

Ingeniería del software

0 Introducción

Juan Rodríguez Hortalá

1

2

Referencias

- Pressman, R.S. *Ingeniería del Software. Un Enfoque Práctico. Sexta Edición*. McGraw-Hill, 2005
- Pankaj Jalote. *A Concise Introduction to Software Engineering*. Springer, 2008

Acerca de la asignatura

Juan Rodríguez Hortalá

juanrh@fdi.ucm.es
despacho 220

tutorías: X-V de 14:00 a 16:00
material docente y comunicación a través
del **Campus Virtual**

grupos B y C de Ingeniería en Informática

3

Objetivos de la Ingeniería del Software

4

El Producto Software

No es **solamente** el **código** que implementa el sistema, también incluye

Documentación para los desarrolladores encargados del mantenimiento

Manuales de usuario

Instaladores e infraestructura de despliegue

Datos e información de configuración en que se apoya el SW ej: bases de datos en las que se apoya un sistema de inventario de un almacén

5

3 C's de la Ingeniería del Software

En un sistema SW industrial buscamos

↓ **Calendario** construir el SW a tiempo

↓ **Coste** lo más barato posible

↑ **Calidad** un buen producto

6

Productividad

Productividad = producto de salida por unidad de entrada

Una medida de la productividad

Entrada = tiempo de trabajo del personal

Salida = KLOC (kilo lines of code)

--> productividad = **KLOC/persona/mes**

Importancia de las **métricas cuantitativas** base para las estimaciones de coste. Métricas vs. opiniones de las personas (optimismo/pesimismo, humor o percepciones subjetivas según la carga de trabajo actual o ambiente de trabajo)

La productividad permite abaratar costes y cumplir con el calendario sin sacrificar la calidad: las 3 Cs se resumen en **Q&P** (quality & productivity)

7

¿Como conseguir la productividad?

productividad

=

1. **efectividad** Qué: elegir bien el objetivo

+

2. **eficiencia** Cómo: alcanzar el objetivo con rapidez

Primero efectividad, luego eficiencia un médico amputa una pierna a un paciente en la mitad de tiempo de lo habitual, y luego presume de su gran eficiencia. Los enfermeros no piensan lo mismo, puesto que ha amputado la pierna equivocada: productividad cero

8

Calidad

Visibilidad interna vs externa

- **externa** la que ven los usuarios: aspectos relacionados con la funcionalidad del sistema, lo que nos han prometido que hace
- **interna** la que ven los desarrolladores encargados del mantenimiento de la aplicación, que deben: corregir bugs, entender el SW, instalarlo e integrarlo con otros sistemas SW

La calidad se descompone en varios **atributos** con visibilidad interna, externa o ambas

- **Funcionalidad** ¿hace lo que se había especificado (implícita o explícitamente)?
- **Robustez** tolerancia a fallos
- **Usabilidad** facilidad para entenderse, aprenderse y utilizarse
- **Eficiencia** rendimiento relativo a los recursos empleados
- **Mantenibilidad** capacidad de modificación: para corregir errores, añadir mejoras, adaptar a cambios del entorno
- **Portabilidad** facilidad de adaptación a otros entornos y plataformas

10

Calidad: robustez

Robustez uno de los atributos más importantes

- densidad de defectos = defectos/KLOC. Sirve como una medida simple
- no todos defectos se pueden detectar al construir el producto muchos aparecen con el tiempo y el uso --> se suele medir la densidad de defectos tras 6 meses/1 año del release. Esa medida es útil para estimar la calidad de proyectos futuros
- ¿qué se considera un defecto? según el estándar de la organización que desarrolló el producto, influida por la presión de los clientes, usuarios o rivales. Ej: antena del iphone

Calidad: mantenibilidad

Mantenibilidad el otro atributo fundamental

- fase de mantenimiento periodo que empieza tras desplegar o distribuir el SW
- ¿porqué es necesario el mantenimiento si el SW no tiene componentes físicas que se puedan estropear con el tiempo?
 - *mantenimiento correctivo* la densidad de defectos 0 es imposible. El mantenimiento correctivo se ocupa de arreglar los bugs que se descubren con el tiempo
 - *mantenimiento adaptativo* para satisfacer las demandas de un entorno siempre cambiante
 - demandas de los usuarios o del mercado: más funcionalidades
 - debidas a cambios en otros sistemas SW con los que se acopla nuestro producto
- el cambio ocurrirá --> preverlo durante el diseño para saber adaptarse más adelante**

- se suele gastar más en mantenimiento que en el desarrollo original del producto SW ratio 80/20, 70/30 o 60/40 como mucho --> mejor obtener una buena mantenibilidad desde la construcción del SW

11

Otros factores: escala

El SW industrial tiende a ser muy grande y complejo

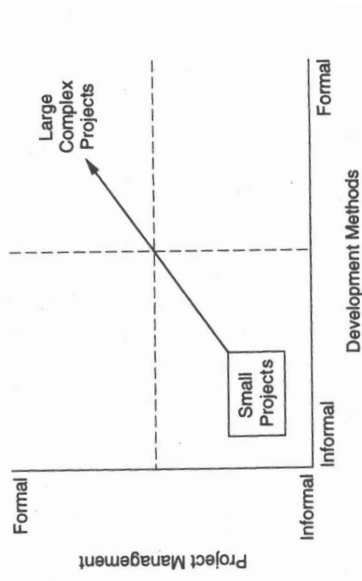
KLOC	Software	Lenguajes	Soluciones diferentes para un mismo problema según la escala a la que se aborde
980	gcc	ansic, cpp, yacc	
200	openssl	ansic, cpp, perl	
100	apache	ansic, sh	
30.000	Red Hat Linux	ansic, cpp	
40.000	Windows XP	ansic, cpp	

Censo de una clase vs censo de un país: el problema es el mismo, contar, pero los métodos son muy diferentes

12

Otros factores: escala

De la misma manera se utilizan métodos distintos para proyectos SW de distinto tamaño.



- proyectos pequeños más informalidad, demasiada burocracia provoca retrasos innecesarios
- proyectos grandes mayor necesidad de coordinación y comunicación formal (lo que implica más documentación) para ser productivos

13

Otros factores: interoperabilidad

Muchos sistemas SW de nueva creación deben diseñarse teniendo en cuenta que deben

- comunicarse con SW anterior
- adaptarse a las restricciones y limitaciones del SW anterior

ej: sistemas para acceso web o mediante dispositivos móviles a un banco: deben comunicarse con el núcleo del banco, usualmente programado en lenguajes como COBOL

14

Procesos

3 factores para lograr Q&P en SW

- **personal** los que finalmente escriben el SW
- **herramientas CASE** (Computer Aided SW Engineering)
- **procesos** secuencia de pasos que debe realizarse para alcanzar un objetivo

Los procesos especifican qué tareas debe hacer el personal y cómo hacerlas. Las herramientas automatizan o ayudan en la consecución de algunas tareas, ayudando a implementar los procesos

La IS se centra en los procesos que son recetas para optimizar el empleo del personal y herramientas disponibles para maximizar la Q&P

15

Procesos Software

16

Múltiples procesos

Proceso = secuencia de pasos para alcanzar un objetivo

+

En un proyecto SW hay varios objetivos que alcanzar

en un proyecto SW se realizan **múltiples procesos a la vez**

Procesos no SW se ocupan de objetivos no SW

- procesos de negocio (estrategia de empresas, negociaciones con otras empresas, búsqueda de mercados, ...), procesos sociales (relaciones entre los empleados, jerarquías, ...), procesos de entrenamiento, reclutamiento de personal ...
- afectan al desarrollo del SW pero no están dentro del campo de estudio de la IS, aunque son vitales en la vida laboral dentro de la empresa

17

Procesos de ingeniería de producto

El objetivo conjunto de estos procesos es la construcción del producto SW objetivo del proyecto

Proceso de desarrollo especifica las **actividades de ingeniería** que hay que llevar a cabo para obtener el producto SW

- los *modelos de proceso de desarrollo* (cascada, prototipo, incrementales, ...) establecen un esquema que seguir para establecer dichas actividades

Proceso de gestión de proyectos especifica cómo **planificar** y **controlar** las actividades del proceso de desarrollo para maximizar las 3 C's

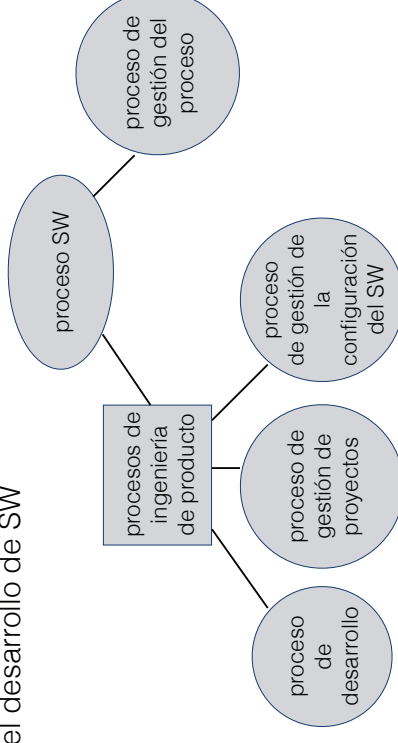
Proceso de gestión de la configuración del SW gestiona el **cambio** para que se siga preservando la integridad de los productos tras los cambios

- durante el desarrollo del proyecto se van generando muchos productos (código, documentación ...) que van evolucionando creándose muchas versiones. El proceso de gestión de configuración organiza estos productos y su versionado

19

Múltiples procesos

Procesos SW se ocupan de las tareas **técnicas** y de **gestión** del desarrollo de SW



- No hay una **nomenclatura unificada** en la literatura sobre IS: en muchos textos se llama "proceso de SW" o simplemente "proceso" al proceso de desarrollo
- A cada hoja del árbol más arriba lo llamamos *proceso SW componente*

18

Proceso de gestión del proceso

El proceso SW en sí mismo debe cambiar para adaptarse

- según la organización que lo implanta lo va depurando y adaptando al ir comprendiéndolo mejor al aplicarlo, o por cambios organizacionales
- por la aparición de nuevas metodologías o tecnologías y herramientas que pudieran mejorar la productividad del proceso

El proceso de gestión de proceso se encarga de **mejorar** la capacidad del **proceso SW** para obtener Q&P para ello se estudia el proceso SW actual, estudiando proyectos pasados llevados a cabo con ese proceso: entender el proceso SW, analizar sus propiedades, encontrar mejoras que aplicar

Otra dimensión de la calidad hasta ahora sólo habíamos considerado calidad del producto, el proceso de gestión de proceso se ocupa de la **calidad del proceso** SW mismo

20

Organización el cuatrimestre

21

Contenidos

En el **primer cuatrimestre** se trató

- Introducción a la ingeniería del software
- Modelos de proceso de desarrollo
- Análisis y especificación de requisitos
- Diseño y modelado de software

En este **segundo cuatrimestre** veremos

- Planificación y gestión de proyectos
- Implementación y validación
- Desplegado y mantenimiento de aplicaciones
- Temas avanzados

22