



Bloque I

Tema 1

Introducción a la Ingeniería del software



Contenido

- Características y Evolución del Software
- Naturaleza y Problemas del Desarrollo de Software
- La Ingeniería del Software
- Solución de Problemas
- Participantes y Papeles
- Otros Conceptos de la Ingeniería del Software
- Principios de la Ingeniería del Software



Tema 1.- Introducción a la Ingeniería del software

Características y Evolución del Software

Un poco de historia

- Primeras décadas:
 - Desarrollar el hardware
 - Reducir costes de procesamiento y almacenamiento
- Década de los ochenta:
 - Desarrollo de la microelectrónica
 - Mayor potencia de cálculo y reducción de costes
- Objetivo actual: mejorar la calidad de las soluciones software.

Características y Evolución del Software

1959 - 1965

- Orientación por lotes
- Distribución limitada
- Software a medida



1965 - 1975

- Multiusuario
- Tiempo real
- Bases de datos
- Software como producto
- Mayores gastos de mantenimiento



1975 - 1989

- Sistemas distribuidos
- Inteligencia Artificial
- Hardware de bajo coste
- Impacto en el consumo
- Redes área local y global
- Gran demanda



1989 -

- Potentes sistemas de sobremesa
- Tecnología de objetos
- Sistemas expertos
- Redes neuronales
- Cliente/servidor
- Tecnologías de Internet.

AUMENTAN los problemas del desarrollo de software:

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



Tema 1.- Introducción a la Ingeniería del software

Características y Evolución del Software

Actualmente el software es el factor decisivo a la hora de elegir entre varias soluciones informáticas disponibles para un problema dado, pero esto no ha sido siempre así.

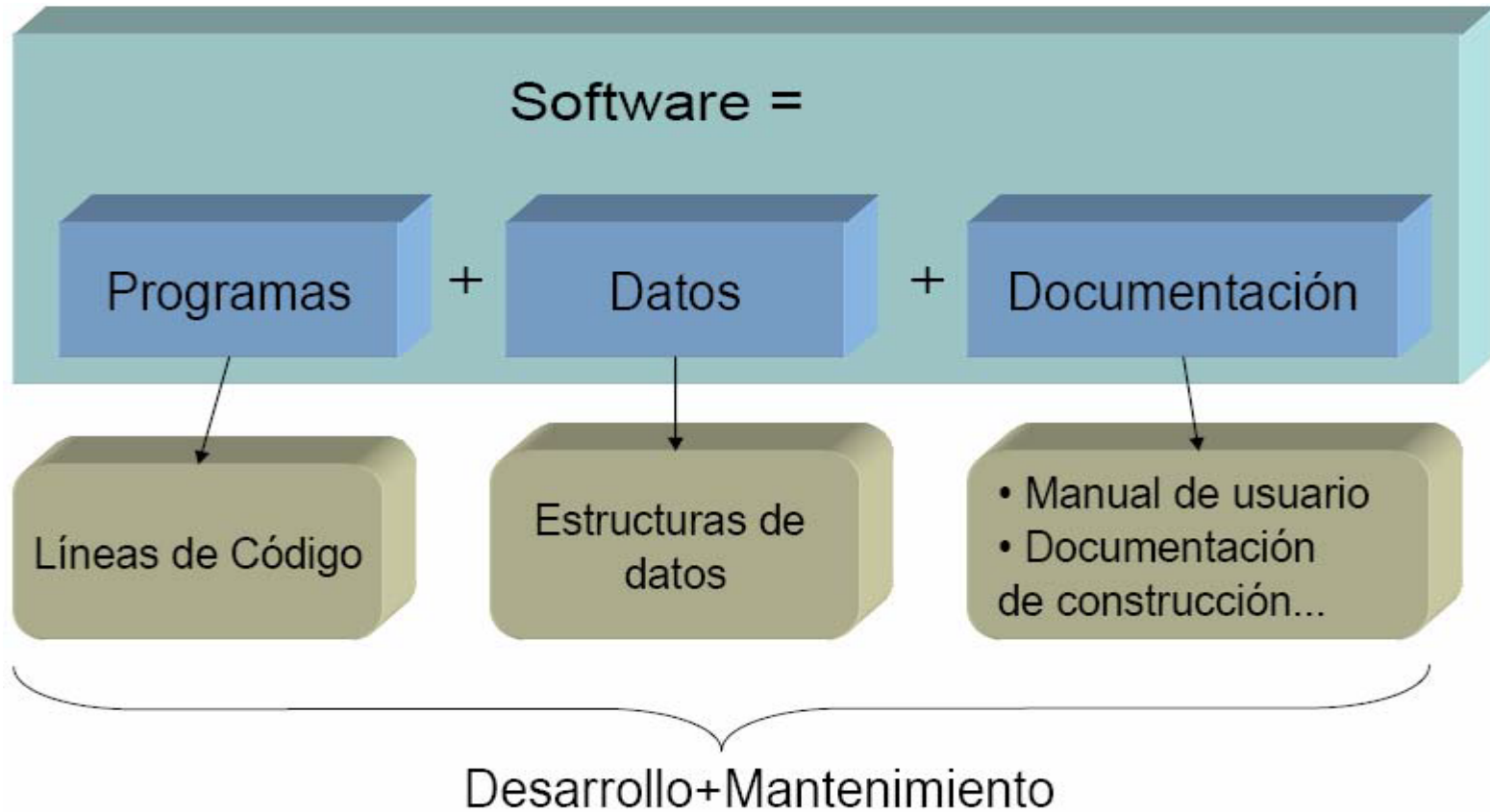
En los primeros años de la informática, el hardware tenía una importancia mucho mayor que en la actualidad. Su coste era comparativamente mucho más alto y su capacidad de almacenamiento y procesamiento, junto con su fiabilidad, era lo que limitaba las prestaciones de un determinado producto.

El software se consideraba como un simple añadido, a cuyo desarrollo se dedicaba poco esfuerzo y no se aplicaba ningún método sistemático.

La programación era un arte de andar por casa, y el desarrollo de software se realizaba sin ninguna planificación.

Debido a este entorno personalizado de software, el diseño era un proceso implícito, realizado en la mente de alguien y la documentación normalmente no existía.

Características y Evolución del Software





Tema 1.- Introducción a la Ingeniería del software

Características y Evolución del 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

Tema 1.- Introducción a la Ingeniería del software

Características y Evolución del Software

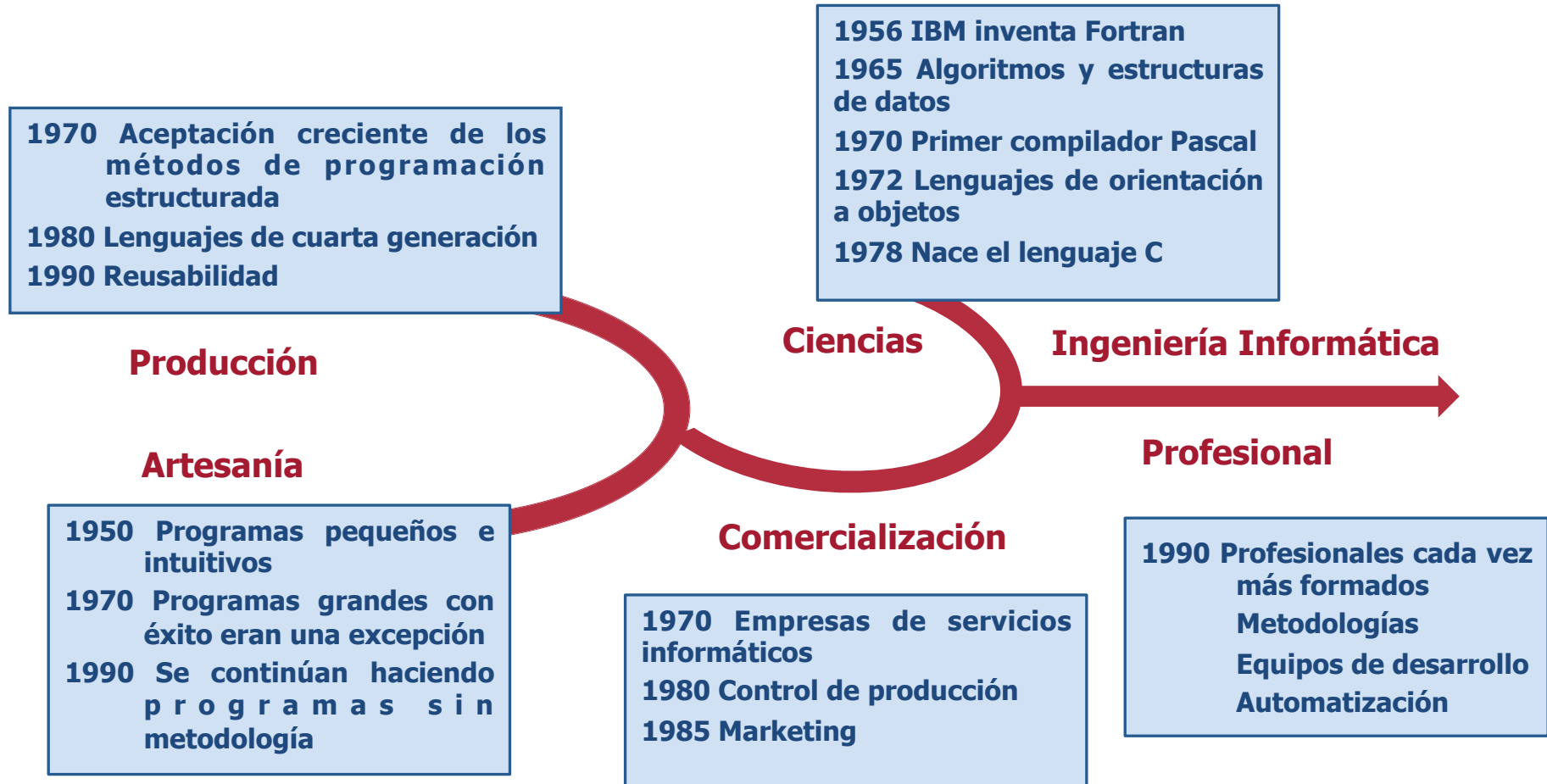


Evolución del Software hacia la Ingeniería



Tema 1.- Introducción a la Ingeniería del software

Características y Evolución del Software



Evolución del Software hacia la Ingeniería

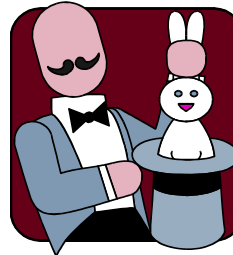
Naturaleza y Problemas del Software

- 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
 - Desarrollo a medida (ausencia de componentes)
- 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 Software

- Causas de la crisis del software
 - Naturaleza lógica del software
 - Mala gestión de los proyectos (ausencia de datos, deficiente comunicación, etc.)
 - Ausencia de entrenamiento formal en nuevas técnicas (programadores vs. ingenieros de software)
 - Resistencia al cambio
 - Mitos del software:



MITOS DE GESTIÓN

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

MITOS DE LOS DESARROLLADORES

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

MITOS DEL CLIENTE

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



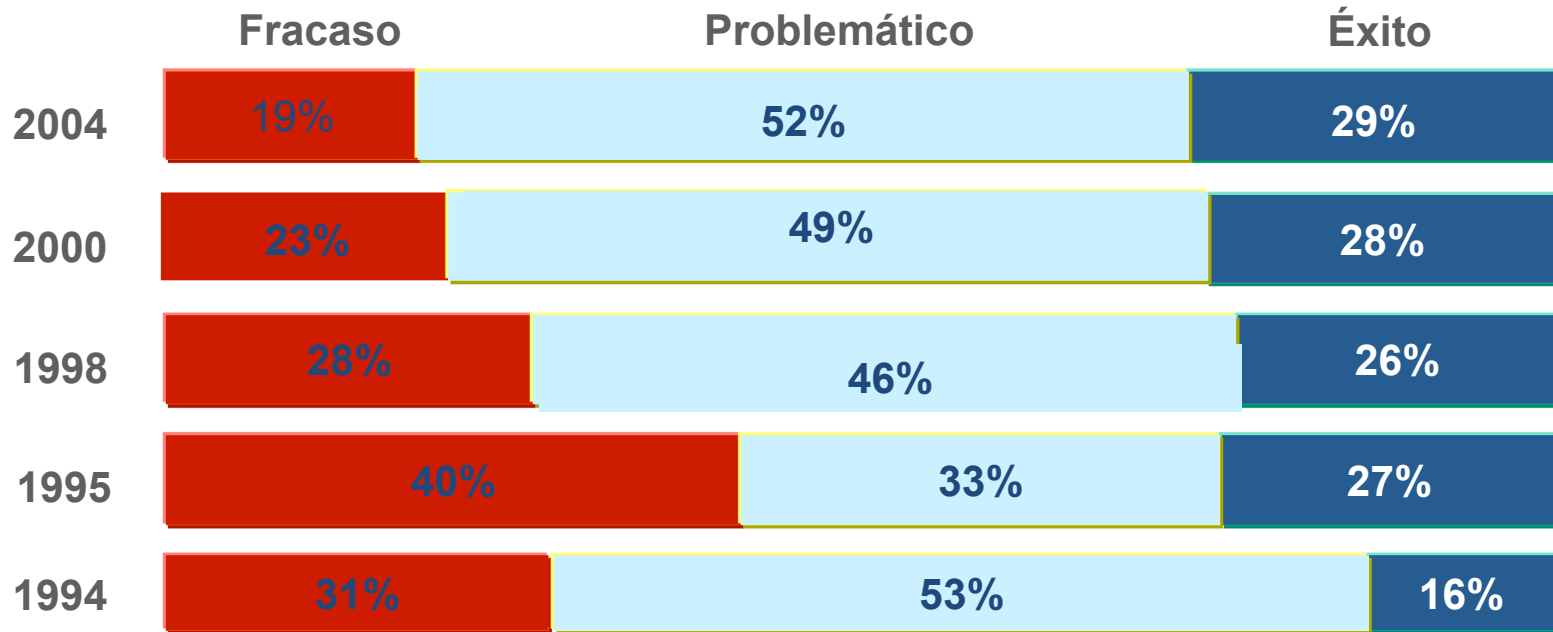
Tema 1.- Introducción a la Ingeniería del software

Naturaleza y Problemas del Software

- 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 de cambiar de plataforma
 - **Eficiencia**: No hay aprovechamiento óptimo de recursos
- Consecuencias:
 - Baja productividad
 - Baja calidad
- Solución:
 - Aplicar la ingeniería del Software en la construcción de sistemas informáticos
 - La necesidad de un enfoque de ingeniería en el desarrollo de software fue propuesta en una conferencia de la OTAN en 1968

Naturaleza y Problemas del Software

Proyecto para el desarrollo de sistemas software



El proyecto se aborta o el sistema no se llega a utilizar



Desbordamiento de agendas o costes. Las funcionalidades no cubren las expectativas. Problemas funcionales



Proyecto realizado en el tiempo previsto, con los costes previstos, con la funcionalidad esperada y ofreciendo un funcionamiento correcto.

Fuente: Standish Group Survey,



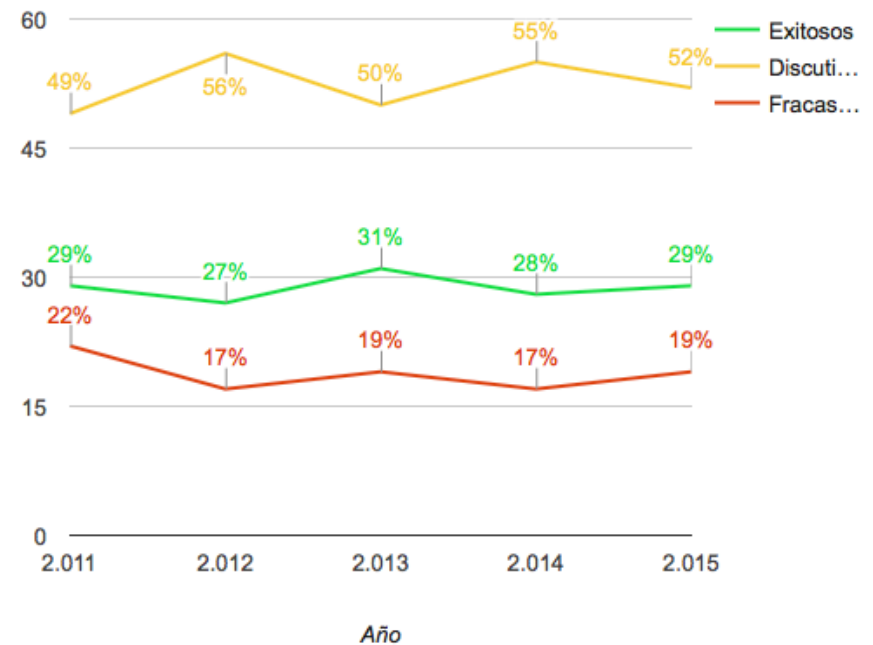
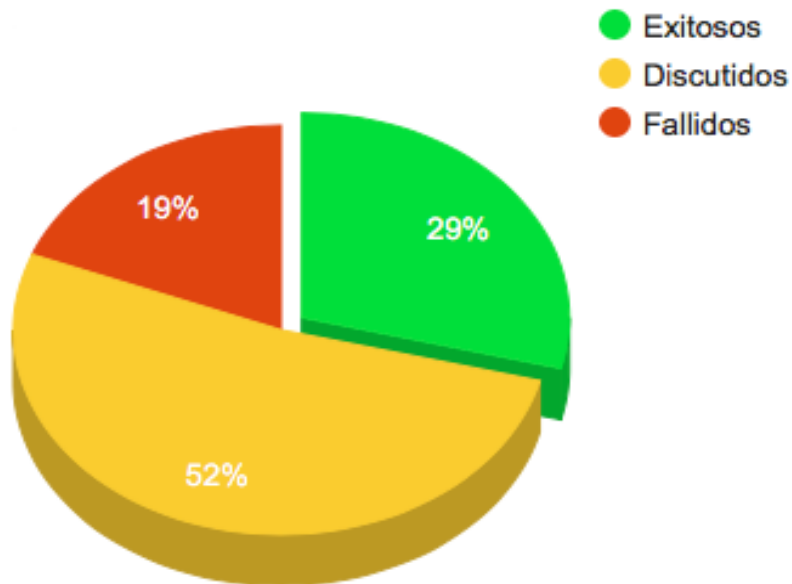
Tema 1.- Introducción a la Ingeniería del software

Naturaleza y Problemas del Software

- Este problema se identificó por primera vez en 1968, año en el que la organización NATO desarrolló la primera conferencia sobre desarrollo de software, y en la que se acuñaron los términos **"crisis del software"** para definir a los problemas que surgían en el desarrollo de sistemas de software, e **"ingeniería del software"** para describir el conjunto de conocimientos que existían en aquel estado inicial.
- Algunas referencias útiles para comprender cuáles eran los conocimientos estables para el desarrollo de software en 1968 son:
 - En 1962 se publicó el primer algoritmo para búsquedas binarias.
 - C. Böhm y G. Jacopini publicaron en 1966 el documento que creaba una fundación para la eliminación de "GoTo" y la creación de la programación estructurada.
 - En 1968 los programadores se debatían entre el uso de la sentencia GoTo, y la nueva idea de programación estructurada; ese era el caldo de cultivo en el que Edsger Dijkstra escribió su famosa carta "GoTo Statement Considered Harmful" en 1968.
 - La primera publicación sobre programación estructurada no vio la luz hasta 1974, publicada por Larry Constantine, Glenford Myers y Wayne Stevens.
 - El primer libro sobre métrica de software fue publicado en 1977 por Tom Gilb.
 - El primero sobre análisis de requisitos apareció en 1979

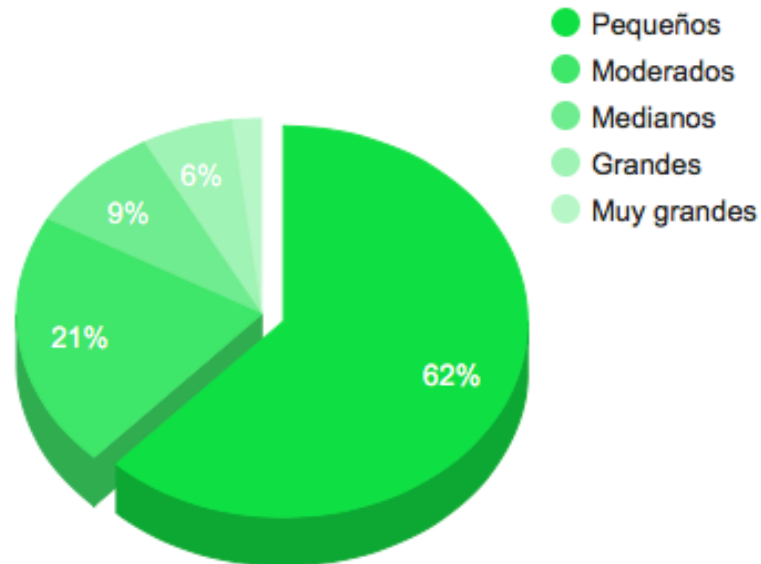
Naturaleza y Problemas del Software

Standish Group en el informe del 2015 ha estudiado unos 50.000 proyectos de todo el mundo y se ha modificado la definición de éxito del proyecto, en lugar de considerarse como el cumplimiento de las tres restricciones (Alcance, presupuesto y plazos), se consideran los plazos, el presupuesto y la obtención de resultados satisfactorios aunque no se cubra el alcance completamente.



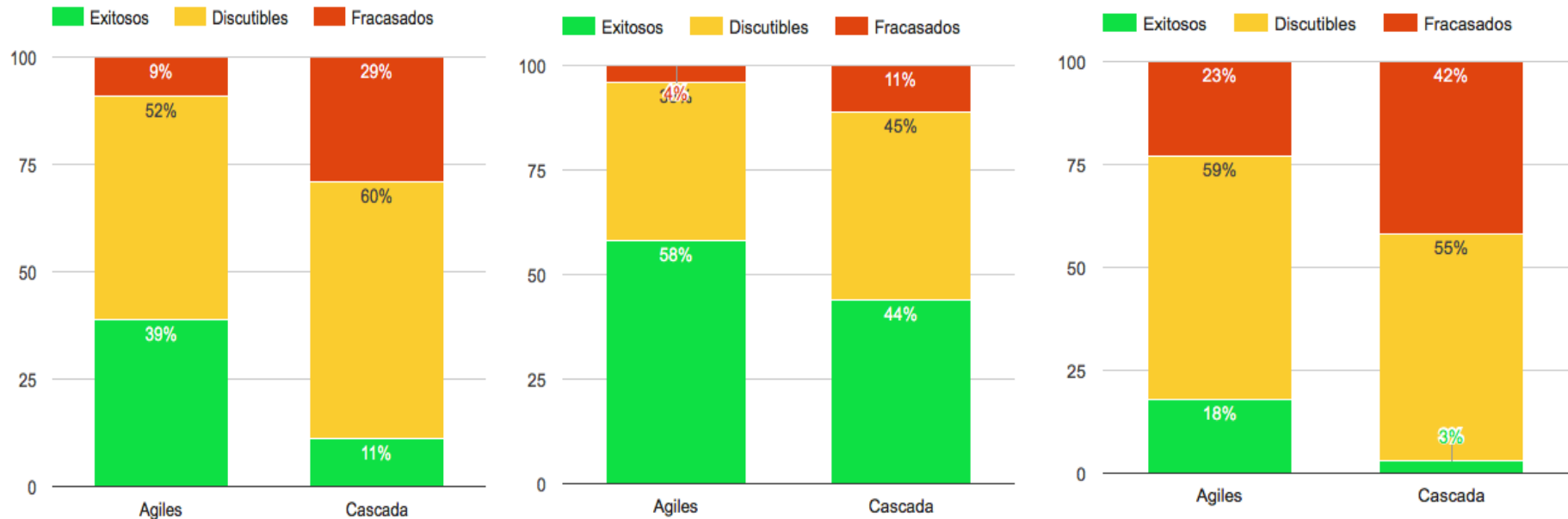
Naturaleza y Problemas del Software

Dado que según lo observado no hay ninguna tendencia en el comportamiento de los proyectos, es decir se mantienen con leves oscilaciones en un 20% para los fallidos, 30% para de éxito y 50% para los discutidos, podemos deducir que ni las metodologías, ni los ciclos de vida, etc. tienen mucha influencia en los resultados. Si nos fijamos en los tamaños de los proyectos que han tenido éxito su segmentación nos indica que el tamaño si importa y debemos aplicar la máxima de "Siempre que puedas que tu proyecto sea pequeño" y aplicar el principio de "divide y vencerás"



Naturaleza y Problemas del Software

Si consideramos las dos metodologías extremas como son Ágil y Cascada para todos los proyectos (gráfica izquierda), para proyectos pequeños (gráfica centro) y para grandes (gráfica derecha) vemos los resultados



Hay que considerar en estos resultados que los proyectos en cascada son mucho mas numerosos que los agiles y por lo tanto los datos son relativos ya que las muestras tienen distinto tamaño, por lo que el dato puede ser engañoso. Además hay muchas mas metodologías predictivas que la considerada.



Tema 1.- Introducción a la Ingeniería del software

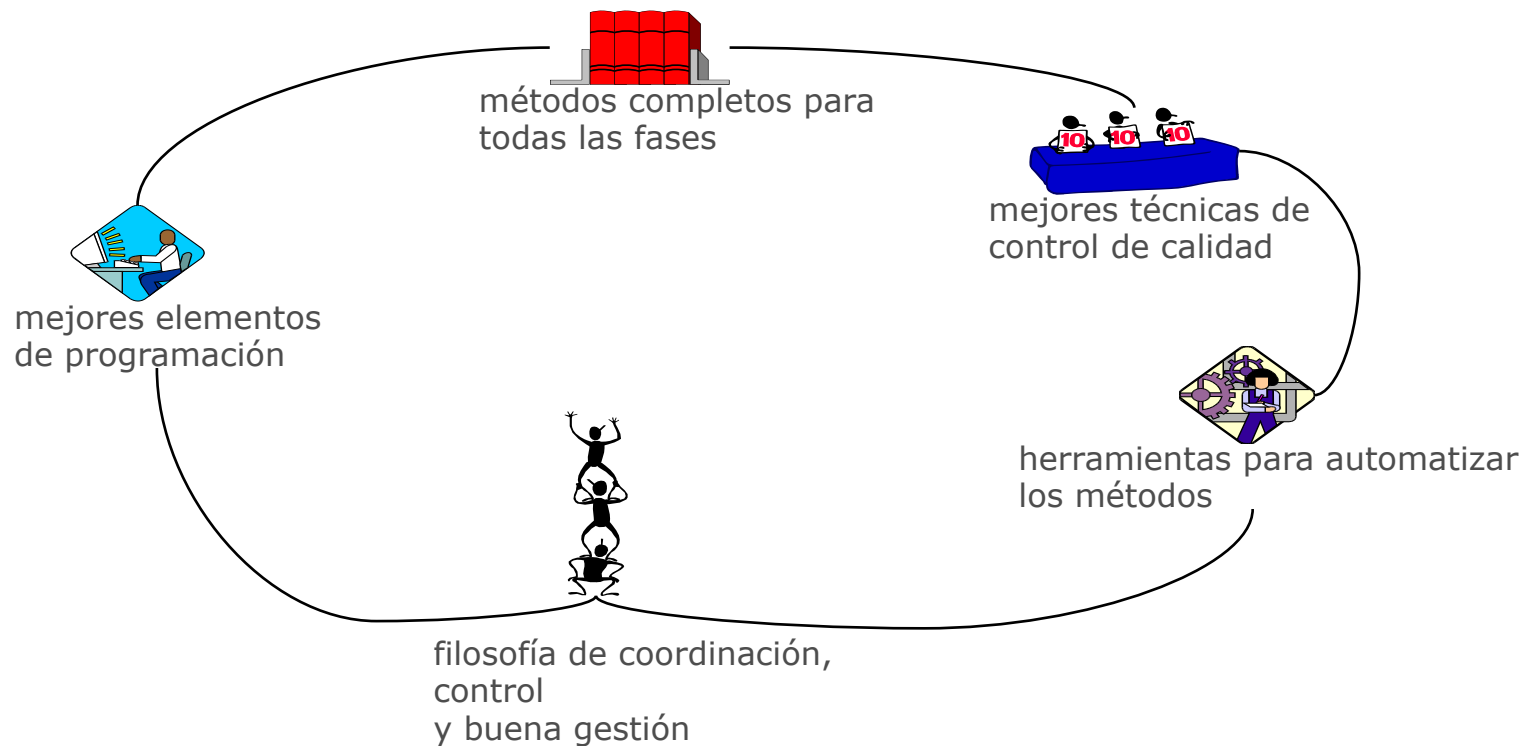
La Ingeniería del Software

Definiciones

- 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:
 - ✓ “disciplina de ingeniería”: aplicación de teorías, métodos y herramientas para solucionar problemas, y teniendo en cuenta restricciones financieras y organizativas
 - ✓ “todos los aspectos de producción”: comprende procesos técnicos del desarrollo y actividades como la administración de proyectos, desarrollo de herramientas, métodos y teorías
- Actividad de
 - ✓ modelado
 - ✓ solución de problemas
 - ✓ adquisición de conocimiento
 - ✓ dirigida por una fundamentación

La Ingeniería del Software

- Trata de ser la respuesta a la crisis del software
- Combinación de elementos:





Tema 1.- Introducción a 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 NATO)

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



Tema 1.- Introducción a la Ingeniería del software

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 (SEI, IEEE, ISO) 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**, introduciendo métodos y formas de trabajo sistemáticos, disciplinados y cuantificables
- La forma de trabajo de programadores individuales surgida por la necesidad de los primeros programas, ha creado una cultura de la programación heroica, para el desarrollo de software que es la principal causa de los problemas apuntados, y en la actualidad una de las principales resistencias a la implantación de técnicas de ingeniería para el desarrollo de sistemas



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software

En definitiva:

¿Qué es la Ingeniería del Software?

- La IS es aplicar el sentido común al desarrollo de sistemas software

¿Qué es el sentido común?

- Planificar antes de desarrollar
- Diseñar antes de programar
- Reutilizar diseños que funcionan y sean mantenibles
- Utilizando las herramientas apropiadas (Herramientas CASE)
 - Edición de diagramas
 - Comprobar la consistencia de los diagramas
 - Generación de documentación
 - Seguimiento de actividades del proyecto
 - Diferente soporte según el tipo
 - ✓ Upper-CASE. Herramientas que ayudan en las actividades de captura de requisitos, análisis y diseño
 - ✓ Lower-CASE. Herramientas para la programación, depuración y pruebas



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software

- La Ingeniería del Software es una ingeniería muy joven que necesitaba:
 - Definirse a sí misma: ¿Cuáles son las áreas de conocimiento que la comprenden?
 - Definir los procesos que intervienen en el desarrollo, mantenimiento y operación del software
 - De las mejores prácticas, extraer modelos de cómo ejecutar esos procesos para evitar los problemas de la “crisis del software”
 - Definir criterios unificadores para las tareas de requisitos, pruebas, gestión de la configuración, etc.
- Los estándares son útiles porque:
 - Agrupan lo mejor y más apropiado de las buenas prácticas y usos del desarrollo de software
 - Engloban los “conocimientos”
 - Proporcionan un marco para implementar procedimientos de aseguramiento de la calidad
 - Proporcionan continuidad y entendimiento entre el trabajo de personas y organizaciones distintas



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software

- Desde la identificación del fenómeno “**crisis del software**”, han sido muchas las organizaciones que han abordado, con mayor o menor rigor, el análisis de problemas en el desarrollo de sistemas de software. Sus trabajos se han encaminado a la localización de las causas; y a la exposición en textos didácticos, normativos o estándares de procesos o prácticas necesarias para abordar el desarrollo, mantenimiento y operación con las mayores garantías de éxito.
- Han sido muchos los departamentos de universidades, organismos de normalización o investigación nacionales o internacionales, sociedades de profesionales, departamentos de defensa, departamentos de calidad y procesos de empresas los que han ido generando normas y estándares.
- Este compendio considera como entidades de mayor reconocimiento internacional, por sus trabajos y esfuerzos realizados para la normalización, y reconocimiento de la Ingeniería del software a: **ISO, IEEE- Computer Society y SEI**.



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software



- ISO
 - Organización Internacional para la Estandarización. Fundada en 1947
 - Son miembros 87 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. La misión del JTC1 es la “estandarización en el campo de los sistemas de tecnologías de la información”, incluyendo microprocesadores y equipos
 - Los estándares o instrucciones técnicas más importantes para la Ingeniería del Software:
 - ✓ ISO/IEC 12207
 - ✓ ISO/IEC TR 15504



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software



- SEI

- Instituto de Ingeniería del software. (SEI <http://www.sei.cmu.edu/>).
- Integrado en la Universidad Carnegie Mellon.
- Los trabajos y aportaciones realizadas por el Instituto de Ingeniería del Software a la Ingeniería del software son también referente mundial de primer orden, siendo la aportación más significativa los modelos de madurez de las capacidades: **CMM** y **CMMI**; que en sus casi 15 años de implantación efectiva en entornos de producción de software han demostrado su efectividad en las dos finalidades que cubren: como marco de referencia para mejora de procesos, y como criterio de evaluación para determinar la madurez, y por tanto fiabilidad de resultados previsibles de una organización de software.



Tema 1.- Introducción a la Ingeniería del software

La Ingeniería del Software



- IEEE Computer Society
 - 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) es una sociedad integrada en IEEE, 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.
- Estándares para la Ingeniería del Software
 - IEEE ha desarrollado estándares para todas las áreas de Ingeniería del Software.
 - Algunos de ellos, correspondientes a las principales áreas específicas de la Ingeniería del Software son:
 - ✓ IEEE Std. 830 Prácticas recomendadas para las especificaciones de software.
 - ✓ IEEE Std. 1362 Guía para la especificación del documento de requisitos "ConOps"
 - ✓ IEEE Std. 1063 Estándar para la documentación de usuario de software.
 - ✓ IEEE Std. 1012 Estándar para la verificación y validación de software.
 - ✓ IEEE Std. 1219 Estándar para el mantenimiento del software



Tema 1.- Introducción a la Ingeniería del software

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 ...**



Tema 1.- Introducción a la Ingeniería del software

Solución de Problemas

- Los ingenieros de software buscan una solución adecuada, en varios pasos:
 - Formular el problema
 - Analizar el problema
 - Buscar soluciones
 - Decidir la solución más adecuada
 - Especificar la solución
- Actividades básicas del desarrollo
 - Obtención de requerimientos
 - Análisis
 - Diseño del sistema
 - Implementación
- otras actividades del desarrollo para evaluar la adecuación de los modelos
 - Revisiones del análisis: el modelo del dominio del problema se compara con la realidad del cliente
 - Revisiones del diseño: el modelo del dominio de la solución se compara con los objetivos del proyecto
 - Pruebas: el sistema se valida contra el modelo del dominio de la solución
 - Administración del proyecto: se compara el modelo del proceso de desarrollo (calendario y presupuesto) con la realidad (trabajos entregados y recursos gastados)



Tema 1.- Introducción a la Ingeniería del software

Participantes y Papeles

- **Participantes:** todas las personas involucradas en el proyecto
 - Cliente: encarga y paga el sistema
 - Desarrolladores: construyen el sistema (analistas, diseñadores, programadores,...)
 - 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
- **Papel (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



Otros Conceptos de la Ingeniería del Software

- **Sistemas y modelos**
 - Sistema: realidad subyacente
 - Modelo: cualquier abstracción de la realidad
- **Productos de trabajo o Entregable**
 - Artefacto o elemento que se produce durante el desarrollo (documento, fragmento de software,...)
 - Dos tipos:
 - ✓ Producto de trabajo interno: producto para el consumo interno del proyecto (por ejemplo, una revisión de la estructura de la base de datos, resultados de pruebas para el gerente,...)
 - ✓ Entrega: producto de trabajo para un cliente (especificación de requisitos, manual de usuario, producto final,...)
- **Actividades, tareas y recursos**
 - Actividad (o fase): conjunto de tareas que se realiza con un propósito específico (obtención de requisitos, entrega, administración,...) que pueden componerse de otras actividades
 - Tarea: unidad elemental de trabajo que puede ser administrada; consumen recursos, dan como resultado productos de trabajo y dependen de productos de trabajo producidos por otras tareas
 - 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



Otros Conceptos de la Ingeniería del Software

- Objetivos, Requerimientos y Restricciones
 - Objetivos:
 - ✓ Principios de alto nivel que se utilizan para guiar el proyecto
 - ✓ Definen los atributos realmente importantes del sistema (seguridad, fiabilidad,...)
 - ✓ A veces hay conflicto entre objetivos (por ejemplo, seguridad y bajo coste) que aumentan la complejidad del proyecto
 - Requerimientos
 - ✓ Características que debe tener el sistema
 - ✓ Requerimiento funcional: área de funcionalidad que debe soportar el sistema (por ejemplo, *proporcionar billetes de tren*)
 - ✓ Requerimiento no funcional: restricción que se establece sobre el funcionamiento del sistema (por ejemplo, *proporcionar billetes de tren en menos de un segundo*)
 - Otras restricciones: por ejemplo, utilización de un determinado lenguaje, de una determinada plataforma o de un sistema antiguo que el cliente no quiere retirar



Otros Conceptos de la Ingeniería del Software

- Notaciones, métodos y metodologías
 - 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. Por ejemplo:
 - ✓ un algoritmo de ordenación es un método para ordenar elementos en una lista
 - ✓ la administración de la configuración es un método para el seguimiento de los cambios
 - Metodología: colección de métodos para la resolución de una clase de problemas (OMT, metodología de Booch, Catalysis, Proceso Unificado de Desarrollo,...)



Tema 1.- Introducción a 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
- Minimizar la distancia intelectual
- Documentar
- Las técnicas son anteriores a las herramientas
- Primero hazlo correcto, luego hazlo rápido
- Probar, probar y probar (** incluye inspecciones **)



Tema 1.- Introducción a la Ingeniería del software

Principios de la Ingeniería del Software

- Introducir las mejoras y modificaciones con cuidado
- Asume responsabilidades
- La entropía del Software es creciente
- La gente es la clave del éxito
- 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



Tema 1.- Introducción a la Ingeniería del software

Principios de la Ingeniería del Software

- Tener procedimientos escritos de desarrollo de software puede ayudar a crear una cultura compartida de buenas prácticas
- 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
- Si mides los que haces, puedes aprender a hacerlo mejor
- Haz lo que tenga sentido, no recurras a los dogmas
- No pueden cambiar todo de una vez. Identifica los cambios que se traduzcan en los mayores beneficios, y comienza a implementarlos