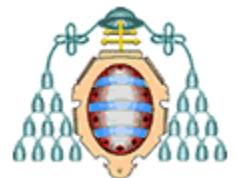


Introducción a la Ingeniería del Software

Ingeniería del Software

Javier Tuya y José García Fanjul
Grupo de Investigación en Ingeniería del Software
<http://giis.uniovi.es>

Curso 2022-2023



Contenido

- Efectos de los fallos del software
- Problemática general y Características del Software
- Ingeniería del Software y Definiciones
- Procesos, Técnicas y Metodologías
- Resumen



¿Alguien quiere un refresco?



Aeropuerto de Gatwick

Costes económicos

Nike Disaster

- \$400 millones de inversión, sistema para automatizar gestión producción y venta. Cadena de suministro. (2001)
 - Objetivo: tiempo de planificación y producción de calzado de 1 mes a 1 semana
 - **Resultado: \$100 millones de caída en ventas. Caída de acciones del 20%.**
 - Problema: El sistema enviaba órdenes de producción duplicadas. Exceso de stock en calzado de baja rentabilidad y escasez en el de alta. Para entregar en tiempo se debieron realizar envíos por avión (\$8 por par) en vez de por barco (75 cents por par).
- Lecciones
 - Software sólido, pero en otros negocios. Necesidad de adaptación, alto riesgo
 - No gestionar/realizar un proyecto piloto y/o paralelo
 - **No se verificaron suficientemente los resultados**
- Referencias:
 - http://www.computerworld.com.au/article/32990/swoosh_stumbles
 - http://www.cio.com/article/32334/Nike_Rebounds_How_and_Why_Nike_Recovered_from_Its_Supply_Chain_Disaster

Seguridad: London Ambulance Dispatching System

- 1.2 millones (libras). Gestión de llamadas y despacho de ambulancias automatizado (90's)
 - Objetivo: Eliminación de radio y teléfono a las estaciones
 - **Resultado: Estimación de 20 muertos debido a los retrasos en la llegada de ambulancias**
 - Problema: Las llamadas se ponían en espera a veces hasta 30 minutos. A veces se perdían. Llamadas duplicadas. La sala de control estaba sobrecargada, problemas de uso del nuevo sistema
- Lecciones:
 - Implantación y puesta en marcha apresurada.
 - Inexperiencia de proveedor con un sistema tan grande
 - No había un sistema de backup. Ausencia de paralelo
 - **Interfaz de usuario muy moderno y llamativo, pero inadecuado cuando hay alto volumen de llamadas**
- Referencias
 - <http://catless.ncl.ac.uk/Risks/13.88.html#subj1.1>
 - <http://catless.ncl.ac.uk/Risks/14.37.html#subj11.1>

Error programación/prueba: Ariane 5 (June 1996)

- 10 años, \$7 billones. Agencia Espacial Europea (ESA)
 - Objetivo: capacidad de poner dos satélites de tres toneladas en órbita. Evolución del Ariane 4
 - Resultado: Junio 1996, a los 39 segundos del lanzamiento se desintegra, perdiendo dos caros satélites (sin seguro)
 - **Problema: A los 36,7 segundos el software de guiado inercial produce overflow al convertir 64 bit (float) a 16 bit (int).** Caída del sistema de backup, a los 0.05 segundos caída del sistema principal. El sistema de control principal recibe datos de diagnóstico que interpreta como datos de vuelo (ha cambiado la posición del cohete). Fuerte reacción para corregir la trayectoria. Desintegración por la fuerza aerodinámica. Autodestrucción.
- Causas:
 - El sistema de guiado inercial es el mismo que en Ariane 4, no se prevé protección contra este overflow porque estas comprobaciones se hacen en otro lugar y no se había producido nunca.
 - La velocidad horizontal de Ariane 5 es cinco veces superior y causa el overflow.
 - **Las especificaciones y pruebas realizadas no tuvieron en cuenta esto.**
- Referencias
 - <http://catless.ncl.ac.uk/Risks/18.29.html#subj16.1>
 - <http://www.di.unito.it/~damiani/ariane5rep.html>

Problemas en el desarrollo

■ IBM Survey, 1997

- **55% de los sistemas cuestan más de lo esperado**
- **68% tardan más tiempo de lo previsto**
- 88% tienen que ser sustancialmente rediseñados



■ Bureau of Labor Statistics (1999)

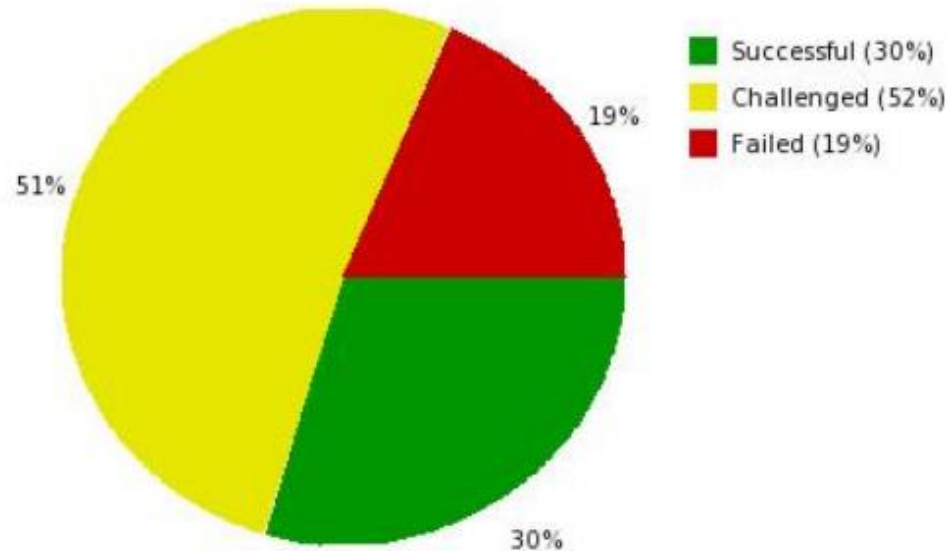
- Por cada 6 nuevos sistemas puestos en operación, 2 cancelados
- Probabilidad de cancelación alrededor de 50% en grandes sistemas
- Promedio de retraso en el plazo de proyectos del 50%

■ Otros estudios

- Callo & Wireless; 90% de proyectos no alcanzan objetivos. 40% de proyectos fracasados, 29% proyectos no entregados
- Gastos adaptación año 2000: \$600.000 Millones (Gartner). Coste demandas y litigios superior al anterior
- CSI, España: 1 de cada tres proyectos se “arrugan”. Menos del 20% consiguen entregarse en plazo y con satisfacción de cliente

¿Ha cambiado algo?

No mucho. Chaos Report 2018.



Challenged projects are projects that were completed but late, over budget, and did not meet the target specifications. Failed projects were cancelled or not used.



Abramos un paréntesis...

- Hemos hablado de errores “ocultos” en productos software que pueden causar graves consecuencias, y de problemas en el desarrollo de software.
- Los medios de comunicación dan, en ocasiones, publicidad a errores relacionados con la informática denominándolos “errores informáticos”.
- ¿Qué es un error informático?

En prensa

INICIO | **TECNOLOGÍA** | CIENTOS MILLONES DE COCHES PUEDEN SER 'HACKEADOS' POR UN ERROR INFORMÁTICO

Cien millones de coches pueden ser 'hackeados' por un error informático

TINDER ›

Dos ingenieros españoles descubren un error que permitía localizar a los usuarios de Tinder

El fallo revelaba la ubicación exacta de las personas que se daban 'match'. La posición se actualizaba cuando abrían la 'app'

MADRID • Suceso

Colapso en el Metro de Madrid por un error informático

SOCIEDAD

Un error informático afecta a decenas de miles de divorcios en Reino Unido

... muchas veces el problema no es la informática...

LA VANGUARDIA | Vida

Al Minuto Internacional Política Opinión Vida Deportes Economía Local Gente Cultura Sucesos Temas

Vida Natural Big Vang Tecnología Salud VangData Qué estudiar Bienestar Ecología Catalunya Religión Más ▾

UN FALLO MUY CARO

Destruye accidentalmente toda su empresa por un error informático

Un proveedor de dominios web borró de forma permanente los datos en los servidores de su compañía y la de sus 1.535 clientes



Lo + Visto

La historia detrás de una de las más icónicas y dramáticas fotos del 11S en Nueva York

El drástico cambio físico de Anna Simón alarma en las redes sociales

¿Qué es un error informático?

- ¿Un fallo no detectado en software en producción?
- ¿El uso inadecuado de software por parte de un usuario?
- ¿Un fallo de red que impide ejecutar un proceso en un servidor?
- ¿Tener tus sistemas parados durante dos horas porque los administradores están eliminando un virus?
- ¿Perder los datos de los clientes de tu empresa y que no tengas copia de seguridad?
- ¿Todo lo anterior?
- **¿Existen errores en otros ámbitos? Por ejemplo ¿hay “errores arquitectónicos” o “errores electrónicos” o “errores industriales”?**

En todos lados cuecen habas

El Supremo condena a Calatrava a pagar 2,96 millones por fallos en el Palacio de Congressos

Desestima el recurso del arquitecto contra el fallo de la Audiencia Provincial y le obliga a indemnizar a Jovellanos XXI por su «conducta negligente» durante las obras

TECNOLOGÍA • Parada provisional de la venta

Samsung suspende las ventas del Galaxy Note 7 por la quema de terminales durante la carga de batería

Bajo investigación un posible defecto de fábrica en las puertas del Ford Focus

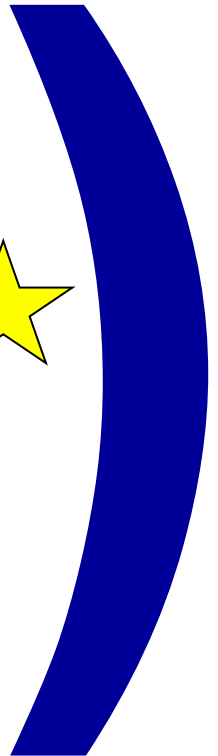
... no es fácil admitir que te equivocas

Una empresa de demoliciones tira abajo la casa incorrecta y culpa a Google Maps del error (y no es la única)



... cerremos el paréntesis

- Los profesionales de la ingeniería informática obtenemos los mejores productos y ofrecemos los mejores servicios posibles para nuestros clientes.
- En el ámbito del desarrollo de software, eso se logra **aplicando principios de ingeniería (ingeniería del software)**.
- Tras cursar esta asignatura estaréis capacitados para aplicar prácticas, técnicas y metodologías de ingeniería al desarrollo de software.
- Hoy empezamos por entender cuáles son las problemáticas particulares de esta ingeniería y sus características.



Problemática General



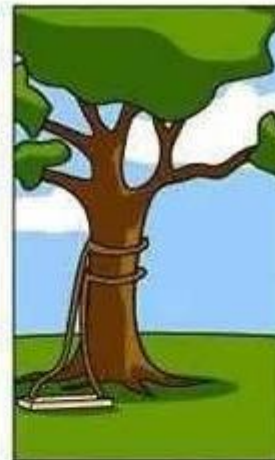
La solicitud del usuario



Lo que entendió el líder del proyecto



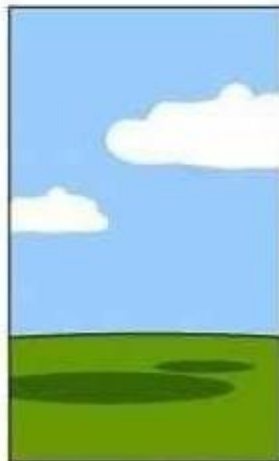
El diseño del analista de sistemas



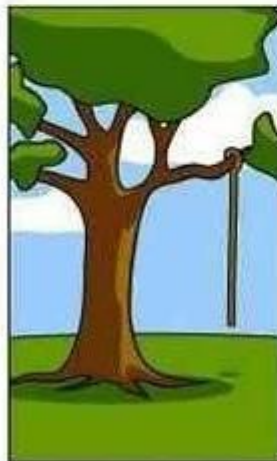
El enfoque del programador



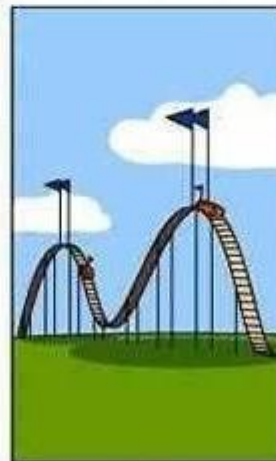
La recomendación del consultor externo



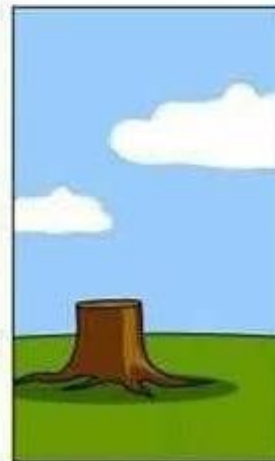
La documentación del proyecto



La implantación en producción



El presupuesto del proyecto



El soporte operativo



Lo que el usuario realmente necesitaba

Características del Software

Instrucciones + Estructuras de Datos + Documentación

■ Qué es el software?

- Definición (Art. 9, R.D. 1/1996 Texto refundido de la Ley de Propiedad Intelectual)
 - "...toda secuencia de instrucciones o indicaciones destinadas a ser utilizadas, directa o indirectamente, en un sistema informático para realizar una función o una tarea o para obtener un resultado determinado, cualquiera que fuere su forma de expresión y fijación. A los mismos efectos, la expresión programas de ordenador comprenderá también su documentación preparatoria."
- **El software no se fabrica como otros productos. Los costes se concentran en la ingeniería, no en la producción**

■ Para qué sirve

- **Para solucionar los problemas de los USUARIOS**
 - Combinaciones de Desarrollo, Implantación, Integración
- El ingeniero del software debe **entender el problema y comunicarse de forma efectiva con el usuario/cliente**



Dimensiones y tipos de software

■ Objetivo:

- ☐ Tiempo real y comunicaciones
- ☐ Empresarial (Business software)
- ☐ Cálculo científico y sistemas inteligentes
- ☐ De ordenador personal

■ Soporte/Canales

- ☐ Empotrado (embedded)
- ☐ Aplicación de consola (ventanas)
- ☐ Web (navegador)
- ☐ Servicio Web (entre aplicaciones)
- ☐ Móvil/ubicuo (tabletas, smartphones)

■ Reutilización:

- ☐ COTS (Commercial off the shelf)
- ☐ A Medida
- ☐ Parametrizable. ERP (Enterprise Resource Planning)

■ Diferenciar las diferentes dimensiones

Naturaleza del software



- Software:
 - ☐ Intangible. Difícil de entender el esfuerzo de desarrollo
 - ☐ Fácil de reproducir y flexibilidad. Coste en desarrollo, diferencia de otros productos de ingeniería
 - ☐ Dependiente de la tecnología subyacente
- Proceso. Dificultades:
 - ☐ Definir los requisitos (cliente). Complejidad oculta al usuario
 - ☐ Gestionar: Dependencia de las personas y equipos. Difícil de automatizar
 - ☐ Probar: Problemas de calidad difíciles de detectar
 - ☐ Modificar: Comprender realmente el cambio y su impacto.
- Conclusiones:
 - ☐ **Hay que aprender a producirlo (“engineer software”)**
 - ☐ **Asegurar calidad, costes y plazos razonables**
 - ☐ ¿Crisis permanente?

Ingeniería del Software

- Término acuñado en 1968. Conf. Ing. Software de la OTAN



- Definición: **The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software** (IEEE STD 610.12-1990, IEEE Standard Glossary of Software Engineering Terminology, IEEE Computer Society, 1990).

- Diferencia con Computer Science: CS es teoría y fundamentos, SE se encarga de los aspectos prácticos del desarrollo e implantación.

- Otras definiciones de autores:

- I Sommerville. “Software engineering is concerned with the theories, methods and tools for developing, managing and evolving software products”
- B.W.Boehm. “The practical application of scientific knowledge in the design and construction of computer programs and the associated documentation required to develop, operate and maintain them”
- F.L. Bauer. “The establishment and use of sound engineering principles in order to obtain economically software that is reliable and works efficiently on real machines”

Ingeniería del Software

- Definiciones ISO (International Organization for Standardization)
 - The systematic application of scientific and technological knowledge, methods, and experience to the design, implementation, testing, and documentation of software (ISO/IEC 2382-1:1993 Information technology - Vocabulary - Part 1).
 - The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software (ISO/IEC 24765:2009 Systems and software engineering vocabulary) (=IEEE)
- **Diccionario de Términos:**
 - **ISO/IEEE SEVOCAB: Software and Systems Engineering Vocabulary:** http://pascal.computer.org/sev_display/

Procesos, técnicas y herramientas

- Proceso de Software: Conjunto de actividades y productos obtenidos durante el desarrollo de un sistema software, independientemente de su tamaño o complejidad
- Técnica: Un procedimiento sistemático empleado por un recurso humano para realizar una actividad que produce un producto, resultado o servicio, y que puede emplear una o más herramientas (PMBOK)
 - Método: Un determinado enfoque (“approach”) para solucionar un determinado problema. Indican como construir técnicamente el software. Incluye:
 - Notación (gráfica y/o textual)
 - Técnicas que han de ser utilizadas para solucionar el problema
- Metodología: Conjunto de prácticas, técnicas, procedimientos y reglas usados por quienes trabajan en una determinada disciplina (PMBOK).
 - Define la secuencia en la que se aplican los métodos, productos o documentos requeridos, controles de calidad, etc (SEI)
- Herramienta: Producto empleado para automatizar procesos o métodos
 - CAD/CAM (Computer Aided Design/Manufacturing)
 - CASE (Computer Aided Software Engineering)



Procesos genéricos



- **Ciclo de vida: Cómo se ordenan y coordinan los diferentes procesos**
 - A framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding (ISO/IEC 12207:2008 Systems and software engineering - Software life cycle processes)
 - Fases (etapas) que se secuencian, solapan o iteran (según un plan)
 - Secuencial o cascada (waterfall):
 - Difícil de conocer los requisitos al principio
 - Sólo se dispone de una versión operativa al final
 - Otros enfoques: incremental, evolutivo, modular, espiral, prototipado, ágil, etc.

Procesos principales (ISO/IEC 12207:2008)

■ Procesos a nivel de Sistema

- ☐ Stakeholder Requirements Definition
- ☐ System Requirements Analysis
- ☐ System Architectural Design
- ☐ Implementation
- ☐ System Integration
- ☐ System Qualification Testing
- ☐ Software Installation
- ☐ Software Acceptance
- ☐ Software Operation
- ☐ Software Maintenance
- ☐ Software Disposal



■ Procesos de Implementación de Software

- ☐ Requirements Analysis
- ☐ Software Architectural Design
- ☐ Software Detailed Design
- ☐ Software Construction
- ☐ Software Integration
- ☐ Software Qualification Testing

System Context Processes

Agreement Processes

Acquisition Process
(Clause 6.1.1)

Supply Process
(Clause 6.1.2)

Organizational Project-Enabling Processes

Life Cycle Model Management Process
(Clause 6.2.1)

Infrastructure Management Process
(Clause 6.2.2)

Project Portfolio Management Process
(Clause 6.2.3)

Human Resource Management Process
(Clause 6.2.4)

Quality Management Process
(Clause 6.2.5)

Project Processes

Project Planning Process
(Clause 6.3.1)

Project Assessment and Control Process
(Clause 6.3.2)

Decision Management Process
(Clause 6.3.3)

Risk Management Process
(Clause 6.3.4)

Configuration Management Process
(Clause 6.3.5)

Information Management Process
(Clause 6.3.6)

Measurement Process
(Clause 6.3.7)

Technical Processes

Stakeholder Requirements Definition Process
(Clause 6.4.1)

System Requirements Analysis Process
(Clause 6.4.2)

System Architectural Design Process
(Clause 6.4.3)

Implementation Process
(Clause 6.4.4)

System Integration Process
(Clause 6.4.5)

System Qualification Testing Process
(Clause 6.4.6)

Software Installation Process
(Clause 6.4.7)

Software Acceptance Support Process
(Clause 6.4.8)

Software Operation Process
(Clause 6.4.9)

Software Maintenance Process
(Clause 6.4.10)

Software Disposal Process
(Clause 6.4.11)

Software Specific Processes

SW Implementation Processes

Software Implementation Process
(Clause 7.1.1)

Software Requirements Analysis Process
(Clause 7.1.2)

Software Architectural Design Process
(Clause 7.1.3)

Software Detailed Design Process
(Clause 7.1.4)

Software Construction Process
(Clause 7.1.5)

Software Integration Process
(Clause 7.1.6)

Software Qualification Testing Process
(Clause 7.1.7)

SW Support Processes

Software Documentation Management Process
(Clause 7.2.1)

Software Configuration Management Process
(Clause 7.2.2)

Software Quality Assurance Process
(Clause 7.2.3)

Software Verification Process
(Clause 7.2.4)

Software Validation Process
(Clause 7.2.5)

Software Review Process
(Clause 7.2.6)

Software Audit Process
(Clause 7.2.7)

Software Problem Resolution Process
(Clause 7.2.8)

Software Reuse Processes

Domain Engineering Process
(Clause 7.3.1)

Reuse Program Management Process
(Clause 7.3.3)

Reuse Asset Management Process
(Clause 7.3.2)

Metodologías Típicas

- Métrica Versión 3 – Metodología de Planificación, Desarrollo y Mantenimiento de sistemas de información
- Proceso Unificado
- MEDEPA – Metodología de Desarrollo de Proyectos del Principado de Asturias
- Ágiles:
 - XP – Extreme Programming
 - SCRUM

Resumen



- **El software es un producto de ingeniería. Debe construirse como tal**
 - Existen particularidades que hacen más difícil el proceso
 - Suele infravalorarse la dificultad de desarrollo, implantación y mantenimiento
 - Debe responder a las necesidades de los usuarios
 - Es preciso entender el problema, comunicar e involucrar al usuario
 - No siempre es desarrollo
 - No olvidar los procesos de gestión (Calidad, Planificación)
- **Conocer la problemática y los procesos de Ingeniería del Software es importante para:**
 - El proveedor (Técnico, gestor y comercial)
 - El cliente final

**Es muy importante saber comunicar:
hablar, escuchar, leer, escribir.**



Bibliografía

- Pressman, R.S., Ingeniería del software, Un enfoque práctico (7ª ed)
 - Cap. 1-2
- Metodología de planificación y desarrollo de sistemas de información MÉTRICA, Versión 3 – Documento de introducción.
- The Risks Digest: Forum On Risks To The Public In Computers And Related Systems (<http://catless.ncl.ac.uk/Risks>)