

¿Qué es el software?

□ Definición:



- ▣ Programas de cómputo y documentación asociada.

Sommerville

□ Definición:

- ▣ instrucciones de ordenador que cuando se ejecutan proporcionan la función y el rendimiento deseado,
- ▣ estructuras de datos que facilitan a los programas manipular adecuadamente la información
- ▣ y los documentos que describen la operación y el uso de los programas.

Pressman

¿Qué es la Ingeniería del Software?

- Una disciplina de la ingeniería que concierne a todos los aspectos de la PRODUCCIÓN DEL SOFTWARE desde su inicio hasta que se sustituye.
- Ingeniería: Aplicación de Métodos, Herramientas, Técnicas, etc., de forma sistemática y organizada
- Todos los aspectos de producción:
 - aspectos técnicos,
 - de gestión de RRHH,
 - de gestión del tiempo, coste, calidad, etc.



Atributos del buen software



▣ Mantenibles

Sommerville.

- Debe ser posible que el software evolucione y siga cumpliendo sus especificaciones

▣ Confiables y seguros

- El software no debe causar daños físicos ni económicos en caso de fallo. Fiabilidad, Seguridad y protección.

▣ Eficientes

- El software no debe desperdiciar los recursos del sistema

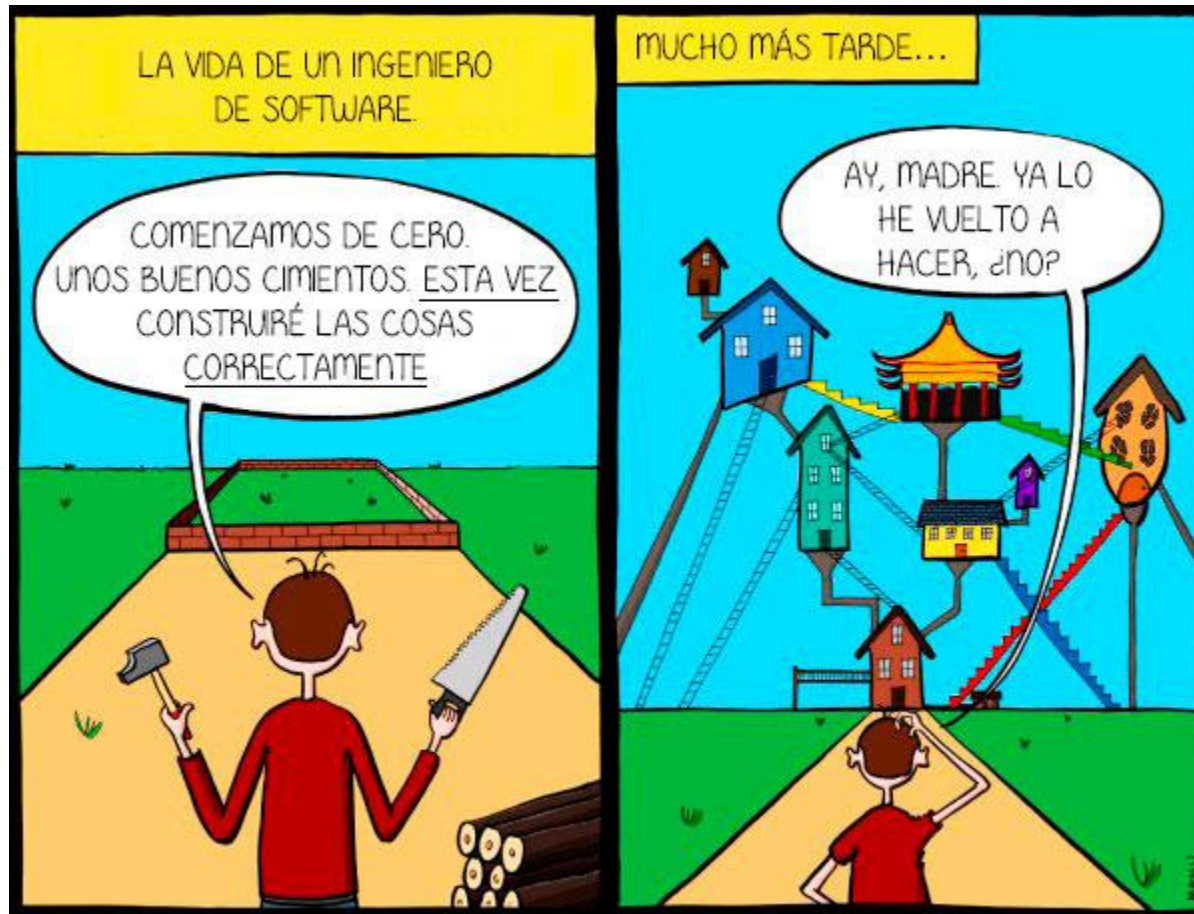
▣ Usables

- Debe contar con Interfaces y documentación adecuadas

Ingeniería vs Preguntas sin resolver

- ¿Por qué lleva tanto tiempo terminar los programas?
 - ▣ Planificación (EDT, Gantt, etc.)
- ¿Por qué es tan elevado el coste?
 - ▣ Construcción a medida
 - ▣ Construcción desde cero.
 - ▣ Estimaciones
- ¿Por qué no es posible encontrar todos los errores antes de entregar el software al cliente?
 - ▣ Planes de Pruebas
- ¿Por qué resulta tan difícil constatar el progreso conforme se desarrolla el software?
 - ▣ Elemento lógico
 - ▣ Procesos
 - ▣ Ciclos de vida

Ingeniería vs Preguntas sin resolver



Ingeniería del software.

¿Qué es el proceso de desarrollo del Software?

- ▣ La secuencia de actividades que conducen a la elaboración de un producto software.
- ▣ Fundamentalmente, pero no sólo:
 - Especificación
 - Desarrollo
 - Validación
 - Evolución vs Mantenimiento

Características del software

- El software se desarrolla, no se fabrica en un sentido clásico

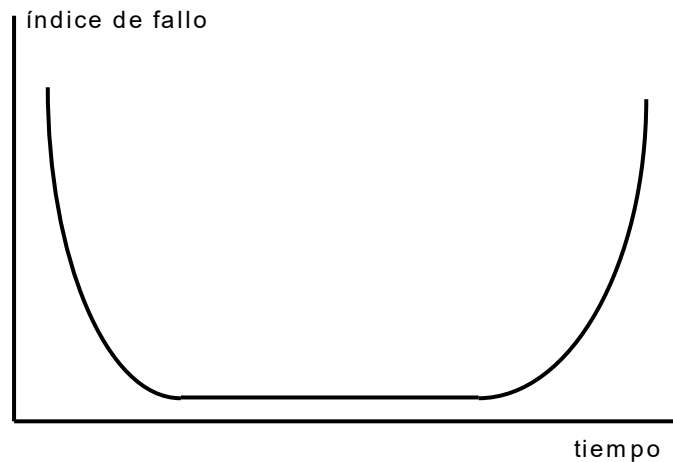
Tabla 1.1. Influencia de los costes de ingeniería en el coste total del producto

	Ingeniería	Producción o Desarrollo	Coste unitario / 100 unidades	Coste unitario / 100.000 unidades
Hardware	1000	50 c.u.	60	50.01
Software	1000	2000	30	0.03

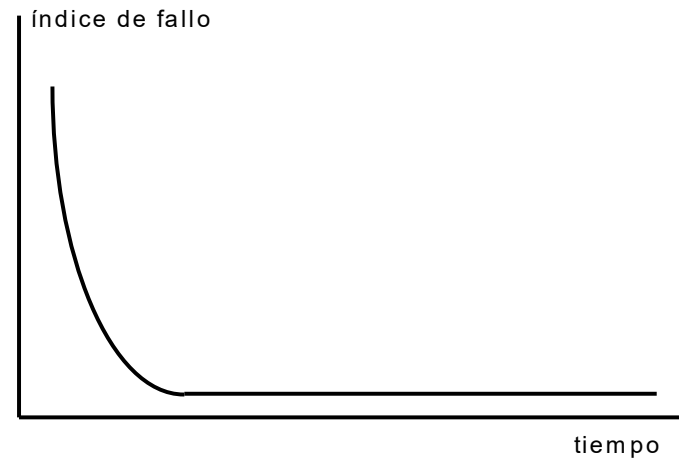
Características del software

- ❑ El software no se fabrica, se desarrolla
- ❑ El software no se estropea

Curva de fallos del hardware



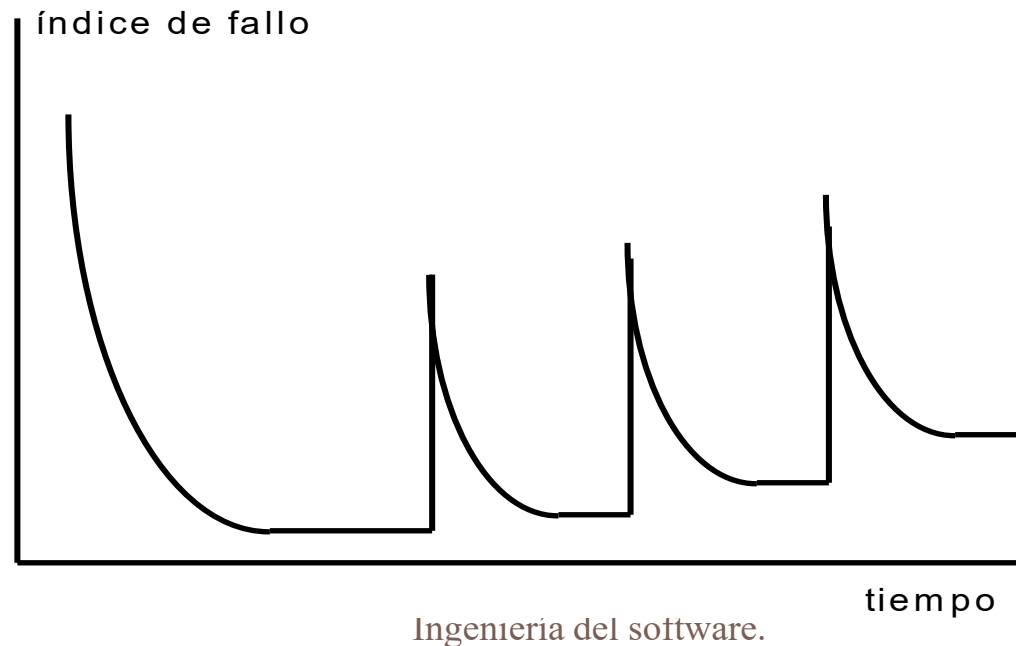
Curva ideal de fallos del software



Características del software

- El software no se fabrica, se desarrolla
- El software no se estropea, se degrada

Curva real de fallos del Software



Características del software

- El software no se fabrica, se desarrolla
- El software no se estropea, se degrada
- Software a medida, o personalizado.
 - ▣ Productos Genéricos: Producidos por una organización para introducir en el mercado
 - ▣ Productos a Medida: Desarrollados bajo pedido
 - ▣ La mayor parte del gasto es en productos genéricos pero hay más esfuerzo en el desarrollo de los productos a medida

Características del software

- Reutilización. Desarrollo por componentes
 - ▣ Posibilidades actuales: Activos o artefactos
 - Reutilización a través de Bibliotecas (Matemáticas, persistencia...)
 - Módulos o Componentes.
 - Integrar, adaptar, envolver
 - Tuning de soluciones genéricas a problemas particulares (ERPs). Configuración de frameworks.
 - Patrones de diseño (FACADE, MVC, etc.)
 - Requisitos???,
 - ▣ Desarrollo basado en componentes

Características del software

□ Reutilización.

▣ Ventajas

- Reduce los costes
- Aumenta productividad
- Aumenta la calidad
- Mejora mantenimiento y soporte
- Mejora control y planificación

▣ Limitaciones

Características del software

□ Reutilización.

▣ Ventajas

- Reduce los costes
- Aumenta productividad
- Aumenta la calidad
- Mejora mantenimiento y soporte
- Mejora control y planificación

▣ Limitaciones

- Complejos vs óptimos
- Inversión inicial
- Reestructuración
Organización y proceso
- Recuperación de activos
 - Concepto
 - Contenido
 - Contexto

Características del software

□ Reutilización.

▣ Estrategia

- Ver qué es reutilizable.
- Construir los activos.
- Difundir la información.
- Obligar al reuso enseñando a operar los activos.

▣ Futuro (Presente?)

- Búsqueda de activos de cualquier tipo en un repositorio (en la nube?)

El software heredado

□ Características

- Es longevo.
- Es crítico para las empresas que lo usan
- Ha sido modificado de forma continuada
 - Debe adaptarse a nuevos entornos.
 - Nuevo hardware
 - La red
 - Nuevos sistemas (BDs)
 - Debe implementar nuevos requerimientos

□ Problemas: poca calidad

- Historia de cambios manejada con pobreza
- No está documentado
- Pruebas sin documentar
- Diseños complejos
- Código oscuro

□ Consecuencias

- Costoso en mantenimiento
- Su evolución tiene un altísimo riesgo para la empresa.

□ Evolución del software.

Evolución del software

□ Evolución de grandes sistemas soft. Leyes de Lehman

(Estudio referido a finales del siglo XX)

- El cambio es continuo
- Crecimiento continuo
 - nuevas funcionalidades continuamente
- La complejidad es creciente
 - Costes extra para simplificación
- Estabilidad organizacional
 - Velocidad de trabajo invariante
- Conservación de la familiaridad:
 - Cambios incrementales constantes, independientes de los recursos
- Decremento de la calidad

Aplicaciones del software

□ Tipos de problema

- ▣ Problemas con solución por pasos específicos
 - Algoritmo, con lenguajes de prog. Procedimentales
- ▣ Problemas que pueden describirse formalmente
 - Lenguajes declarativos. SQL
- ▣ Problemas basados en conocimiento heurístico
 - Sistemas expertos, basados en la ejecución de reglas
- ▣ Problemas que no sabemos como se resuelven, pero conocemos algunas soluciones concretas
 - Utilizaremos redes neuronales

Aplicaciones del software

- Clasificación por Categoría (Pressman).
 - ▣ S. empotrado: lavadoras, hornos, coches, etc.
 - ▣ S. de sistemas: Sistemas operativos, compiladores,...
 - ▣ S. científico y de ingeniería: Cálculo numérico
 - ▣ S. de aplicación: Software a medida.
 - ▣ S. de línea de productos: Office, juegos, etc.
 - ▣ S. de Inteligencia Artificial
 - ▣ Aplicaciones basadas en Web, o en la nube.

Otras Clasificaciones del software

(Sommerville)

- Por su estructura:
 - ▣ Orientados a función.
 - ▣ Orientados a componentes.
 - ▣ Orientados a listas.
 - ▣ Orientados a objetos.
- Por su función:
 - ▣ Programas o Sistemas de Usuario
 - ▣ Interfaces Hombre-Maquina.
 - ▣ Herramientas de Software.
 - ▣ Librerías.
 - ▣ Sistemas de uso genérico:
Compiladores, S.O's,
Procesadores de Texto, etc.
 - ▣ Bases de Datos.
 - ▣ Sistemas basados en Web.
- Por su plataforma de computo:
 - ▣ Sistemas embebidos.
 - ▣ Sistemas de computo distribuido.
 - ▣ Sistemas de computo paralelo.
 - ▣ Sistemas de tiempo real.
 - ▣ Sistemas basados en Chips.
 - ▣ Wearable computing systems.
 - ▣ Sistemas de computo ubicuos.

Problemas del desarrollo de software

- Planificación y estimación de costes, imprecisos
- Productividad baja: duración mayor que la esperada
- Mala calidad a la entrega del producto
- Cliente insatisfecho
 - ⇒ Rediseño del producto



Problemas del desarrollo de software

- Planificación y estimación de costes, imprecisos
 - ▣ No hay estudios de realizaciones previas
 - ▣ CMMI: 1ª Norma: Medir.
 - ▣ Los responsables de proyecto pueden no ser expertos informáticos
 - ▣ Los expertos en informática dirigen proyectos sin conocimientos de gestión.
- Productividad baja: duración mayor que la esperada
- Mala calidad a la entrega del producto
- Cliente insatisfecho \Rightarrow Rediseño del producto

Problemas del desarrollo de software

- Planificación y estimación de costes, imprecisos
- Productividad baja: duración mayor que la esperada
 - ▣ Especificaciones ambiguas o incorrectas
 - ▣ Muchas modificaciones sobre la marcha
 - ▣ Falta de documentación
 - ▣ Soluciones:
 - ▣ Administración de requisitos
 - ▣ Metodologías ágiles
 - ▣ Gestión de la configuración \Rightarrow G. del cambio.

Problemas del desarrollo de software

- Planificación y estimación de costes, imprecisos
- Productividad baja: duración mayor que la esperada
- Mala calidad a la entrega del producto
 - Aseguramiento de la calidad
 - Procesos de Verificación y Validación
- Cliente insatisfecho → Rediseño del producto

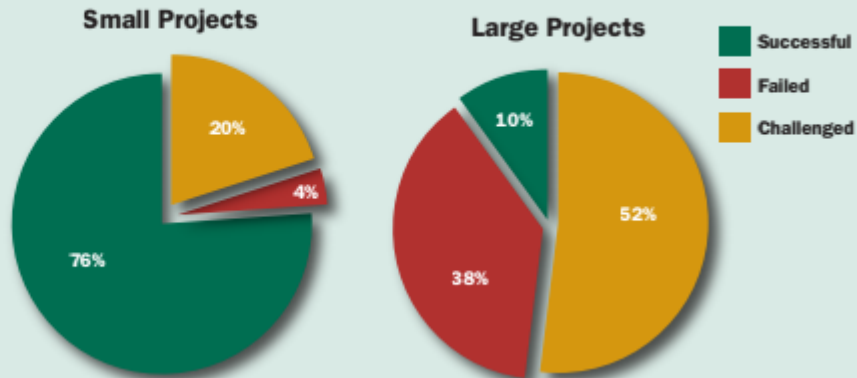
Problemas del desarrollo de software

□ CHAOS Report (Standish Group)

	1994	1996	1998	2000	2002	2004	2011	2012	2013	2014	2015
Éxito	16%	27%	26%	28%	34%	29%	29%	27%	31%	28%	29%
Comprometidos	53%	33%	46%	49%	51%	53%	49%	56%	50%	55%	52%
Cancelados	31%	40%	28%	23%	15%	18%	22%	17%	19%	17%	19%

CHAOS RESOLUTION BY LARGE AND SMALL PROJECTS

Project resolution for the calendar year 2012 in the new CHAOS database. Small projects are defined as projects with less than \$1 million in labor content and large projects are considered projects with more than \$10 million in labor content.



Problemas del desarrollo de software

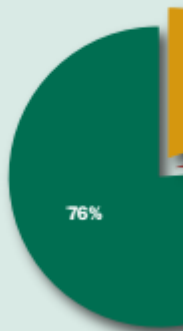
□ CHAOS Report 2013 (Standish Group)

	1994	1996	1998	2000	2002	2004	2011	2012	2013	2014	2015	2017
Éxito	16%	27%	26%	28%	34%	29%	29%	27%	31%	28%	29%	33%
Comprometidos	53%	33%	46%	49%	51%	53%	49%	56%	50%	55%	52%	48%
Cancelados	31%	40%	28%	23%	15%	18%	22%	17%	19%	17%	19%	19%

CHAOS RESOLUTION BY LARGE AND SMALL PROJECTS

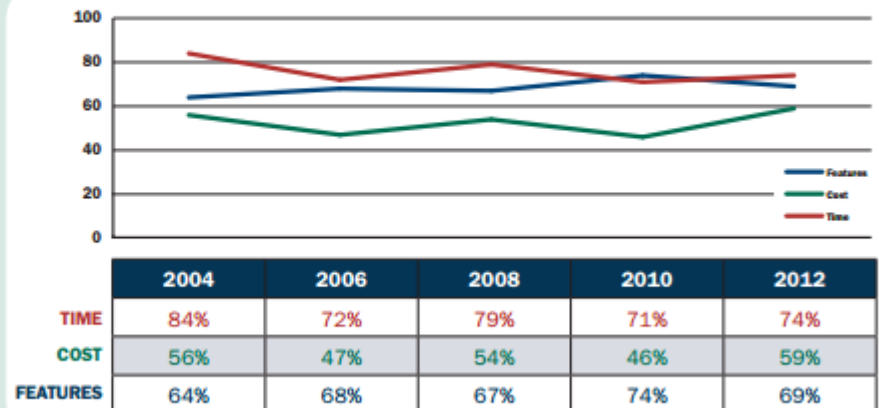
Project resolution for the calendar year 2012 in the new CHAOS database. Small projects are defined as projects with less than \$1 million in labor content and large projects are considered projects with more than \$10 million in labor content.

Small Projects



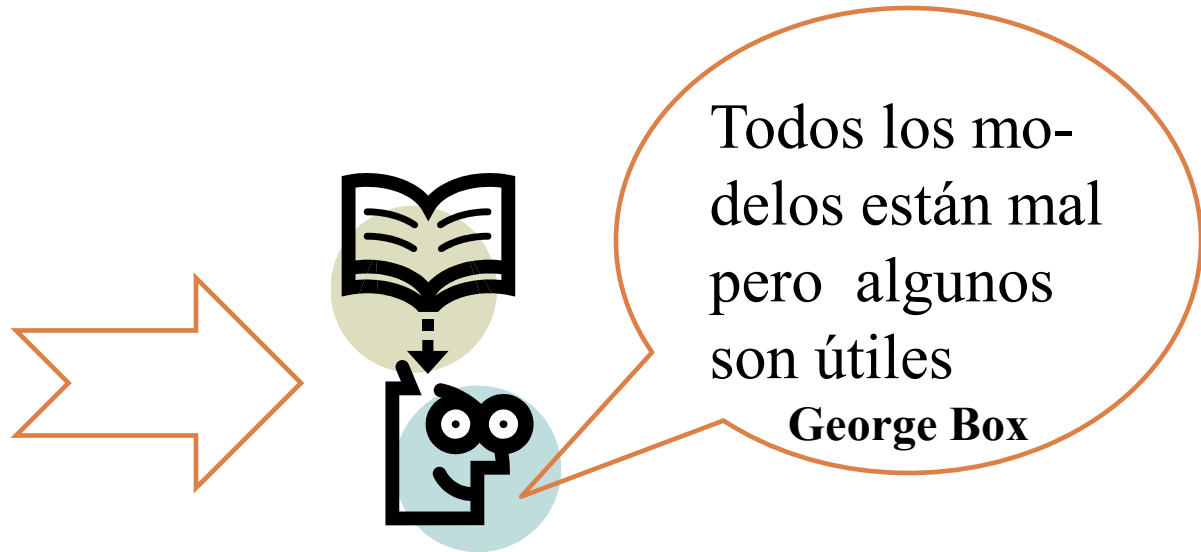
OVERRUNS AND FEATURES

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



Mitos del software

- Mitos de la administración
 - ▣ Ya se tiene un libro lleno de estándares y procedimientos para la construcción del software.



Ingeniería del software.

Mitos del software

- Mitos de la administración
 - ▣ Ya se tiene un libro lleno de estándares y procedimientos para la construcción del software.
 - ▣ Si un proyecto se atrasa sólo tengo que añadir personal
 - ▣ Si subcontrato un proyecto de software puedo despreocuparme y esperar a que se construya

Mitos del software

□ Mitos del Cliente

- ▣ Un enunciado general de los objetivos es suficiente para comenzar a escribir programas
- ▣ Podemos cambiar continuamente los requerimientos del programa porque el software es flexible y se adapta

Mitos del software

□ Mitos del desarrollador

- ▣ Una vez el programa ha sido escrito y puesto a funcionar el trabajo está terminado
- ▣ Mientras el programa no se esté ejecutando, no existe forma de evaluar su calidad
- ▣ El único producto del trabajo que puede entregarse para que un proyecto se considere un éxito es el programa en funcionamiento
- ▣ La I.S. obliga a emprender la elaboración de documentación que hará el proceso más lento.

Bibliografía

- Pressman, R.S.
 - ▣ Ingeniería del Software. Un enfoque práctico
 - 6ª Edición. 2005
- Sommersville, I.
 - ▣ Ingeniería de Software. 9ª Edición 2011
- Chaos Report
 - ▣ <https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf>
 - ▣ https://www.standishgroup.com/sample_research_files/CHAOSReport2015-Final.pdf