Tema 1.
Introducción a la Ingeniería del Software

Metodología y Desarrollo de Programas

Este documento se ha realizado a partir de la presentación de:

Ingeniería del Software para principiantes

Sorey Bibiana García Zapata

Ingeniería Informática

Politécnico Jaime Isaza Cadavid

Introducción

- Ingeniería del software
  - Ingeniería, sí con todo lo que ello supone
  - Objetivo: desarrollo costeable de sistemas software
    - ✓ Menos restricciones “físicas”
    - ✓ Pero... es algo más abstracto y, por tanto, más complejo

- Sistema software
  - ¿Qué es?
  - ¿Qué características tiene?
  - ¿Qué es lo más costoso del desarrollo software?

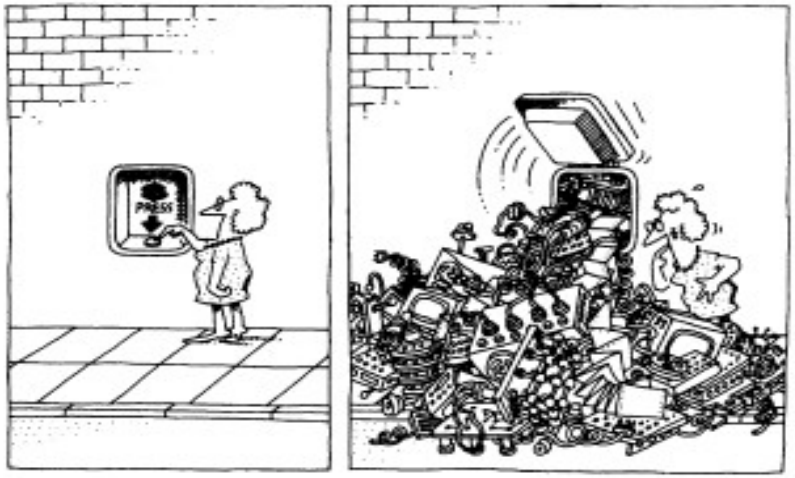
Introducción

- **Software**:
  1) Instrucciones de ordenador que cuando se ejecutan **cumplen una función** y tienen un **comportamiento deseado**,
  2) Estructuras de datos que facilitan a los programadores la adecuada manipulación de la información, y
  3) Documentos que describen la operación y el uso de los programas.

- Características
  - El software no se fabrica, se desarrolla (coste de ingeniería)
  - El software no se estropea, se deteriora
  - El software se construye a medida

- El proceso de construcción de software requiere, como cualquier otra ingeniería, identificar las tareas que se han de realizar sobre el software y aplicar estas tareas de forma ordenada y efectiva

¿Porqué es tan difícil hacer software?



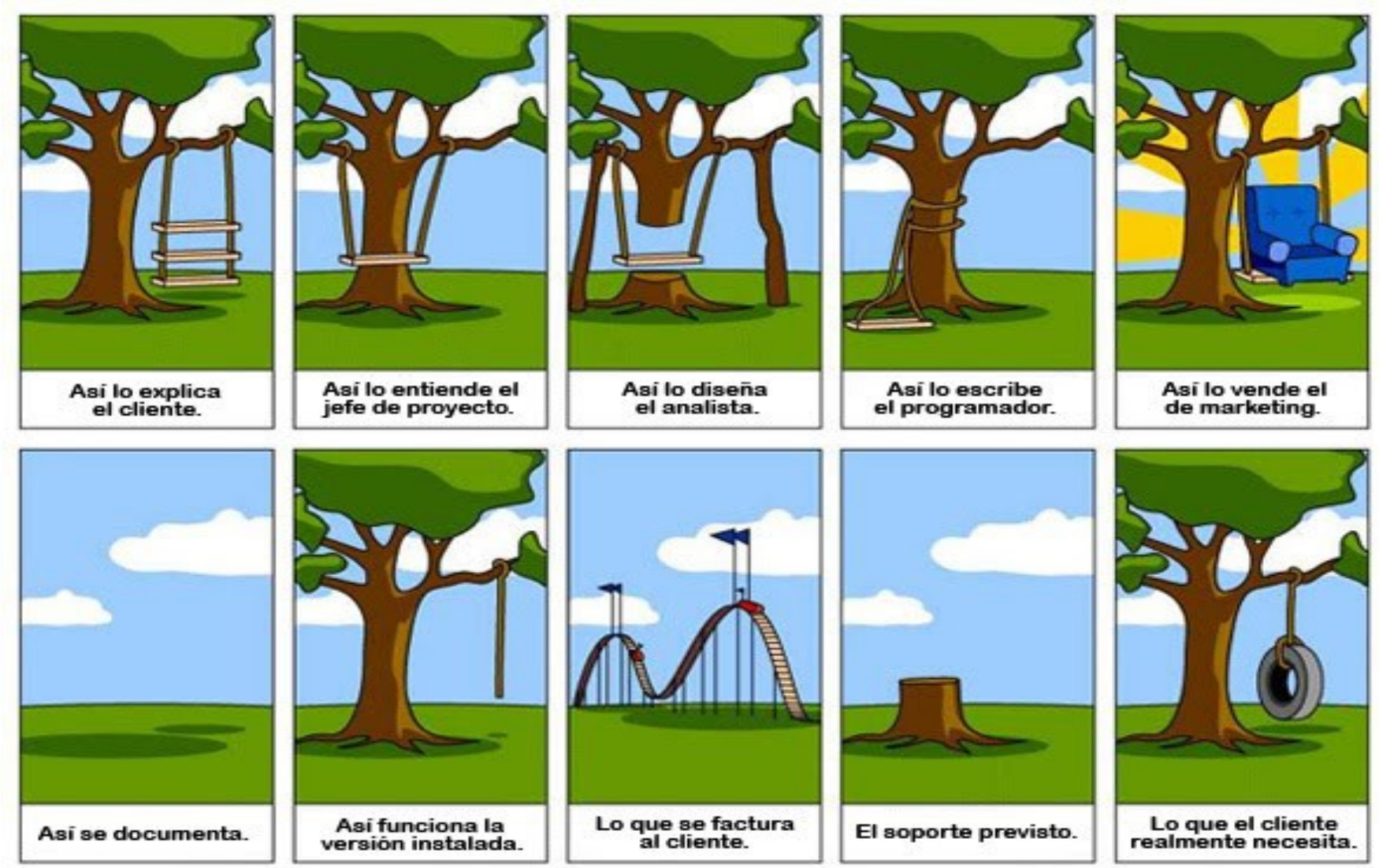
La tarea del equipo de desarrollo de software es ofrecer ilusión de simplicidad.
[Booch 94]

http://www.youtube.com/watch?v=L2zqTYgcpfg

Problemas relacionados con el software

- Suelen estar construidos y mantenidos por una sola persona (software artesanal)
- Codificación inmediata (Costo mayor de lo previsto inicialmente, tiempo de desarrollo excesivo, la calidad del código producido es imprevisible y en promedio baja)
- La planificación y la estimación de costes son muy imprecisas
- Dificultad de entendimiento entre cliente y empresa de desarrollo =>El cliente queda insatisfecho

- Entendimiento Cliente - Empresa



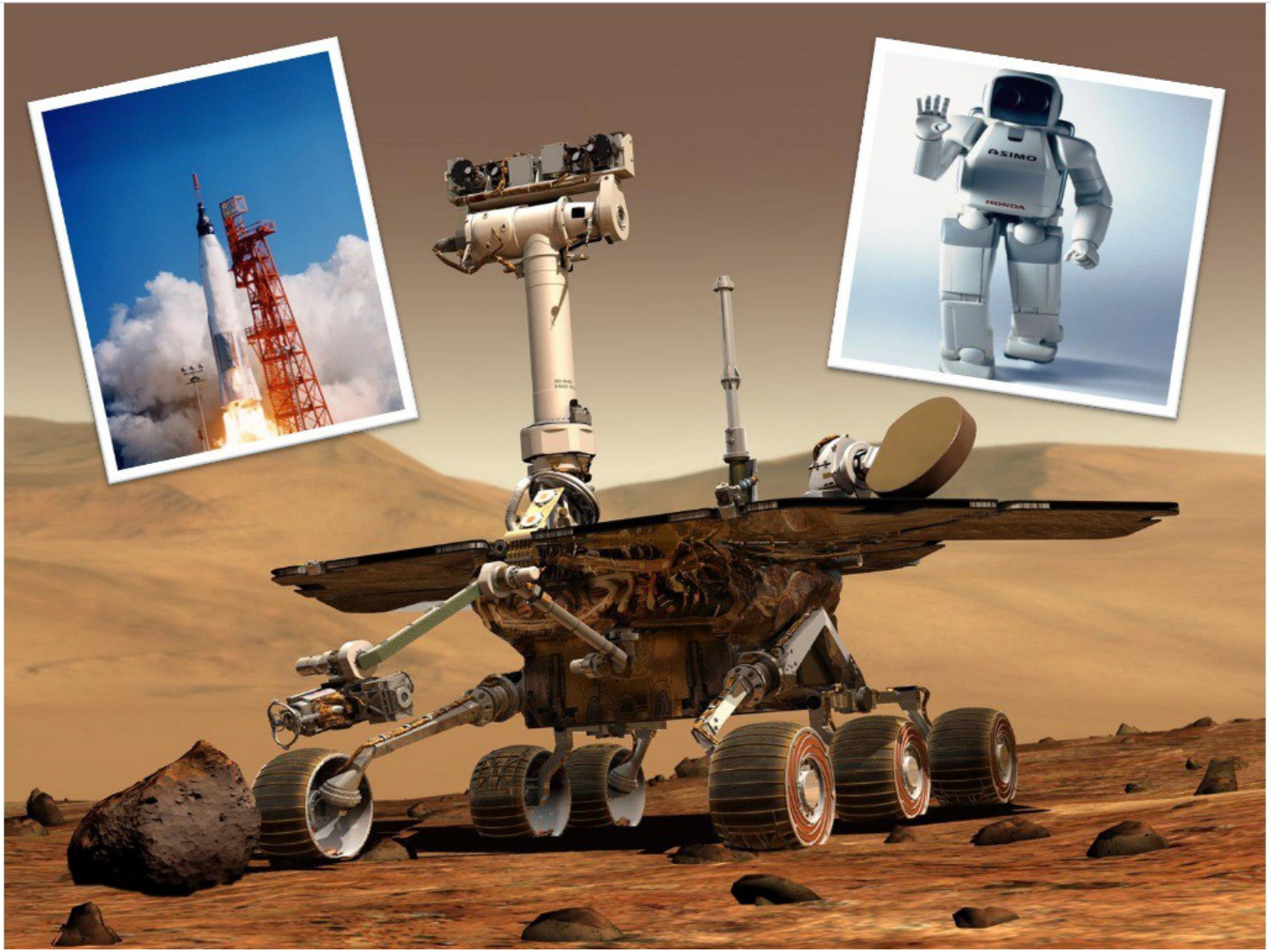
- Propiedades de los sistemas de software simples o artesanales
  - Suelen estar construidos y mantenidos por una sola persona (software artesanal)
    - ✓ No suelen pasar de 1000000 de líneas de código, aunque el número de líneas no sea la mejor medida de la complejidad del software
  - Ciclo de vida corto
  - Pueden construirse aplicaciones alternativas en un periodo razonable de tiempo
  - No necesitan grandes esfuerzos en análisis y diseño

- Propiedades de los sistemas de software complejos
  - También se denomina software de dimensión industrial
  - Es muy difícil o imposible que un desarrollador individual pueda comprender todas las sutilidades de su diseño
  - La complejidad es una propiedad esencial de estos sistemas, que puede dominarse, pero no eliminarse
  - Ejemplo: Sistema de reservas, anulaciones y venta de billetes aéreos para un conjunto de compañías aéreas que se pueda utilizar en cualquier lugar del mundo

Ha pensado alguna vez, **¿Dónde hay software?**


HEROES BAJO CERO
TIM ALLEN
TOM HANKS
THE TOYS ARE BACK IN TOWN!
JEDI
Sith
COMING SOON!

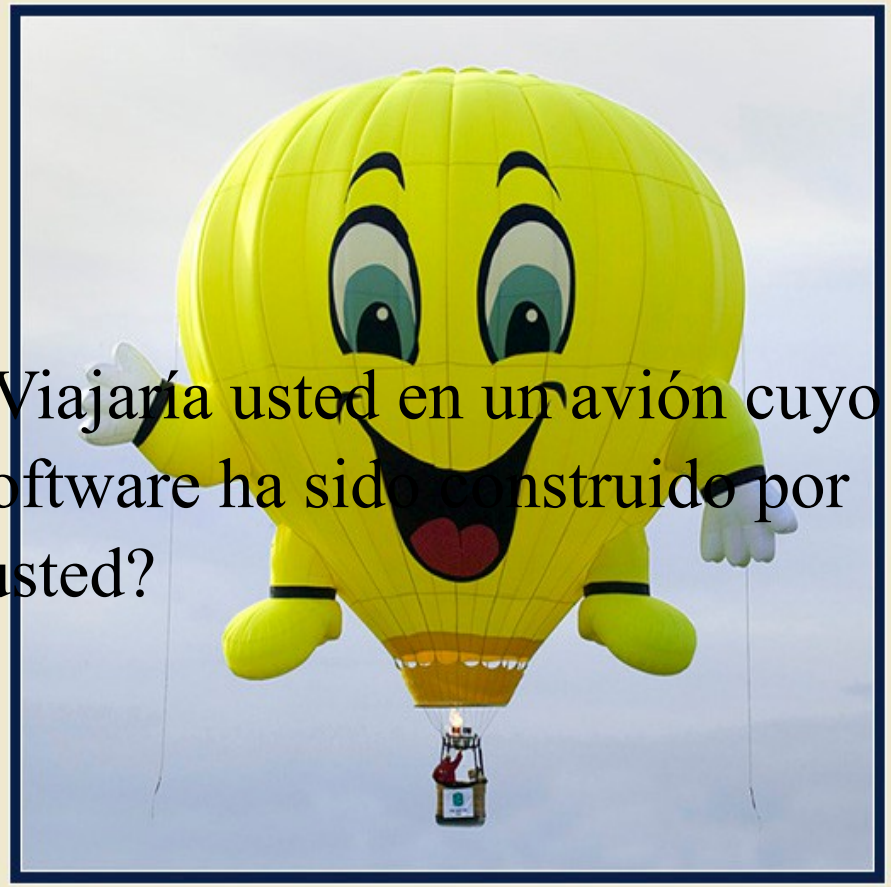

VAIO
SONY
HELIO
Windows XP
Home Edition
AOL
PLAYSTATION 3
Wii


ASIMO
HONDA

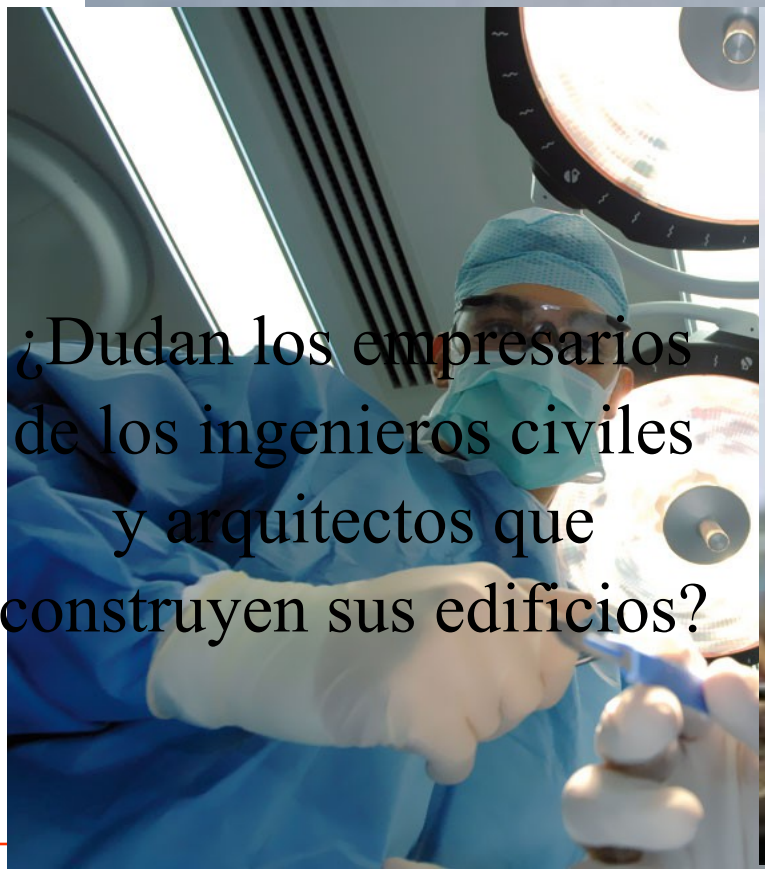

TransMilenio
U064
Avianca
Avianca
301

La pregunta más bien sería **¿Dónde NO hay software?**

- ¿Un par de preguntas?



¿Viajaría usted en un avión cuyo software ha sido construido por usted?



¿Dudan los empresarios de los ingenieros civiles y arquitectos que construyen sus edificios?

- Si desconoce las **motivaciones** asociadas a su trabajo, las **responsabilidades** que implica y las **consecuencias** que acarrea su mal desempeño, es decir, si no sabe **ingeniería del software** o le importa poco lo que significa, pues usted no sabe hacer software, sólo sabe **programar**

- Un programador es sólo un rol del conjunto de roles implicados en el proceso de desarrollo de software

- Para desarrollar software existen una serie de **roles asociados**, encargados de **analizar**, **planificar** y **establecer**, **qué** es lo que va a desarrollarse, **cómo**, con **cuantos recursos**, en **cuento tiempo** e incluso a que **nivel de calidad**

- Roles existentes en la Ingeniería del Software
  1. Programadores (Aplicaciones, sistemas, bases de datos)
  2. Analistas de Sistemas
  3. Administradores
  4. Jefe de Proyectos
  5. Diseñadores de Sistema
  6. Documentador
  7. Probadores
  8. Arquitecto: red, sistemas, web
  9. Expertos en seguridad
  10. Integradores
  11. .....

¿Qué es la ingeniería del software?

¿Qué es la ingeniería del software?

«El establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales» (Bauer, 1972)

«*La disciplina tecnológica y de gestión que concierne a la producción y el mantenimiento sistemático de productos software desarrollados y modificados dentro de unos plazos estipulados y costes estimados*.» (Fairley, 1985)

«*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 Std 610-1990)

¿Qué es la ingeniería del software?

«Ingeniería es la aplicación sistemática de conocimiento científico en la creación y construcción de soluciones, que satisfacen una buena relación efectividad/precio, de problemas prácticos al servicio de la humanidad. La ingeniería del software es la forma de ingeniería que aplica los principios de las ciencias de la computación y las matemáticas en la obtención de soluciones de los problemas del software que satisfacen una buena relación efectividad/precio.»

SEI Report on Undergraduate Software Engineering Education, 1990.

Ingeniería del Software. Una cosa más....

- ¿Qué sucede con la calidad del software?

Ingeniería del Software. Una cosa más....

- ¿Cómo probáis que vuestro código funciona?

Calidad del software

Uno de los objetivos principales: Producir software de calidad

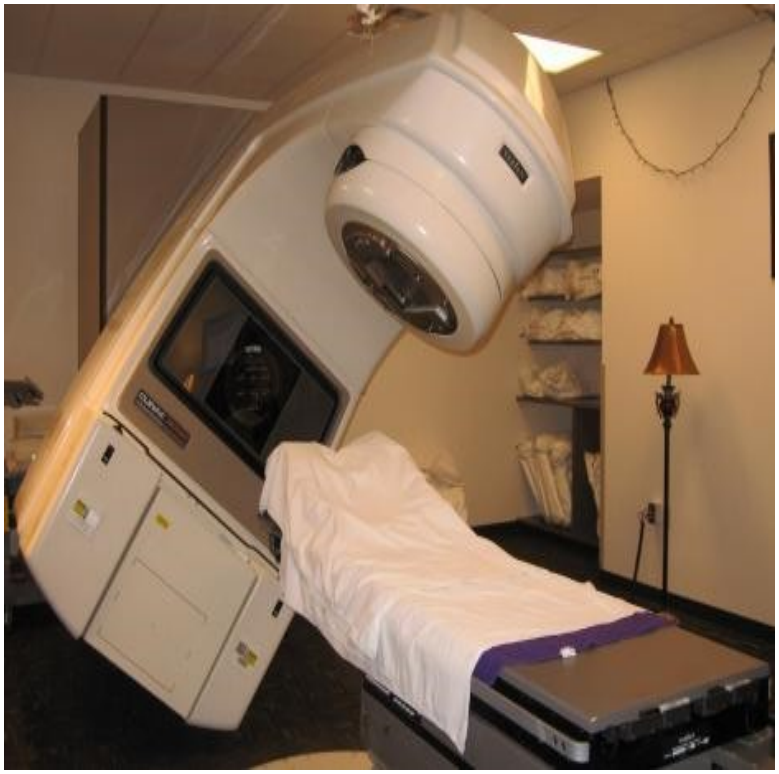
*"La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario". (*IEEE, Std. 610-1990).

"Concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario" (Pressman, 1998)

- Acaso en software, ¿"lo que importa" es que básicamente funcione?

Therac-25 (1985 – 1987)

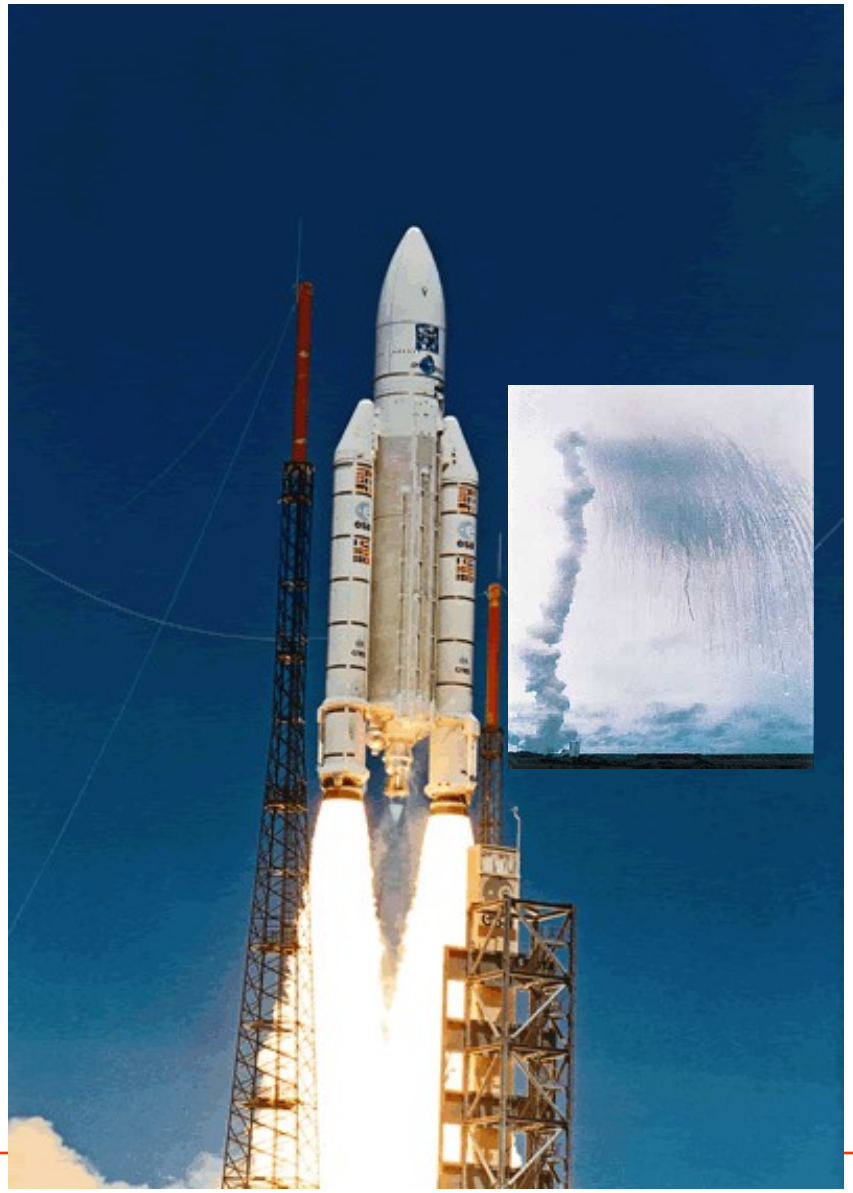
Era una máquina empleada en terapia de radiación, producida por *Atomic Energy of Canada Limited*, notoria por haber sido objeto de un error software, causando al menos seis accidentes y que le costó la vida al menos a cinco personas.



Mariner 1 (28 de Julio de 1962)

Un guión en las instrucciones del programa de guiado del cohete provocó la desviación del Atlas y tuvo que enviarse un comando para su autodestrucción a los 4 minutos y 53 segundos de su lanzamiento.

Calidad del Software



Vuelo 501 del Ariane-5 (4 de Junio de 1996)

Otro ejemplo documentado sobre el daño ocasionado por software mal diseñado es el de la explosión de la lanzadera Ariane-5, cuando a 40 segundos después de la iniciación de la secuencia de vuelo, la lanzadera se desvió de su ruta, se partió y explotó.

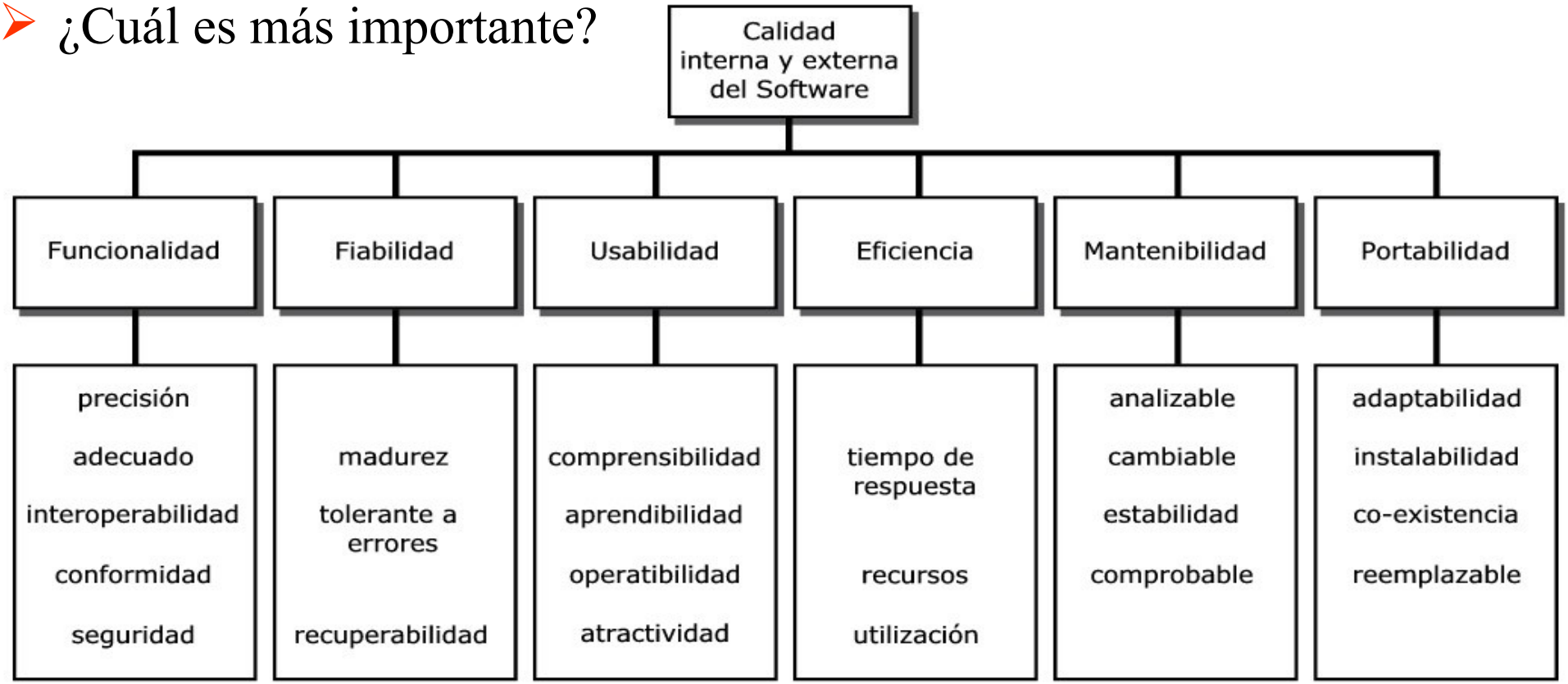
En el proyecto global se invirtieron 10 años de construcción y 7 mil millones de euros, lo que supuso un duro golpe para la Agencia Espacial Europea (ESA).

Caza Harrier (revista Pilot 1996)

Dos oficiales de policía usaban una pistola radar para localizar a motoristas. Se les quedó el radar apuntando hacia arriba. Pasó entonces un Harrier a baja altura, detectó el radar y lo identificó como enemigo. La respuesta automática era el disparo de un misil. Afortunadamente operaba desarmado.

Calidad del Software

- Factores que determinan la calidad del software
  - Externos. Detectados por el usuario de producto
  - Internos. Detectados sólo por los desarrolladores
- ¿Cuál es más importante?



Concluyendo .....

El desarrollo de software incluye todas las disciplinas asociadas a la ingeniería de software

desde el análisis hasta la puesta en producción

Importante!

...desarrollar software, no es programar…

Futuros temas ....

Se verá:

- **Proceso de Desarrollo**
  - Conjunto de actividades, métodos, prácticas y herramientas utilizados en la producción y evolución de software.
- **Un modelo de Ciclo de Vida**
  - Divide el desarrollo en fases y prescribe las actividades que deben realizarse en cada fase
  - Proporciona criterios para determinar cuándo cada fase de desarrollo ha terminado
- **Usando distintas metodologías**
  - Estructurada, orientada a objetos, ágiles.....

Análisis, Diseño, implementación, Integración y Verificación, Instalación y Mantenimiento

Bibliografía

- [Larman99] UML y Patrones. C. Larman. Prentice Hall, 1999.
- [Booch99]: El Lenguaje Unificado de Modelado. G. Booch, J. Rumbaugh, I. Jacobson. Addison Wesley Iberoamericana, 1999.
- [Booch94]: Object-Oriented Analysis and Design. G. Booch. Benjamin/Cummings, 1994.
- [BJR97]: The UML Specification Document. G. Booch, I. Jacobson and J. Rumbaugh. Rational Software Corp., 1997.
- [Jacobson92] Object-Oriented Software Engineering: A Use Case Driven Approach. I. Jacobson. Addison-Wesley, 1992.
- [Rumbaugh91] Object-Oriented Modeling and Design. J. Rumbaugh et al.Prentice-Hall,1991.
- [Pressman07] Software Engineering: A Practitioner's Approach, 7/e, McGraw-Hill, 2009