

X Reunión de Responsables de Sistemas de Información



La Antigua, Guatemala
22-25 de septiembre de 2008



Calidad del Software



José Luis Fuertes Castro
Facultad de Informática
Universidad Politécnica de Madrid




POLITÉCNICA

Contenidos

1. Introducción
2. Control y Garantía de la Calidad del Software
 1. Controles estáticos y dinámicos
 2. Actividades constructivas de la calidad
 3. El coste de la calidad
3. Medidas
 1. Introducción a las medidas
 2. Qué, por qué y cómo miden los ingenieros
 3. Medidas y métricas
 4. Medidas en Ingeniería del Software
 5. Clasificación de medidas
4. Calidad del Software
 1. Introducción a la calidad
 2. Modelos de calidad del software
 3. Mejora del proceso software
5. Conclusiones

Calidad del Software *José Luis Fuertes*

Introducción



Introducción

✦ Software:

- Especifica el conjunto de programas informáticos que se desarrollan en el entorno de una computadora
- Papel del software en la sociedad ha cambiado significativamente en los últimos 50 años
 - ◆ mejoras en rendimiento del hardware
 - ◆ cambios en arquitecturas informáticas
 - ◆ aumentos de memoria y capacidad de almacenamiento
 - ◆ variedad de dispositivos de entrada y salida
 - ◆ ...
- Importancia profesional y personal de la Informática
 - ◆ Disminuye el precio del *hardware*
 - ◆ Aumenta la demanda de nuevas aplicaciones

Calidad del Software *José Luis Fuertes*

Introducción

- ✦ Sistemas cada vez más sofisticados y complejos
- ✦ Se ha pasado de ser una herramienta para la resolución de problemas a una industria
- ✦ Características del software
 - El software se desarrolla, no se fabrica
 - El software no se estropea
 - Aunque la industria tiende a ensamblar componentes, la mayoría del software se construye a medida

Calidad del Software *José Luis Fuertes*

Introducción

- ✦ Un poco de historia...
 - Años 40 → Fundamentos de la computación moderna
 - Principios de los 60 → Consolidación del desarrollo de software de aplicaciones relativamente grandes
 - Finales de los 60 → Crisis del software
 - ◆ Imposibilidad de desarrollar proyectos de una dimensión acorde con los avances tecnológicos de la época
 - 1968 → El comité científico de la OTAN patrocina una conferencia en Alemania, para identificar, clasificar y discutir los problemas que se producían en el desarrollo de grandes proyectos
 - ◆ Resultado: el desarrollo del software es una tarea de ingeniería (se acuña el término de Ingeniería del Software)

Calidad del Software *José Luis Fuertes*

Introducción

+ Ingeniería:

- Diccionario de la Real Academia Española:
"Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía"



+ Software:

- Diccionario de la Real Academia Española:
"Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora"



Introducción

+ Ingeniería del Software:

- Boehm, 1976



"Aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar y mantenerlos"

- IEEE, 1993



"Aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software"

Introducción

+ El producto software:

- Es un sistema software desarrollado para un cliente con la documentación que describe cómo instalar y usar el sistema
- Objetivo de la Ingeniería del Software
 - ◆ Producir productos software

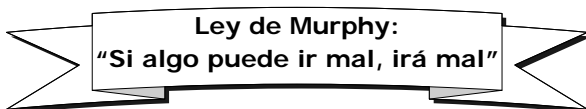
Introducción

+ El proceso software:

- Es el conjunto de actividades y resultados que permite producir un producto software
- Actividades fundamentales del proceso software:
 - ◆ Especificación del software: se define la funcionalidad del software y el modo de operación
 - ◆ Desarrollo del software: se produce el software de acuerdo con las especificaciones
 - ◆ Validación del software: el software debe ser validado para asegurarse que hace lo que espera el cliente
 - ◆ Evolución del software: el software debe evolucionar para acomodarse a las nuevas necesidades del cliente
- Actividad compleja, muchas actividades distintas

Introducción

+ El software no es perfecto



Introducción

+ Las causas de los fallos informáticos pueden ser muy diversas:

- Especificaciones incorrectas o incompletas
- Análisis equivocados
- Diseños con fallos
- Programación con errores
- Validación poco precisa

Introducción

- ✦ Actualmente se usa el software en gran diversidad de ámbitos
- ✦ Los defectos del software pueden causar importantes problemas e, incluso, daños físicos
 - Los defectos en programas financieros o editores son poco importantes, poco costosos y nadie sale herido
 - Cuando el software “pilota” aviones, “conduce” autos, “controla” el tráfico aéreo o “vigila” centrales nucleares, los defectos pueden ser muy peligrosos

Introducción

- ✦ La solución al problema es la **calidad**
 - ¿Cuándo tiene buena calidad el software?
 - ¿Qué características influyen al determinar la calidad del software?
 - ¿Cómo medir la calidad del software?
 - ¿Cómo controlar y garantizar la calidad del software?

Introducción

- ✦ Es necesario medir la calidad...
...pero es una tarea difícil
- ✦ ¿Por qué medir?

“Si algo puede ser medido y expresado con números, entonces se sabe algo acerca de ello”



Kelvin

“No se puede controlar lo que no se puede medir”



DeMarco

Control y Garantía de la Calidad



Control de la Calidad

- ✦ Objetivos:
 - Comprobar si un producto posee una determinada característica de calidad en el grado requerido
 - ◆ Si no es así, tiene un **defecto**
 - Identificar defectos y corregirlos
- ✦ Actividades de control:
 - Controles estáticos
 - ◆ Analizan el producto sin necesidad de ejecutarlo
 - Controles dinámicos
 - ◆ Analizan el producto durante su funcionamiento

Control de la Calidad

Controles Estáticos

- ✦ Controles manuales
 - Informales
 - ◆ Por los autores del desarrollo o por personal equivalente
 - Comprobación de escritorio (*desk checking*)
 - Revisión por pares (*peer review*)
 - Disciplinados
 - ◆ Son técnicas de grupo
 - Auditorías
 - Revisiones
- ✦ Controles automáticos
 - Sistemas informáticos o con un proceso algorítmico
 - ◆ Análisis estático automático
 - ◆ Verificación formal

Control de la Calidad

Controles Dinámicos

- ✦ Tipos de pruebas
 - Prueba **modular**
 - ◆ Se prueba cada módulo aislado del resto del sistema
 - Prueba de **integración**
 - ◆ Se prueba que las interfaces entre los módulos son correctas
 - Prueba de **sistema**
 - ◆ Se prueba si el sistema satisface todos los requisitos del usuario
 - Prueba de **aceptación**
 - ◆ Su objetivo es demostrar al usuario que el sistema satisface sus necesidades
 - Prueba de **regresión**
 - ◆ Se comprueba que la nueva versión del producto no es de menos calidad que la anterior

Garantía de Calidad

- ✦ Definición
 - Conjunto de actividades de planificación, estimación y supervisión del proceso de desarrollo, que se realizan de forma independiente al equipo de desarrollo, de tal forma que los productos software resultantes cumplen los requisitos establecidos y satisfacen los niveles de calidad exigidos

Garantía de Calidad

Actividades Constructivas

- ✦ Modelos de proceso software
 - Idealización del proceso de desarrollo y mantenimiento del software
 - Se descompone el proceso en fases, definiendo las actividades a realizar
- ✦ Métodos y formalismos
 - La sistematización de los procedimientos facilita la prueba de los resultados obtenidos
 - Se producen incrementos en la productividad (50-150%)

Garantía de Calidad

Actividades Constructivas

- ✦ Herramientas y entornos de desarrollo
 - Las herramientas CASE (*Computer Aided Software Engineering*) facilitan la producción de software
 - Los modernos entornos de desarrollo han simplificado el proceso de implementación
- ✦ Lenguajes de programación
 - Su importancia (en términos de calidad) se hace patente en el mantenimiento
 - Características interesantes: módulos, compilación separada, ocultación de información, flujo de control estructurado, fuerte tipado, OO, orientación a eventos...

Garantía de Calidad

Actividades Constructivas

- ✦ Documentación
 - Juega un papel esencial
 - Características importantes: consistencia, claridad, comprensibilidad, completitud...
- ✦ Factores humanos
 - Factores a considerar: cultura de la organización, comunicación, entorno físico, formación, motivación, dirección, liderazgo...
- ✦ Otros
 - Estándares y convenciones, ejemplos y patrones, gestión de configuración, control del código, gestión de problemas...

Garantía de Calidad

El Coste de la Calidad

- ✦ Intervienen dos factores:
 - Prevención de errores
 - Detección de defectos

Distribución de defectos detectados según la fase en la que se cometieron:

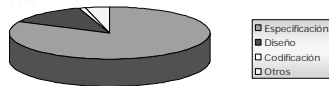


Garantía de Calidad El Coste de la Calidad

† Intervienen dos factores:

- Prevención de errores
- Detección de defectos

Porcentaje, sobre el coste total de corrección de los defectos, según la fase en la que se cometieron:

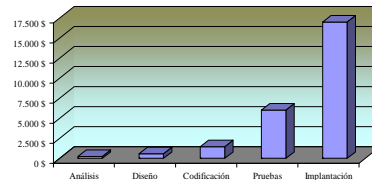


Garantía de Calidad El Coste de la Calidad

† Intervienen dos factores:

- Prevención de errores
- Detección de defectos

Coste de corrección según la fase en la que se detectaron los defectos:



Medidas



Introducción a las Medidas Definiciones

† ¿Qué es la medición?

- La medición es la asociación de valores numéricos con un objeto o acción
- Se interpreta este valor como la cantidad de calidad o de cierto atributo poseído por dicho objeto o acción

Introducción a las Medidas Definiciones

† Medir

- Diccionario de la Real Academia Española:
 - ◆ Comparar una cantidad con su respectiva unidad, con el fin de averiguar cuántas veces la segunda está contenida en la primera

† Medida

- Diccionario de la Real Academia Española:
 - ◆ Acción y efecto de medir
 - ◆ Expresión del resultado de una medición
 - ◆ Cada una de las unidades que se emplean para medir longitudes, áreas o volúmenes de líquidos o áridos

Medidas ¿Qué Miden los Ingenieros?

† La ingeniería puede definirse como el proceso que produce productos útiles

† Puede describirse lo que miden los ingenieros en dos categorías:

- Medidas del producto
- Medidas del proceso

Medidas

¿Qué Miden los Ingenieros?

- † Medidas del producto
 - Medidas estáticas
 - ◆ Pueden calcularse cuando el objeto no está en uso
 - ◆ Tamaño, longitud, altura, anchura, peso, capacidad, volumen...
 - Medidas dinámicas
 - ◆ Describen el comportamiento de un objeto mientras está en uso
 - ◆ Velocidad, consumo, disipación de calor, nivel de ruido...

Medidas

¿Qué Miden los Ingenieros?

- † Medidas del proceso
 - Se usan para cuantificar la actividad humana de la ingeniería
 - Ejemplos:
 - ◆ Tamaño de un equipo de desarrollo, esfuerzo, tiempo, coste, productividad...

Medidas

¿Por qué Miden los Ingenieros?

- † Para describir el estado actual del mundo
 - Cada medida describe un aspecto concreto del estado del mundo hoy
 - Si se mide el estado del mundo periódicamente, puede ser posible descubrir patrones y tendencias
 - ◆ Permite dar explicaciones del comportamiento del mundo real
 - Leyes o teorías científicas
 - Los ingenieros pretenden descubrir:
 - ◆ el comportamiento de los sistemas que diseñan y construyen
 - ◆ las tendencias en el propio proceso de ingeniería

Medidas

¿Por qué Miden los Ingenieros?

- † Para establecer los requisitos cuantitativamente y demostrar su cumplimiento
 - No puede imaginarse un proyecto de ingeniería sin requisitos cuantitativos
 - ◆ Construcción de un puente (longitud, altura, carga máxima, caudal del río...)
 - ◆ Pequeños electrodomésticos (tamaño, peso, coste...)
 - ◆ Vehículos (peso, consumo, espacio interior, espacio para equipaje, resistencia a los impactos...)
 - ◆ Software (tiempo de respuesta, memoria ocupada...)
 - Para demostrar que se cumplen los requisitos cuantitativos, es necesario medir

Medidas

¿Por qué Miden los Ingenieros?

- † Para realizar un seguimiento del progreso y predecir los resultados
 - Las mediciones periódicas de lo que se va obteniendo permite realizar un seguimiento cuantitativo del proyecto
 - ◆ Pueden ponerse de manifiesto tendencias inusuales que, al detectarlas, pueden corregirse
 - Los ingenieros del software cuentan el número de defectos encontrados durante las pruebas para calibrar los modelos de fiabilidad
 - ◆ Pueden predecir cuándo se completarán las pruebas y cuándo se alcanzará el nivel de fiabilidad requerido

Medidas

¿Por qué Miden los Ingenieros?

- † Para analizar costes y beneficios
 - El corazón de la ingeniería: el compromiso
 - ◆ Hay muchas formas de diseñar productos y muchas formas de diseñar sus componentes
 - ◆ Cada diseño tiene ventajas y desventajas
 - ◆ El ingeniero debe valorarlas todas y establecer un compromiso entre unas y otras
 - A veces se acepta una cualidad negativa para favorecer otra cualidad
 - A veces se acepta menos de un atributo deseable para obtener más de otro atributo deseable
 - ★ Ej.: Coche: se deben establecer compromisos entre peso, economía, espacio del habitáculo, confort y precio

Medidas

¿Cómo Miden los Ingenieros?

- † Tradicionalmente, los ingenieros usan instrumentos
- † Los instrumentos pueden proporcionar medidas erróneas
 - La exactitud de un instrumento es una indicación de la diferencia entre la lectura del instrumento y los datos de entrada
 - La precisión de un instrumento es una indicación de lo repetible que resulta una medición con una exactitud determinada

Medidas

¿Cómo Miden los Ingenieros?

- † Muestreo
 - Selección y medida de una parte representativa de la población para deducir parámetros o características de toda la población.
 - ♦ Se usa cuando es imposible o poco práctico medir a toda la población
 - Ejemplo
 - ♦ Control de calidad en un proceso de fabricación
 - Se elige uno de cada 1000 productos y se realizan medidas
 - Utilizando varias técnicas estadísticas, se extraen conclusiones acerca de todos los productos a partir de las medidas realizadas en los productos seleccionados

Teoría de Medidas

Medidas y Métricas

- † Informalmente:
 - Medida
 - ♦ Forma perfectamente definida de asociar valores numéricos a los atributos de los elementos de una entidad
 - Métrica
 - ♦ Criterio para determinar la diferencia o distancia entre dos entidades

Teoría de Medidas

Medidas y Métricas

- † Matemáticamente:
 - Medida
 - Sea A un conjunto de objetos físicos o empíricos
 - Sea B un conjunto de objetos formales (números)
 - Se define una **medida** μ como una relación unívoca de A a B :
 $\mu: A \rightarrow B$
 - ♦ Cada objeto tiene una y solo una medida

Teoría de Medidas

Medidas y Métricas

- † Matemáticamente:
 - Métrica:
 - Sea A un conjunto de objetos
 - Sea \mathfrak{R} el conjunto de números reales
 - Sea $m: A \otimes A \rightarrow \mathfrak{R}$ una medida
 - Entonces m es una **métrica** si cumple:
 - $m(x, y) \geq 0, \forall x, y \in A$
 - $m(x, y) = 0 \Leftrightarrow x = y$
 - $m(x, y) = m(y, x), \forall x, y \in A$
 - $m(x, z) \leq m(x, y) + m(y, z), \forall x, y, z \in A$

Teoría de Medidas

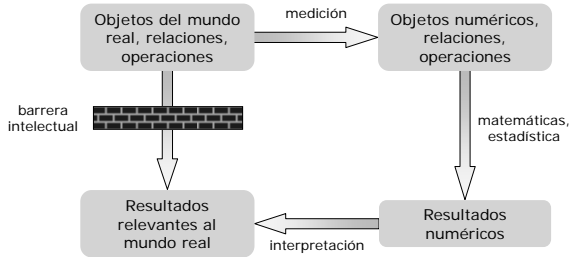
Medidas y Métricas

- † Este concepto de “métrica” como distancia entre dos entidades tiene muy poco sentido en el mundo del software
- † En cambio, sí tiene sentido calcular una medida para dos productos y compararlas

~~Métrica del Software~~ \Leftrightarrow Medida del Software

Teoría de Medidas

† Barrera intelectual



Medidas en Ingeniería del Sw.

- † ¿Es bueno un programa?
- † ¿Cómo de fiable será el sistema una vez instalado?
- † ¿Cuántos errores se esperan encontrar?
- † ¿Cuántas pruebas tengo que hacer?
- † ¿Cuál será el coste de las pruebas?
- † ¿Será difícil mantener el sistema?
- † ¿Cuánto costará construir un sistema similar al que construí hace 5 años?
- † ¿Cuánto tardaré en desarrollar el sistema?
- † ...

Medidas en Ingeniería del Sw.

- † Los ingenieros del software se enfrentan diariamente a estas preguntas
- † La habilidad para realizar las medidas apropiadas es fundamental para los ingenieros del software
- † También es importante saber cómo medir
- † Es un área de investigación candente en Ingeniería del Software
 - No hay resultados estándar universalmente aceptados en el terreno de las medidas del software

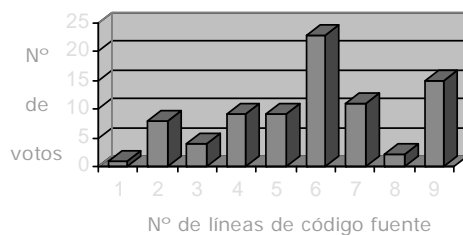
Medidas en Ingeniería del Sw.

- † Ejemplo de medida de tamaño del software
 - Líneas de código fuente (LOC)

```
#define LOWER 0 /* límite inferior de la tabla */
#define UPPER 300 /* límite superior */
#define STEP 20 /* tamaño del paso */

main () /* Tabla de conversión de Fahrenheit a Celsius */
{
    int fahr;
    for (fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
        printf ("%4d %6.1f\n", fahr, (5.0 / 9.0) * (fahr - 32));
}
```

Medidas en Ingeniería del Sw.



Medidas en Ingeniería del Sw.

- † Características deseables de las medidas
 - Deben ser sencillas y estar definidas con precisión
 - Deben ser objetivas
 - Deben ser fáciles de obtener
 - Deben ser válidas
 - Deben ser robustas
 - Deben definir sus valores y sus límites
 - Los conjuntos de medidas deben ser consistentes

Medidas en Ingeniería del Sw.

- ✦ Necesidad de las medidas del *software*
 - Solucionar los problemas del *software*
 - ◆ Estimaciones de tiempo y costes más precisas
 - ◆ Mayor productividad
 - ◆ Productos de mejor calidad
 - Gestión del *software* no es efectiva
 - ◆ Desarrollo de *software* es muy complejo
 - ◆ Se tiene pocas medidas buenas del proceso o del producto
 - ◆ La mejora del proceso necesita identificar, medir y controlar sus parámetros fundamentales

Medidas en Ingeniería del Sw.

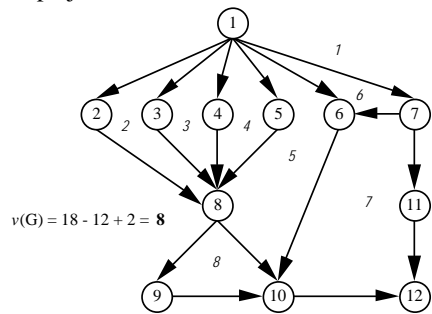
- ✦ Gran cantidad de medidas del software
 - 1976: Complejidad ciclométrica (McCabe)
 - 1977: Ciencia del software (Halstead)
 - 1980: Complejidad del flujo de control, del flujo de datos y del programa (Oviedo)
 - 1981: Flujo de información (Kafura y Henry)
 - 1985: Estabilidad en el mantenimiento (Yau y Collofello)
 - 1988: "Ley de Demeter" (Lieberherr)
 - 1993: Medidas del Diseño (Chen y Lu)
 - 1994: Conjunto de medidas (Chidamber y Kemerer)
 - 1999: Complejidad del código y del diseño (Etz Korn et al.)

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Complejidad de McCabe [McCabe, 1976]
 - número de caminos de ejecución posibles en un programa
 - $v(G) = e - n + 2p$
 - ◆ e : bifurcaciones
 - ◆ n : nodos con código secuencial
 - ◆ p : número de grafos inconexos
 - $v(G) = \pi + 1$
 - ◆ π : número de predicados

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Complejidad de McCabe [McCabe, 1976]

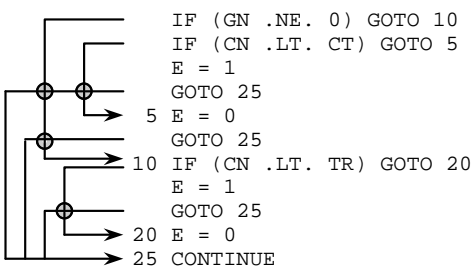


Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Ciencia del Software [Halstead, 1977]
 - Conjunto de medidas y estimaciones sobre el código fuente
 - Medidas
 - ◆ Contenido de Inteligencia
 - ◆ Nivel del Programa
 - ◆ Longitud del Programa
 - ◆ Volumen del Programa
 - ◆ Dificultad
 - ◆ Esfuerzo

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Complejidad del control de flujo [Woodward, 1979]



Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Flujo de información [Henry y Kafura, 1981]
 - Complejidad interna (C_{ip}) de un módulo p :
 - ◆ Número de líneas de código (LOC) de dicho módulo, longitud de Halstead, complejidad ciclomática...
 - *Fan-in* de un módulo p :
 - ◆ Número de flujos de datos entrando al módulo p
 - *Fan-out* de un módulo p :
 - ◆ Número de flujos de datos saliendo del módulo p
 - Complejidad del flujo de información de un módulo p (C_p):

$$C_p = C_{ip} \cdot (\text{fan-in} \cdot \text{fan-out})^2$$

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Ley de Demeter [Lieberherr, 1988]
 - Para todas las clases C y todos los métodos M de C , todos los objetos a los que M envía mensajes deben ser:
 - ◆ Los objetos que son argumentos de M (incluyendo el propio objeto de la clase C)
 - ◆ Los objetos que son atributos de C
 - Medida:
 - ◆ Número de veces que se viola la ley

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - WMC (*Weighted Methods per Class*)
 - ◆ Suma ponderada de la complejidad de cada uno de los métodos de una clase
 - Sea la clase C con sus métodos M_1, M_2, \dots, M_n
 - Sea c_1, c_2, \dots, c_n la complejidad de cada método
 - Entonces, para dicha clase C :

$$WMC = \sum_{i=1}^n c_i$$

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - DIT (*Depth of Inheritance Tree*)
 - ◆ Profundidad de una clase C en el árbol de herencia
 - ◆ DIT es una medida de cuántas superclases pueden afectar potencialmente a cada clase
 - ◆ Medida: contar los niveles del grafo de herencia

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - NOC (*Number of Children*)
 - ◆ Número de clases subordinadas inmediatamente a una clase
 - ◆ NOC es una medida de cuántas clases heredarán los métodos de la clase padre
 - ◆ Está relacionada con la noción del alcance de los miembros
 - ◆ Medida: contar los hijos directos de una clase con una relación de herencia

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - CBO (*Coupling Between Objects classes*)
 - ◆ Cantidad de clases a la que está acoplada (número de objetos que actúan sobre otro)
 - ◆ Relacionada con la noción de que un objeto está acoplado con otro si uno de ellos actúa sobre el otro (si los métodos de uno usan métodos o atributos del otro)

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - RFC (*Response For a Class*)
 - ◆ Conjunto de métodos de una clase que potencialmente pueden ejecutarse como respuesta a la llegada de un mensaje recibido en un objeto de esa clase
 - ◆ Medida:
 - M: conjunto de todos los métodos de la clase C
 - R_i: conjunto de los métodos llamados por el método i

$$RFC = M \bigcup_{v_i} R_i$$

Medidas en Ingeniería del Sw. Ejemplos de Medidas del Software

- ✦ Conjunto de medidas para OO [Chidamber y Kemerer, 1991]
 - LCOM (*Lack of Cohesion Of Methods*)
 - ◆ Proporciona una medida de la relativa disparidad natural de los métodos de una clase

Medidas en Ingeniería del Sw. Clasificaciones de Medidas del Software

- ✦ Clasificación basada en los elementos que se miden [Henry]:
 - Medidas **Léxicas**
 - ◆ Se basan en contar *tokens*
 - Medidas **Semánticas**
 - ◆ Se basan en la aplicación de los conceptos de la teoría de la información a la formulación de medidas
 - Medidas de **Conectividad**
 - ◆ Miden el grado de interconectividad entre los componentes del sistema observando el flujo de información entre ellos

Medidas en Ingeniería del Sw. Clasificaciones de Medidas del Software

- ✦ Clasificación de las medidas del producto [Kafura]:
 - Medidas de la **Estructura**
 - ◆ Se basa en el análisis de la estructura del diseño
 - Medidas del **Código**
 - ◆ Se basan en los detalles del código fuente
 - Medidas **Híbridas**
 - ◆ Se basan tanto en detalles de implementación como en la estructura del diseño

Medidas en Ingeniería del Sw.

- ✦ Principios de las medidas del software:
 - Medir es un mecanismo ideal para caracterizar, evaluar, predecir y proporcionar motivación para los diversos aspectos de los procesos de construcción del software
 - Las medidas deben aplicarse tanto sobre el proceso software como en el producto
 - Debe estar claramente indicado el propósito de cada medida
 - Se necesitan tanto medidas objetivas como subjetivas
 - La mayor parte de los aspectos del producto y del proceso software son demasiados complicados para ser identificados por una única medida

Medidas en Ingeniería del Sw.

- ✦ Principios de las medidas del software:
 - Los entornos de desarrollo y mantenimiento deben estar preparados para las medidas
 - La tarea de medir no se debe limitar a utilizar modelos y medidas tal como han sido definidas en otros entornos
 - El proceso de medida debe ser de arriba hacia abajo, en vez de ser de abajo hacia arriba, para poder definir un conjunto de objetivos operativos, especificar las medidas apropiadas, permitir interpretaciones y análisis contextuales válidos y proporcionar retroalimentación para el aprendizaje y el seguimiento

Medidas en Ingeniería del Sw.

- ✦ Principios de las medidas del software:
 - Las medidas deben asociarse con interpretaciones, pero estas interpretaciones deben corresponder con un determinado contexto
 - Se necesitan múltiples mecanismos para la recopilación y validación de datos
 - Para evaluar y comparar proyectos y para llevar a cabo modelos se necesita una base histórica de experiencias

Medidas en Ingeniería del Sw.

- ✦ Aplicación del *principio de incertidumbre de Heisenberg* al cálculo de las medidas

"Es imposible determinar simultáneamente la posición y velocidad exactas de un electrón"

- ✦ Efecto *Hawthorne*

"La medida de cualquier parámetro de un proyecto y su asociación con una evidencia significativa influirá en la utilidad de dicha medida"

Calidad del Software



Calidad del Software Introducción

- ✦ Calidad de un producto

- Punto de vista subyacente:
 - ◆ Excelencia innata
- Punto de vista del producto:
 - ◆ Contenidos del producto
- Punto de vista del usuario:
 - ◆ Adecuación para el uso
- Punto de vista industrial:
 - ◆ Conforme a requisitos
- Punto de vista del valor del producto:
 - ◆ Diseño al coste

Software

Calidad del Software Introducción

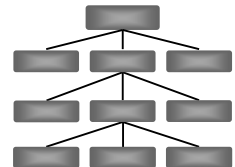
- ✦ Calidad
 - Diccionario de la Real Academia Española
"Propiedad o conjunto de propiedades inherentes a una cosa, que permiten **apreciarla como igual, mejor o peor** que las restantes de su especie"
- ✦ Calidad del software
 - IEEE
"Grado con el cual el cliente o usuario percibe que el software **satisface** sus expectativas"



Calidad del Software Modelos de Calidad

- ✦ Modelos de calidad del software

- Intentan cuantificar la calidad del software
- Se descompone la calidad en niveles estructurados
- Los distintos modelos se diferencian por:
 - ◆ la relación entre los niveles
 - ◆ la cantidad de niveles
 - ◆ los conceptos de cada nivel



Calidad del Software

Modelos de Calidad

1976	Modelo de Boehm
1977	Modelo de McCall
1978	
1979	
1980	
1981	
1982	
1983	
1984	
1985	Modelo de Arthur
1986	
1987	
1988	Modelo de Gilb / Modelo de Deutsch y Willis
1989	
1990	Modelo de Schulmeyer
1991	
1992	Modelo de Gillies / Modelo REBOOT
1993	
1994	
1995	Modelo de Dromey
1996	
1997	
1998	
1999	
2000	
2001	ISO 9126

Calidad del Software

Modelo de Calidad ISO/IEC 9126

ISO/IEC 9126:

- Software Engineering – Product Quality
- 4 partes:
 - ◆ ISO/IEC IS 9126-1:2001
 - Quality Model (15-6-2001)
 - ◆ ISO/IEC TR 9126-2:2003
 - External Metrics (1-7-2003)
 - ◆ ISO/IEC TR 9126-3:2003
 - Internal Metrics (1-7-2003)
 - ◆ ISO/IEC TR 9126-4:2004
 - Quality in use Metrics (1-4-2004)

Calidad del Software

Modelo de Calidad ISO/IEC 9126-1

Modelo en dos partes para la calidad del software

- Calidad interna y externa
 - ◆ Subdivisión en 6 características
 - ◆ Características
 - Subdivisión en subcaracterísticas
- Calidad en el uso
 - ◆ Subdivisión en 4 características

Calidad del Software

Modelo de Calidad ISO/IEC 9126-1

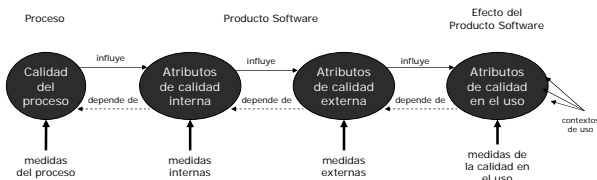
Características

- Se muestran externamente al usar el software y son el resultado de atributos internos
- Son aplicables a cualquier tipo de software
- Conceptos “características” y “subcaracterísticas”
 - ◆ Proporcionan una terminología consistente para la calidad del producto software
 - ◆ Proporcionan un marco para especificar requisitos de calidad del software
 - ◆ Permiten definir compromisos entre capacidades del producto software

Calidad del Software

Modelo de Calidad ISO/IEC 9126-1

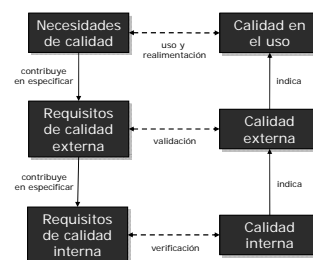
La calidad en el ciclo de vida

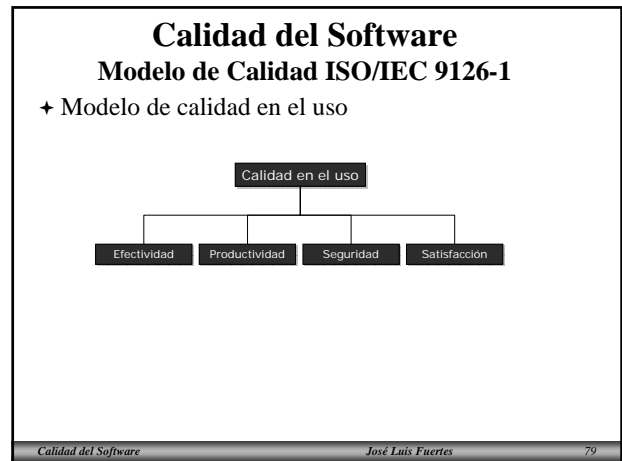
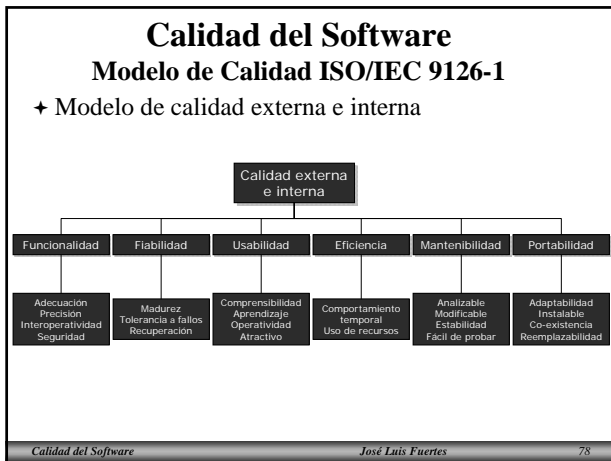


Calidad del Software

Modelo de Calidad ISO/IEC 9126-1

La calidad en el ciclo de vida





Mejora del Proceso Software

Antecedentes

✦ El proceso software

- Secuencia de pasos necesarios para desarrollar software
- Establece el marco de trabajo técnico y de gestión para aplicar los métodos, herramientas y el personal a la tarea de la construcción de programas

Calidad del Software José Luis Fuertes 80

Mejora del Proceso Software

Antecedentes

✦ Mejora del proceso software

- Monitorizar, medir y revisar el rendimiento del proceso estándar al ser aplicado sobre proyectos individuales
- Todo el personal implicado en el proceso software debe participar en las actividades de mejora
- Es primordial fijar y perseguir objetivos cuantitativos y medibles para mejorar el proceso, dirigiendo estos objetivos a perfeccionar la calidad del producto
- Debe establecerse un programa de mejora del proceso que impulse a los desarrolladores a mejorar su propio proceso de trabajo y a participar en las mejoras de sus compañeros

Calidad del Software José Luis Fuertes 81

Mejora del Proceso Software

Antecedentes

✦ *Carnegie Mellon University (CMU) Software Engineering Institute (SEI)*

- *Capability Maturity Model (1991)*
 - ◆ Establecer programas de mejora del proceso software
 - ◆ Indica los objetivos de las medidas a usar en cada nivel
- *Personal Software Process (1993)*
 - ◆ Conseguir mejores Ingenieros del Software
 - ◆ Incluye algunas medidas básicas y del proceso

Calidad del Software José Luis Fuertes 82

Mejora del Proceso Software

CMM

✦ Una forma de discriminar entre niveles de madurez del proceso es la habilidad de los desarrolladores y gestores para ver y entender qué es lo que ocurre durante todo el proceso de desarrollo

- En el nivel inferior de madurez, no se comprende ni el proceso, pero conforme crece la madurez, se comprende y se puede definir mejor
- Tanto las medidas como la habilidad para ver y entender están íntimamente relacionadas, puesto que un desarrollador puede medir sólo lo que resulta visible en un proceso y las medidas ayudan a incrementar esa visibilidad

Calidad del Software José Luis Fuertes 83

Mejora del Proceso Software CMM

- El CMM puede servir como una guía para determinar qué medir primero y cómo planear un plan de medidas comprensivo y adecuado
- Tipos de medidas por nivel del CMM:

Nivel de Madurez	Características	Objetivo de las Medidas
1. Inicial:	Ad hoc, caótico	Establecer las bases para planear y estimar
2. Repetible:	Los procesos dependen de los individuos	Seguimiento y control del proyecto
3. Definido:	Los procesos se definen y se institucionalizan	Definición y cuantificación de los procesos y productos intermedios
4. Gestionable:	Se miden los procesos	Definición, cuantificación y control de los subprocesos y los elementos
5. Optimizado:	Los procesos se retroalimentan con las mejoras	Optimización dinámica y mejora durante el proyecto

Mejora del Proceso Software PSP

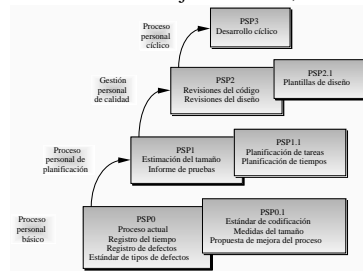
- El PSP es un proceso de perfeccionamiento diseñado para ayudar a controlar, gestionar y mejorar la forma de trabajar
- Marco de trabajo estructurado con formularios, guías y procedimientos para desarrollar software
- PSP proporciona los datos históricos necesarios para mejorar el proceso
- Principal objetivo: conseguir mejores ingenieros del software

Mejora del Proceso Software PSP

- Permite comprender por qué se han cometido errores y cuál es la mejor forma de encontrarlos
- Se puede determinar la calidad de las revisiones, los tipos de errores no detectados y los métodos más efectivos para cada ingeniero

Mejora del Proceso Software PSP

- El PSP está dividido en cuatro fases:
 - PSP0, PSP1, PSP2 y PSP3
 - Fases intermedias de mejora: PSP0.1, PSP1.1 y PSP2.1



Conclusiones

- La calidad del software puede verse como un problema económico
 - Aunque es fundamental obtener un software de calidad, cada medida, cada test, cada revisión... consume tiempo y dinero
- Si se desea un software de alta calidad, hay que asegurarse de que cada una de sus partes tenga alta calidad



Conclusiones

- ✦ Razones para la lentitud en la adopción del uso de medidas para evaluar la calidad del software:
 - No existen medidas de la calidad del software universales
 - ◆ Existen algunas medidas útiles para ciertos entornos
 - ◆ Existirán medidas de calidad ampliamente aceptadas, cuando madure la investigación en medidas de calidad del software
 - Incluso sabiendo qué medidas usar, no es fácil obtener los datos
 - Incluso con los datos, no es obvio cómo interpretar y usar los números
 - La gente se resiste a que se mida la calidad de su trabajo

Conclusiones

- ✦ ISO/IEC 9126-1: Modelo de Calidad del Software
 - Los estándares, por sí solos, no son suficientes
 - Se necesita una disciplina para aplicarlos
 - Si no se comprende el proceso del negocio, no resultará útil aplicar ningún método de evaluación de la calidad ni de mejora del proceso
- ✦ La medida de los atributos del software puede usarse para predecir o medir indirectamente la calidad del software