UML, *Unified Modelling Language*, es una notación gráfica orientada a objetos para representar modelos)

- **Método:** técnica repetible para resolver un problema específico.

- **Metodología:** colección de métodos para la resolución de una clase de problemas



Índice

1. ¿Qué es el Software?
2. Evolución Histórica
3. Naturaleza y Problemas del Desarrollo de Software
4. La Ingeniería del Software
5. Conceptos de la Ingeniería del Software
6. Principios de la Ingeniería del Software



Principios de la Ingeniería del Software

- Haz de la calidad la razón de trabajar
- Una buena gestión es más importante que una buena tecnología
- Las personas y el tiempo no son intercambiables
- Seleccionar el modelo de ciclo de vida adecuado
- Entregar productos al usuario lo más pronto posible
- Determinar y acotar el problema antes de escribir los requisitos
- Realizar un diseño
- Documentar
- Las técnicas son anteriores a las herramientas
- Primero hazlo correcto, luego hazlo rápido
- Probar, probar y probar (incluye inspecciones)
- Asume responsabilidades
- La gente es la clave del éxito
- Introducir las mejoras y modificaciones con cuidado



Principios de la Ingeniería del Software

- Nunca dejes que tu jefe o cliente te convenza para hacer mal un trabajo
- La gente necesita sentir que su trabajo es apreciado
- La educación continua es responsabilidad de cada miembro del equipo
- El compromiso del cliente es el factor más crítico en la calidad del software
- Tu mejor desafío es compartir la visión del producto con el cliente
- La mejora continua de tu proceso de desarrollo de software es posible y esencial
- La calidad es el principal objetivo; la productividad a largo plazo es una consecuencia de alta calidad
- Haz que los errores los encuentre un colaborador y no un cliente
- Una clave en la calidad en el desarrollo de software es realizar iteraciones en todas las fases de desarrollo
- La gestión de errores y solicitud de cambios es esencial para controlar calidad y el mantenimiento
- Haz lo que tenga sentido, no recurras a los dogmas
- No se puede cambiar todo de una vez. Identifica los cambios que se traduzcan en los mayores beneficios, y comienza a implementarlos

