




Métricas de Calidad de Software

Universidad de Morón
Facultad de Informática, Ciencias de la
Comunicación y Técnicas Especiales

Herramientas y Procesos de Software

1



¿Qué pasa si no medimos?

- No sabemos si estamos mejorando
- No podemos establecer metas

2

¿Qué se puede medir ?

- El proceso del software (para mejorarlo).
- El proyecto del software (para ayudar a estimar, control de calidad, evaluación de productividad, control de proyectos).
- Calidad del producto (para ayudar en la toma de decisiones tácticas a medida que el proyecto evoluciona).

3

Medidas , métricas e indicadores

- **Medida:** Indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto.
- **Medición:** Es el acto de determinar una medida.
- **Métrica:** Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado.
- **Indicador:** Es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del SW, del proyecto del SW o del producto en sí.

4

Métricas

Que miden	Productividad Calidad
Beneficios	Futuras Estimaciones
Medidas	Directas Indirectas

5

Mediciones del software

- Las mediciones del mundo físico se pueden clasificar de dos maneras:
 - **Medidas directas:** Ej. Longitud de un tornillo
 - **Medidas indirectas:** Ej. Calidad de los tornillos producidos, medidos contando los artículos defectuosos
- Las métricas del SW, se categorizan de forma similar
 - **Directas:** líneas de código producidas (LDC), velocidad de ejecución, tamaño de memoria
 - **Indirectas:** funcionalidad, calidad, complejidad.

6

Costo y métricas de calidad

Defecto/fallo Vs. Error

- **Defecto y fallo:** son sinónimos dentro del contexto del proceso del software. Implican un problema de calidad que es descubierto una vez entregado el software al cliente.
- **Error:** Problema de calidad detectado por los ingenieros u otros dentro del proceso del software.

7

Indicadores Proyecto/Proceso

- **Indicadores de proceso:** Permiten a una organización de ingeniería del software tener una visión profunda de la eficacia de un proceso ya existente. (las métricas del proceso se recopilan de todos los proyectos y durante un largo período de tiempo)
- **Indicadores de proyecto:** Permiten al gestor de proyectos del SW
 1. evaluar el estado de un proyecto en curso;
 2. seguir la pista de los riesgos potenciales;
 3. detectar la áreas de problemas antes de que se conviertan en críticas;
 4. ajustar el flujo y las tareas del trabajo;
 5. evaluar la habilidad del equipo.

8

Métricas simples

-orientadas al tamaño-

- Errores por KLDC (miles de líneas de código, KiloLDC).
- Defectos por KLDC.
- Páginas de documentos por KLDC.
- Errores/persona-mes.
- LDC/persona mes.

9

Métricas

- Por término general, para la evaluación de la calidad, es más habitual centrarse en medidas del producto que en medidas del proceso.
- Una métrica es una asignación de un valor a un atributo (tiempo, complejidad, etc.) de una entidad software, ya sea un producto (código) o un proceso (pruebas).

10

Principios de medición

- Los objetivos de la medición deben establecerse antes de empezar la recogida de los datos.
- Todas la métricas deben definirse sin ambigüedades.
- La métricas deben obtenerse basándose en una teoría válida para el dominio de la aplicación.
- Siempre que sea posible, automatizar la recogida de los datos y el análisis.
- Aplicar técnicas estadísticas válidas para establecer relaciones entre los atributos internos y características externas de la calidad del producto.
- Establecer directrices de interpretación y recomendación.

11

Características de las métricas

- Simples y fáciles de calcular
- Empírica e intuitivamente persuasivas
- Consistentes y objetivas
- Independientes del lenguaje de programación
- Eficaz mecanismo para realimentar la calidad

12

Etapas del proceso de medición

- **Formulación:** la obtención de medidas y métricas apropiadas para la representación del SW que se desea desarrollar.
- **Colección:** mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- **Análisis:** cálculo de las métricas.
- **Interpretación:** evaluación de los resultados.
- **Realimentación:** recomendaciones obtenidas a través de la interpretación de las métricas obtenidas por el equipo de desarrollo.

13

Métricas

- Para la evaluación de las características del SW, utilizaremos métricas. Clasificación:
 - Clasificación 1:
 - Métricas de producto.
 - Métricas de proceso.
 - Clasificación 2:
 - Métricas basadas en atributos internos del producto:
 - Medidas de estructuración de un programa.
 - Métricas de complejidad.
 - Métricas de cobertura de pruebas.
 - Métricas de calidad del diseño.
 - Métricas basadas en atributos externos del producto:
 - Métricas de portabilidad.
 - Métricas de defectos.
 - Métricas de usabilidad.
 - Métricas de mantenibilidad.
 - Métricas de fiabilidad.

14

Métricas

- Métricas basadas en código fuente:
 - Nº de líneas de código.
 - Nº de líneas de comentario.
 - Nº de instrucciones.
 - Densidad de documentación.

- Métricas basadas en estructura de diseño:
 - Relacionadas con el control intramodular.
 - Relacionadas con el acoplamiento entre clases.

- Métricas para sistemas orientados a objetos:
 - Acoplamiento.
 - Herencia.
 - Cohesión.

15

Métricas del Modelo de Análisis

- **Métricas basadas en la función**
 - Utilizada para medir el tamaño del sistema a construir.
 - Se calcula: el nro. de entradas del U, de salidas del U, de peticiones al U, de archivos, y de interfaces externas, asociados a un nro. de complejidad (simple, medio, complejo), más un factor de complejidad.
 - Se aplica en forma análoga a las LDC.
- **Métrica Bang**
 - Utilizada para medir el tamaño del sistema a construir.
 - Se calcula: el nro. de primitivas funcionales (burbujas), ED, O, R, estados, transiciones, primitivas modificadas (funciones externas adaptadas), ED de entrada, ED de salida, ED grabados.
- **Métrica de la calidad de la Especificación**
 - Valoración del modelo con la especificación de requisitos (completitud, consistencia, ambigüedad, comprensión, etc.).
 - Se plantea una fórmula para cada uno de los atributos de la especificación, incluyendo req. Funcionales y no funcionales, interpretación, etc.).

16

Métricas del Modelo de Diseño

- **Métricas del Diseño Arquitectónico:** se centran en la arquitectura del programa y la eficiencia de los módulos. Son de caja negra.
 - Medidas de la Complejidad Estructural, de Datos, del Sistema.
- **Métricas a nivel de Componentes:** se centran en las características internas del módulo. Son de caja blanca. Necesitan el diseño procedimental.
 - Métricas de Cohesión
 - Métricas de Acoplamiento
 - Métricas de complejidad del flujo de control del programa
- **Métricas del Diseño de Interfaz**

17

Métricas del Código Fuente

- Se tendrá en cuenta:
 - El nro. de operadores diferentes que aparecen en el progr.
 - El nro. de operandos diferentes que aparecen en el progr.
 - El nro. total de veces que aparece el operador,
 - El nro. total de veces que aparece el operando.
- De allí se calcula:
 - Longitud del programa
 - Volumen de información (en bits) necesarios para escribir un programa.
 - Nivel del programa (complejidad del SW)
 - Tiempo de desarrollo
 - Costo de desarrollo
 - Nro. esperado de fallos en el SW

18

Métricas para Pruebas

- La mayoría de las métricas se concentran en el proceso y no en producto.
- Debe apoyarse en las métricas del análisis y del diseño.
- Las métricas *basadas en la función*, pueden utilizarse para predecir el esfuerzo global de las pruebas.
- La métrica *Bang* puede proporcionar una indicación del nro. De los casos de prueba necesarios para examinar una medida primitiva (burbuja del DFD).
- A medida que avanzan las pruebas, hay métricas que indican la completitud de las mismas:
 - Amplitud de las pruebas (cuantos requisitos se han probado).
 - Profundidad de las pruebas (% de los caminos básicos probados).
 - Perfiles de fallos (para dar prioridad y categorizar los errores encontrados).

19

Métricas para el Mantenimiento

- Pueden utilizarse todas las métricas anteriores.
- El estándar IEEE982.1-1998, sugiere un Índice de Madurez del SW, que proporcionan una indicación de la estabilidad del SW (basada en los cambios que ocurren).
- Se tendrá en cuenta:
 - Nro. de módulos en la versión actual
 - Nro. de módulos en la versión actual que han cambiado
 - Nro. de módulos en la versión actual que se han añadido
 - Nro. de módulos en la versión anterior que se han borrado.

20

Métricas Técnicas para Sistemas OO

○ Características de las métricas OO:

- **Localización:** indica la manera en que la información se concentra en un programa.
- **Encapsulamiento:** influye en las métricas, cambiando el enfoque de la medición de un módulo simple a un paquete de datos y módulos de procesos (operaciones).
- **Ocultamiento de Información**
- **Herencia:** anidamiento de clases.
- **Abstracción:** las métricas representan abstracciones en términos de medición de una clase.

21

Métricas del Modelo de Diseño OO

- **Tamaño:** Población (cantidad de entidades), Volumen (cantidad de entidades en momento de ejecución), Longitud (medida de interconexión de elementos) y funcionalidad.
- **Complejidad:** está determinada por la forma de interrelacionar clases.
- **Acoplamiento:** conexiones entre elementos.
- **Suficiencia:** grado en que un componente refleja todas las propiedades y operaciones pertinentes.
- **Integridad:** grado en que un componente tiene todo lo que necesita.
- **Cohesión:** grado en que el conjunto de propiedades que contiene son parte del problema.
- **Originalidad:** grado en que una clase contiene operaciones atómicas o primitivas.
- **Similitud:** entre clases, estructuras o comportamiento.
- **Volatilidad:** probabilidad de cambio.

22

Métricas orientadas a Clases

- **Serie de métricas CK**
 - Métodos ponderados por clase
 - Árbol de profundidad de herencia
 - Número de descendientes
 - Acoplamiento ente clases
 - Respuesta para una clase
 - Carencia de cohesión en los métodos
- **Métricas propuestas por Lorenz y Kidd**
 - Tamaño de la clase (nro. de operaciones, atributos)
 - Nro. de operaciones redefinidas para una subclase
 - Nro. de operaciones añadidas por una subclase
 - Índice de especialización
- **Colección de métricas MDOO**
 - Factor de herencia de métodos
 - Factor de acoplamiento
 - Factor de polimorfismo

23

Métricas orientadas a Operaciones

- Tamaño medio de operación (nro. de mensajes enviados por la operación).
- Complejidad de la operación (se puede aplicar cualquier métrica convencional).
- Nro. de parámetros por operación

24

Métricas para Pruebas OO

- **Encapsulamiento**
 - Carencia de cohesión en métodos
 - Porcentaje público y protegido
 - Acceso público a datos (de una clase a otra)
- **Herencia**
 - Nro. de clases raíz
 - Nro. de padre directos
 - Nro. de descendientes y árbol de profundidad de herencia
- **Complejidad y Polimorfismo de una clase**
 - Métodos ponderados por clase
 - Acoplamiento ente clases
 - Respuesta para una clase

25

Para lograr buenas métricas

- Utilice el sentido común y una sensibilidad organizativa al interpretar datos de métricas.
- No utilice métricas para evaluar a particulares.
- Trabaje con profesionales y equipos para establecer objetivos claros y métricas que se vayan a utilizar para alcanzarlos.
- No utilice nunca métricas que amenacen a particulares y/o equipos.
- La medición es esencial si se desea construir un SW con calidad.

26

Métricas Técnicas del SW

- La medición asigna números a entidades del mundo real, a través de un modelo de medición.
- Facilitan una base para que el análisis, diseño, codificación y prueba puedan ser conducidos más objetivamente y valorados más cuantitativamente.
- Ayudan a la evaluación de los modelos de análisis y diseño.
- Proporcionan una indicación de la complejidad de los diseños procedimentales y del código fuente.
- Ayudan al diseño de pruebas más efectivas

27