MÉTRICAS DEL SOFTWARE:

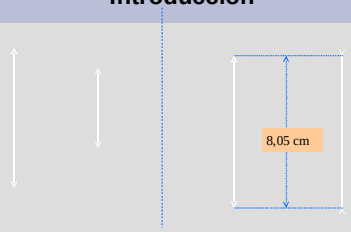
Conceptos básicos, definición y formalización

Calidad de Sistemas de Software
Octubre de 2006

Resumen y traducción inicial
Dra. Coral Calero Muñoz
Universidad de Castilla-La Mancha

CONTENIDOS

Introducción



¿Cómo saber cual es mayor?

INTRODUCCIÓN

- Las métricas son un buen medio para entender, monitorizar, controlar, predecir y probar el desarrollo software y los proyectos de mantenimiento (Briand et al., 1996)
- En general, la medición persigue tres objetivos fundamentales: ayudarnos a entender qué ocurre durante el desarrollo y el mantenimiento, permitirnos controlar qué es lo que ocurre en nuestros proyectos y poder mejorar nuestros procesos y nuestros productos (Fenton y Pfleeger, 1997).

INTRODUCCIÓN

- Las métricas del software es un término que se asigna a un amplio rango de actividades diversas, por ejemplo:
  - medidas y modelos de estimación de coste y esfuerzo
  - modelos y medidas de productividad
  - aseguramiento y control de calidad.
  - recogida de datos
  - medidas y modelos de calidad
  - modelos de fiabilidad
  - modelos y evaluación de ejecución
  - complejidad computacional o algorítmica.

INTRODUCCIÓN

En software hay tres clases de entidades cuyos atributos podemos querer medir:

- **Procesos**: Son actividades software que normalmente conllevan el factor tiempo. Atributos internos interesantes: el tiempo (duración del proceso), el esfuerzo (asociado al proceso) y el número de incidentes de un tipo específico que se dan durante el proceso (por ejemplo el número de errores de requisitos encontrados durante la construcción de la especificación).
- **Productos**: son entregables, artefactos o documentos generados en el ciclo de vida del software. Ejemplos de atributos externos: la fiabilidad del código, la entendibilidad de un documento de especificación, la mantenibilidad del código fuente e.t.c., ejemplos de atributos internos: la longitud, funcionalidad, modularidad o correción sintáctica de los documentos de especidicación.
- **Recursos**: son todos aquellos elementos que hacen de entrada a la producción software. Por ejemplo el personal, los materiales, las herramientas y los métodos. Un atributo interesante es el coste. En el caso del personal, además del coste, se suele medir la productividad

INTRODUCCIÓN

- Últimamente ha aparecido un gran número de métricas para capturar atributos del software de una forma cuantitativa.
- Sin embargo, muy pocas métricas han sobrevivido a la fase de definición y se usan en la industria. Esto se debe a múltiples problemas, entre ellos:
  - Las métricas no se definen siempre en un contexto en el que el objetivo de interés industrial que se pretende alcanzar

INTRODUCCIÓN

  - Las definiciones de métricas no siempre tienen en cuenta el *entorno o contexto* en el que serán aplicadas
  - No siempre es posible realizar una *validación teórica* adecuada de la métrica porque el atributo que queremos medir no siempre está bien definido
  - Un gran número de métricas nunca se ha *validado empíricamente*
- Esta situación ha conducido frecuentemente a cierto grado de ambigüedad en las definiciones, propiedades y asunciones de las métricas, haciendo que el uso de las mismas sea difícil, la interpretación peligrosa y los resultados contradictorios.
- Para evitarlo es necesario contar con un método de

CONTENIDOS

FORMALIZACIÓN DE MÉTRICAS

- La medición de software es una disciplina relativamente joven, y no existe consenso general sobre la definición exacta de los conceptos y terminología que maneja.
- Creación de una ontología entre:
  - Universidad de Castilla-La Mancha
  - Universidad de Málaga
  - Universidad Nacional de La Pampa
  - Universidad Politécnica de Cataluña
  - Universidad Politécnica de Valencia

FORMALIZACIÓN DE MÉTRICAS

- ¿Qué es una ontología? (Gruber, 1995)
  - Una ontología es una especificación de una conceptualización
- En otras palabras:
  - Mediante la definición de ontologías se pretende reunir y formalizar el conocimiento sobre un determinado dominio de problema.
  - Se mejora el entendimiento y la comunicación mediante el establecimiento de vocabularios comunes, lo que facilita la reutilización y la interoperabilidad de los

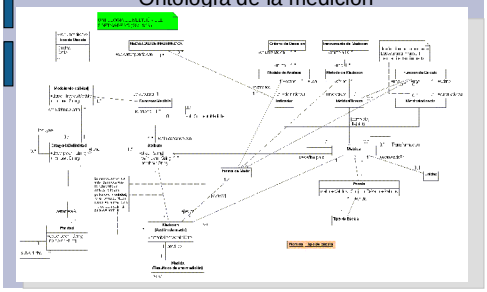
FORMALIZACIÓN DE MÉTRICAS

La propuesta aquí presentada está basada en cuatro conceptos fundamentales:

- la *forma de medir*;
- la acción de medir (denominada *medición*);
- el resultado de la medición (denominada *medida*); y
- el concepto de *métrica* (definido como “una *forma de medir* + una *escala*”)

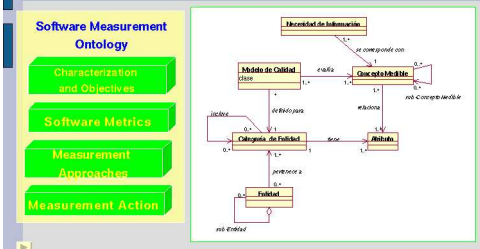
FORMALIZACIÓN DE MÉTRICAS

Ontología de la medición



FORMALIZACIÓN DE MÉTRICAS

FORMALIZACIÓN DE MÉTRICAS



FORMALIZACIÓN DE MÉTRICAS

Necesidad de información

Definición

Información necesaria para gestionar un proyecto (sus objetivos, hitos, riesgos y problemas).

Relaciones

Una *necesidad de información* se asocia a un *concepto medible*.

Una *necesidad de información* se satisface con uno o más *indicadores*.

Ejemplos

Conocer el nivel de productividad de los programadores del proyecto en comparación con lo habitual en otros proyectos en la organización.

Determinar si los recursos del proyecto son adecuados para satisfacer sus objetivos.

Evaluar el rendimiento de la actividad de codificación.

Evaluar si un producto software satisface las expectativas del

FORMALIZACIÓN DE MÉTRICAS

Concepto medible

Definición

Relación abstracta entre *atributos* y *necesidades de información*

Relaciones

Un concepto medible se asocia a una o varias necesidades de información.

Un concepto medible puede incluir otros conceptos medibles.

Un concepto medible relaciona uno o más atributos.

Un concepto medible está incluido en uno o más *modelos de calidad*.

Ejemplos

Relación de productividad de un equipo de desarrollo frente a un grado de productividad objetivo.

Adecuación de la tecnología.

FORMALIZACIÓN DE MÉTRICAS

Modelo (de calidad)

Definición

Un marco conceptual que especifica una serie de *conceptos medibles* y sus relaciones, para una determinada *categoría de entidad*.

Relaciones

Un modelo (de calidad) está definido para una determinada categoría de entidad.

Un modelo (de calidad) evalúa uno o varios *conceptos medibles*.

Ejemplos

Modelo de calidad para productos software de ISO 9126.

Factores de calidad de McCall [McCall 1977].

FORMALIZACIÓN DE MÉTRICAS

Categoría de entidad

Definición

Una colección de *entidades* caracterizadas por satisfacer un cierto predicado común.

Relaciones

Una *categoría de entidad* puede "incluir a" una o varias *categorías de entidad*, y puede "estar incluida en" una o varias *categorías de entidad*.

Una *categoría de entidad* tiene uno o varios atributos.

Una *categoría de entidad* puede tener definidos varios *modelos de calidad*.

Ejemplos

"Programas", "Programas en C", "Componentes software", "Componentes software para comunicaciones", etc.

Procesos, productos, servicios, proyectos, o recursos son ejemplos de categorías de entidad

FORMALIZACIÓN DE MÉTRICAS

Entidad

Definición

Un objeto que va a ser caracterizado mediante una *medición* de sus *atributos* [ISO-15939]

Relaciones

Una *entidad* puede pertenecer a una o más *categorías de entidad*.

Una *medición* se realiza sobre los *atributos* de una *entidad*

Ejemplos

El programa "holaMundo.c".

Notas

Una entidad puede ser un proceso, un producto, un servicio, un proyecto, o un recurso concreto.

Una entidad puede ser física –tangible– (p.e. un computador) o abstracta (p.e. un programa en C).

FORMALIZACIÓN DE MÉTRICAS

Atributo

Definición

Una propiedad mensurable, física o abstracta, que comparten todas las *entidades* de una *categoría de entidad*.

Relaciones

Un atributo solo puede pertenecer a una categoría de entidad.

Una medición se realiza sobre los atributos de una entidad

Un atributo tiene definida cero, una o varias métricas.

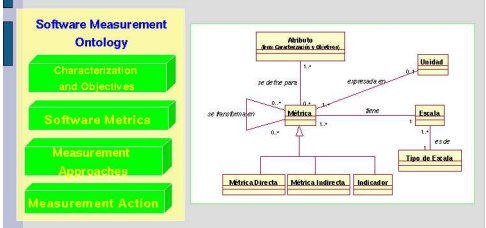
Un atributo está relacionado con uno o más conceptos medibles.

Ejemplos

El *atributo* "tamaño de código fuente", como *atributo* de la *categoría de entidad* "programas en C" va a ser diferente del *atributo* "tamaño de código fuente" de la *categoría de entidad* "programa en Ada".

El "tamaño" de un "módulo en C" no es el mismo atributo (aunque tenga el mismo nombre) que el "tamaño" de un "diagrama de clases UML".

FORMALIZACIÓN DE MÉTRICAS



FORMALIZACIÓN DE MÉTRICAS

Atributo

Definición

Una propiedad mensurable, física o abstracta, que comparten todas las *entidades* de una *categoría de entidad*.

Relaciones

Un atributo solo puede pertenecer a una categoría de entidad.

Una medición se realiza sobre los atributos de una entidad

Un atributo tiene definida cero, una o varias métricas.

Un atributo está relacionado con uno o más conceptos medibles.

Ejemplos

El *atributo* "tamaño de código fuente", como *atributo* de la *categoría de entidad* "programas en C" va a ser diferente del *atributo* "tamaño de código fuente" de la *categoría de entidad* "programa en Ada".

El "tamaño" de un "módulo en C" no es el mismo atributo (aunque tenga el mismo nombre) que el "tamaño" de un "diagrama de clases UML".

FORMALIZACIÓN DE MÉTRICAS

Métrica

Definición

Una *forma de medir* (*método de medición*, *función de cálculo* o *modelo de análisis*) y una *escala*, definidas para realizar *mediciones* de uno o varios *atributos* .

Relaciones

Una métrica está definida para uno o más *atributos*

Dos *métricas* pueden relacionarse mediante una función de transformación. El tipo de dicha función de transformación va a depender del tipo de *escala* de ambas métricas.

Una métrica puede expresarse en una *unidad* (sólo para métricas cuya escala sea de tipo intervalo o ratio)

Ejemplos

La métrica "líneas de código" puede ser definida para realizar mediciones del "tamaño" de un "módulo en C" y para realizar mediciones del "tamaño" de un "programa en Ada".

FORMALIZACIÓN DE MÉTRICAS

Unidad

Definición

Una cantidad particular, definida y adoptada por convención, con la que poder comparar otras cantidades de la misma clase para expresar sus magnitudes respecto a esa cantidad particular [ISO-15939]

Relaciones

Una unidad sirve para expresar una o varias métricas cuyo tipo de escala sea intervalo o relación.

Ejemplos

Kilómetros, metros, millas.

Líneas de código, Páginas, Persona-mes,

Número de módulos, Número de clases,..

Dólares, Pesos, Horas, días, Meses, Años,...

FORMALIZACIÓN DE MÉTRICAS

Escala

Definición

Un conjunto de valores con propiedades definidas [ISO 14598-1]

Relaciones

Toda escala es de un cierto "*Tipo de Escala*".

Ejemplos

Los valores que puede tomar la métrica "lenguaje de Programación usado en un proyecto": Pascal, C, Java (Nominal).

El nivel de madurez CMM: 1, 2, 3, 4, 5 (Ordinal).

El tamaño de un código software expresado en líneas de código: Conjunto de los números naturales (Relación).

La temperatura expresada en grados centígrados o grados Fahrenheit (Intervalo).

FORMALIZACIÓN DE MÉTRICAS

Tipo de Escala

Definición

Indica la naturaleza de la relación entre los valores de la escala [ISO 15939]

Relaciones

A un *Tipo de Escala* pertenecen una o más *Escalas*.

Ejemplos

Nominal, Ordinal, Intervalo, Relación y Absoluta

FORMALIZACIÓN DE MÉTRICAS

OJO

Métrica directa

Definición

Una *métrica* de la cual se pueden realizar *mediciones* sin depender de ninguna otra *métrica* y cuya *forma de medir* es un *método de medición*.

Relaciones

La forma de medir una métrica directa es un método de medición.

Una métrica directa puede ser utilizada en funciones de cálculo.

Ejemplos

LCF (líneas de código fuente escritas).

HPD (horas-programador diarias).

CHP (costo por hora-programador, en unidades monetarias).

FORMALIZACIÓN DE MÉTRICAS

Métrica indirecta

Definición

Una *métrica* cuya *forma de medir* es una *función de cálculo,* es decir, las *mediciones* de dicha *métrica* utilizan las *medidas* obtenidas en *mediciones* de otras *métricas directas* o *indirectas*.

Relaciones

La forma de medir una métrica indirecta es una función de cálculo.

Una métrica indirecta puede usarse en una función de cálculo.

Ejemplos

HPT (horas-programador totales).

LCFH (líneas de código fuente por hora de programador).

CTP (costo total actual del proyecto, en unidades monetarias).

CLCF (costo por línea de código fuente).

FORMALIZACIÓN DE MÉTRICAS

Indicador

Definición

Una *métrica* cuya *forma de medi*r es un *modelo de análisis*, es decir, las *mediciones* de dicha *métrica* utilizan las *medidas* obtenidas en las *mediciones* de otras *métricas (directas, indirectas o indicadores)* junto con *criterios de decisión*.

Relaciones

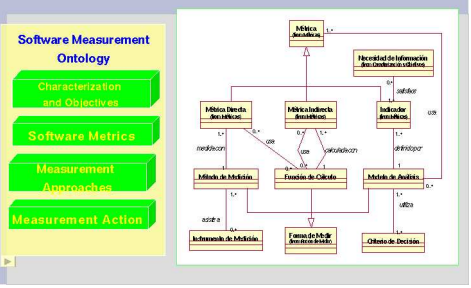
Un indicador satisface necesidades de información.

Un indicador es definido por un modelo de análisis.

Ejemplos

PROD (productividad de los programadores).

FORMALIZACIÓN DE MÉTRICAS



FORMALIZACIÓN DE MÉTRICAS

Métrica

Definición

Una *forma de medir* (*método de medición*, *función de cálculo* o *modelo de análisis*) y una *escala*, definidas para realizar *mediciones* de uno o varios *atributos* .

Relaciones

Una métrica está definida para uno o más *atributos*

Dos *métricas* pueden relacionarse mediante una función de transformación. El tipo de dicha función de transformación va a depender del tipo de *escala* de ambas métricas.

Una métrica puede expresarse en una *unidad* (sólo para métricas cuya escala sea de tipo intervalo o relación)

Ejemplos

La métrica "líneas de código" puede ser definida para realizar mediciones del "tamaño" de un "módulo en C" y para realizar mediciones del "tamaño" de un "programa en Ada".

FORMALIZACIÓN DE MÉTRICAS

Métrica directa

Definición

Una *métrica* de la cual se pueden realizar *mediciones* sin depender de ninguna otra *métrica* y cuya *forma de medir* es un *método de medición*.

Relaciones

La forma de medir una métrica directa es un método de medición.

Una métrica directa puede ser utilizada en funciones de cálculo.

Ejemplos

LCF (líneas de código fuente escritas).

HPD (horas-programador diarias).

CHP (costo por hora-programador, en unidades monetarias).

FORMALIZACIÓN DE MÉTRICAS

Métrica indirecta

Definición

Una *métrica* cuya *forma de medir* es una *función de cálculo,* es decir, las *mediciones* de dicha *métrica* utilizan las *medidas* obtenidas en *mediciones* de otras *métricas directas* o *indirectas*.

Relaciones

La forma de medir una métrica indirecta es una función de cálculo.

Una métrica indirecta puede usarse en una función de cálculo.

Ejemplos

HPT (horas-programador totales).

LCFH (líneas de código fuente por hora de programador).

CTP (costo total actual del proyecto, en unidades monetarias).

CLCF (costo por línea de código fuente).

FORMALIZACIÓN DE MÉTRICAS

Indicador

Definición

Una *métrica* cuya *forma de medi*r es un *modelo de análisis*, es decir, las *mediciones* de dicha *métrica* utilizan las *medidas* obtenidas en las *mediciones* de otras *métricas (directas, indirectas o indicadores)* junto con *criterios de decisión*.

Relaciones

Un indicador satisface necesidades de información.

Un indicador es definido por un modelo de análisis.

Ejemplos

PROD (productividad de los programadores).

FORMALIZACIÓN DE MÉTRICAS

Necesidad de información

Definición

Información necesaria para gestionar un proyecto (sus objetivos, hitos, riesgos y problemas).

Relaciones

Una *necesidad de información* se asocia a un *concepto medible.*

Una *necesidad de información* se satisface con uno o más *indicadores*.

Ejemplos

Conocer el nivel de productividad de los programadores del proyecto en comparación con lo habitual en otros proyectos en la organización (ver ejemplo desarrollado).

Determinar si los recursos del proyecto son adecuados para satisfacer sus objetivos.

Evaluar el rendimiento de la actividad de codificación.

Evaluar si un producto software satisface las expectativas del

FORMALIZACIÓN DE MÉTRICAS

Método de medición

Definición

La *forma de medir* una *métrica directa*. Secuencia lógica de operaciones, descritas de forma genérica, usadas para realizar *mediciones* de un *atributo* respecto de una *escala* específica.

Relaciones

Un método *de medición* define una o más *métricas directas*.

Un método *de medición* puede usar *Instrumentos de Medición*.

Ejemplos

Contar líneas de código.

Anotar cada día las horas dedicadas por los programadores al proyecto.

Valorar el grado de dificultad de un problema.

FORMALIZACIÓN DE MÉTRICAS

Función de cálculo

Definición

La *forma de medir* una *métrica indirecta*. Algoritmo o cálculo realizado para combinar dos o más métricas directas y/o indirectas.

Relaciones

Una función de cálculo usa cero o más métricas directas.

Una función de cálculo usa cero o más métricas indirectas.

Una función de cálculo utiliza al menos una métrica (sea directa o indirecta).

Una función de cálculo define una o más métricas indirectas.

Ejemplos

LCFH = LCF / HPT [métrica indirecta definida en base a 2 métricas directas].

¿métrica indirecta definida en base a sólo 1 métrica directa].

CTP = CHP * HPT [métrica indirecta definida en base a 2 métricas, una directa y otra indirecta].

FORMALIZACIÓN DE MÉTRICAS

Modelo de análisis

Definición

La *forma de medir* un *indicador*. Algoritmo o cálculo realizado para combinar una o más *métricas* (directas, indirectas o indicadores) con *criterios de decisión* asociados.

Relaciones

Un modelo de análisis define uno o más indicadores.

Un modelo de análisis utiliza uno o más criterios de decisión.

Un modelo de análisis usa una o más métricas.

Ejemplos

Modelo de Análisis para obtener la métrica PROD. Utiliza los valores de las métricas LCF, HPT, LCFH y CTP para establecer un valor cualitativo de la productividad de los programadores en este proyecto. El modelo se basa en extraer de una base histórica de proyectos de la organización los valores medios de LCF, HPT, LCFH (LCFHvm) y CTP del subconjunto de proyectos similares (aquellos que tienen LCF entre el 80% y el 120%)

FORMALIZACIÓN DE MÉTRICAS

Instrumento de medición

Definición

Instrumento que asiste o es útil a un *método de medición*.

Relaciones

Un instrumento *de medición* asiste a uno o más *métodos de medición*.

Ejemplos

Un reloj es un *instrumento de medición* que asiste al *método de medición* contar el paso del tiempo.

Una herramienta CASE que sirva para contar líneas de código es un *instrumento de medición* que asiste al *método de medición* contar líneas de código.

FORMALIZACIÓN DE MÉTRICAS

Forma de medir

Definición

Conjunto de operaciones cuyo objeto es determinar el valor de una *medida*. Una *forma de medir* puede ser un *método de medición*, *función de cálculo* o *modelo de análisis*.

Relaciones

Una *forma de medir* es ejecutada en cada *medición*, dependiendo de la *métrica* que calcula.

Ejemplos

Véase los ejemplos de *método de medición*, *función de cálculo* o *modelo de análisis*, ya que la *forma de medir* es una generalización de ellos.

FORMALIZACIÓN DE MÉTRICAS

Criterio de decisión

Definición

Valores umbrales, objetivos, o patrones, usados para determinar la necesidad de una acción o investigación posterior, o para describir el nivel de confianza de un resultado dado.

Relaciones

Un *criterio de decisión* es utilizado en uno o más *modelos de análisis*.

Ejemplos

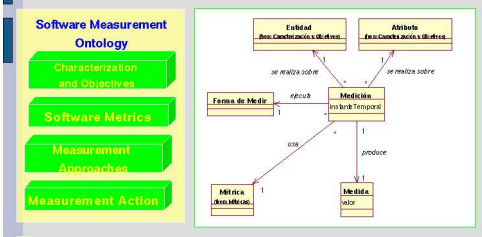
LCFH/LCFHvm < 0’70 => PROD=’muy baja’.

0’70 ≤ LCFH/LCFHvm < 0’90 => PROD=’baja’.

0’90 ≤ LCFH/LCFHvm < 1’10 => PROD=’normal’.

1’10 ≤ LCFH/LCFHvm < 1’30 => PROD=’alta’.

FORMALIZACIÓN DE MÉTRICAS



FORMALIZACIÓN DE MÉTRICAS

Entidad

Definición

Un objeto que va a ser caracterizado mediante una *medición* de sus *atributos* [ISO-15939]

Relaciones

Una *entidad* puede pertenecer a una o más *categorías de entidad*.

Una *medición* se realiza sobre los *atributos* de una *entidad*

Ejemplos

El programa "holaMundo.c".

Notas

Una entidad puede ser un proceso, un producto, un servicio, un proyecto, o un recurso concreto.

Una entidad puede ser física –tangible– (p.e. un computador) o abstracta (p.e. un programa en C).

FORMALIZACIÓN DE MÉTRICAS

Atributo

Definición

Una propiedad mensurable, física o abstracta, que comparten todas las *entidades* de una *categoría de entidad*.

Relaciones

Un atributo solo puede pertenecer a una categoría de entidad.

Una medición se realiza sobre los atributos de una entidad

Un atributo tiene definida cero, una o varias métricas.

Un atributo está relacionado con uno o más conceptos medibles.

Ejemplos

El *atributo* "tamaño de código fuente", como *atributo* de la *categoría de entidad* "programas en C" va a ser diferente del *atributo* "tamaño de código fuente" de la *categoría de entidad* "programa en Ada".

El "tamaño" de un "módulo en C" no es el mismo atributo (aunque tenga el mismo nombre) que el "tamaño" de un "diagrama de clases UML".

FORMALIZACIÓN DE MÉTRICAS

Forma de medir

Definición

Conjunto de operaciones cuyo objeto es determinar el valor de una *medida*. Una *forma de medir* puede ser un *método de medición*, *función de cálculo* o *modelo de análisis*.

Relaciones

Una *forma de medir* es ejecutada en cada *medición*, dependiendo de la *métrica* que calcula.

Ejemplos

Véase los ejemplos de *método de medición*, *función de cálculo* o *modelo de análisis*, ya que la *forma de medir* es una generalización de ellos.

FORMALIZACIÓN DE MÉTRICAS

Medición (Acción de medir)

Definición

La acción que permite obtener el valor de una *medida* para un *atributo* de una *entidad*, usando una *forma de medir*.

Relaciones

Cada *medición* produce una *medida*.

Una *medición* usa una *métrica*, la cual debe estar definida para el atributo objeto de la medición.

Una *medición* es llevada a cabo usando una *forma de medir*. Esta forma de medir es la que define la *métrica* usada en la *medición*.

Una *medición* se realiza para un *atributo* de una *entidad*. El atributo ha de estar definido para la categoría a la que pertenece dicha entidad.

Ejemplos

Acción consistente en usar la forma de medir "contar el número de

FORMALIZACIÓN DE MÉTRICAS

Métrica

Definición

Una *forma de medir* (*método de medición*, *función de cálculo* o *modelo de análisis*) y una *escala*, definidas para realizar *mediciones* de uno o varios *atributos* .

Relaciones

Una métrica está definida para uno o más *atributos*

Dos *métricas* pueden relacionarse mediante una función de transformación. El tipo de dicha función de transformación va a depender del tipo de *escala* de ambas métricas.

Una métrica puede expresarse en una *unidad* (sólo para métricas cuya escala sea de tipo intervalo o ratio)

Ejemplos

a métrica "líneas de código" puede ser definida para realizar mediciones del "tamaño" de un "módulo en C" y para realizar mediciones del "tamaño" de un "programa en Ada".

FORMALIZACIÓN DE MÉTRICAS

Medida

Definición

Resultado de una *medición*.

Relaciones

Una *medida* es el resultado de una *medición*

Ejemplos

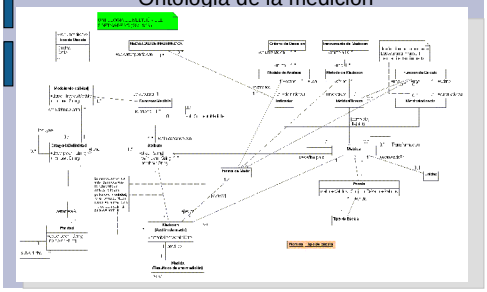
35.000 líneas de código, 200 páginas, 50 clases.

5 meses desde el comienzo al fin del proyecto.

0,5 fallos por cada 1.000 líneas de código.

FORMALIZACIÓN DE MÉTRICAS

Ontología de la medición



FORMALIZACIÓN DE MÉTRICAS

- Ejemplo

Supongamos una organización que lleva a cabo un proyecto de desarrollo de un software X. En un determinado momento el responsable del proyecto necesita saber si la productividad es adecuada, es decir:

- La necesidad de información es conocer el nivel de productividad de los programadores del proyecto en comparación con lo habitual en otros proyectos en la organización.

FORMALIZACIÓN DE MÉTRICAS

- Las métricas a utilizar podrían ser:
- *Directas*:
  - LCF (líneas de código fuente escritas). El método de medición es contar las líneas utilizando como instrumento una herramienta CASE.
  - HPD (horas-programador diarias). El método de medición es que el responsable del proyecto anota cada día las horas dedicadas por los programadores al proyecto.
  - CHP (coste por hora-programador, en unidades monetarias). El método de medición es consultar el plan del proyecto, donde se tuvo que indicar este

FORMALIZACIÓN DE MÉTRICAS

- *Indirectas*:
  - HPT (horas-programador totales). La función de cálculo es un sumatorio de las HPD de cada día: [métrica indirecta definida en base a sólo 1 métrica directa].
  - LCFH (líneas de código fuente por hora de programador). La función de cálculo es una simple división: LCFH = LCF / HPT [métrica indirecta definida en base a 2 métricas directas].
  - CTP (coste total actual del proyecto, en unidades monetarias). La función de cálculo establece que el CTP es el producto del coste unitario de cada hora por el total de horas empleadas: CTP = CHP * HPT [métrica indirecta definida en base a 2 métricas, una directa y otra indirecta].
  - CLCF (coste por línea de código fuente). CLCF = LCF/CTP.

FORMALIZACIÓN DE MÉTRICAS

- *Indicadores*:
  - PROD (productividad de los programadores). El modelo de análisis utiliza los valores de las métricas LCF, HPT, LCFH y CTP para establecer un valor cualitativo de la productividad de los programadores en este proyecto. El modelo se basa en extraer de una base histórica de proyectos de la organización los valores medios de LCF, HPT, LCFH (LCFHvm) y CTP del subconjunto de proyectos similares (aquellos que tienen LCF entre el 80% y el 120%). Los criterios de decisión establecidos son:
    - LCFH/LCFHvm < 0'70 => PROD='muy baja'.
    - 0'70 ≤ LCFH/LCFHvm < 0'90 => PROD='baja'.
    - 0'90 ≤ LCFH/LCFHvm < 1'10 => PROD='normal'.

CONTENIDOS

- Introducción
- Formalización de métricas
- Medición en IS
- Conclusiones

MEDICIÓN EN IS

- Ingeniería del Software:
  - Colección de técnicas utilizadas cuando se aplica una aproximación ingenieril a la construcción de productos software.
  - Con aproximación ingenieril queremos decir gestionar, presupuestar, planificar, modelar, analizar, diseñar, implementar, testear y mantener.
- Estas actividades junto con las herramientas y técnicas para integrarlas se han visto como la solución a la llamada crisis del software:
  - poca calidad, sistemas entregados tarde o con el presupuesto sobrepasado

MEDICIÓN EN IS

- Las disciplinas ingenieriles necesitan métodos desarrollados en base a modelos y teorías.
  - Cuando diseñamos circuitos eléctricos usamos teorías como la Ley de Ohm que describe la relación entre resistencia, intensidad y voltaje en un circuito.
- Estas teorías han evolucionado teniendo la medición como base, también usamos la medición para poder aplicarlas.
- Por lo tanto, para poder construir un circuito con una intensidad y corriente específicas, sabemos qué voltaje se requiere y tenemos los instrumentos que nos permiten medir si tenemos ese voltaje en una pila dada.
- Sería difícil imaginar cómo las disciplinas de

MEDICIÓN EN IS

- Sin embargo, la medición ha sido completamente ignorada dentro de la Ingeniería del Software:
  - Todavía fallamos en dar objetivos medibles cuando desarrollamos productos software. Por ejemplo, se dice que será amigable, fiable y mantenible, sin especificar qué significa esto en términos medibles.
  - Fallamos al medir diferentes componentes que permiten calcular los costes reales de los proyectos software. Por ejemplo, normalmente no sabemos cuánto tiempo fue realmente invertido en el diseño, comparado con el testeo.
  - No intentamos cuantificar la calidad de los productos que producimos. Estos, por ejemplo, significa que no podemos decir a un usuario cómo de fiable va a ser un producto en términos de fallos en un periodo dado de uso.
  - Todavía nos basamos en evidencias anecdóticas para convencernos de comprobar otra nueva tecnología o herramienta de desarrollo revolucionaria.

MEDICIÓN EN IS

- Solemos ver informes que hacen afirmaciones como que el 80% de los costos del software son de mantenimiento o que hay una media de 55 errores en cada 1.000 líneas de código.
- Sin embargo, no se dice cómo se obtuvieron esos resultados, cómo se diseñaron y ejecutaron los experimentos, qué entidades fueron medidas y cómo y cuales fueron los márgenes de error, sin estos datos no podemos repetir las mediciones de forma objetiva en nuestros entornos para tener comparaciones con los estándares de la industria.
- Todos estos problemas derivados de una medición insuficiente se agravan por una falta de aproximación rigurosa a la medición

MEDICIÓN EN IS

- Como ya hemos dicho, se dice que la producción software está en crisis, tiene costes excesivos, baja productividad y poca calidad.
- En resumen, la producción software está generalmente fuera de control y se ha llegado a sugerir que esto es debido a que no medimos.

MEDICIÓN EN IS

- Las actividades de medición deben tener objetivos claros, que serán los que determinarán los tipos de entidades o atributos que deben ser medidos, los objetivos variarán de acuerdo con el tipo de personal involucrado en los diferentes niveles de desarrollo y uso del software.
- A continuación vamos a dar una lista de diferentes cosas que se necesita medir para alcanzar diferentes objetivos.
- Además se van a presentar desde el punto de vista de los gestores y los ingenieros.

MEDICIÓN EN IS

Gestores:

- Necesitan medir el costo de los diferentes procesos de la producción software.
- Necesitan medir la productividad de la plantilla para determinar los pagos.
- Necesitan medir la calidad de los productos software para poder comprobar diferentes proyectos.
- Necesitan definir objetivos medibles para los proyectos. Por ejemplo, que tan fiable debe ser el sistema final.
- Necesitan medir repetidamente atributos de recursos y procesos con el fin de determinar los factores que afectan al coste y a la productividad.
- Necesitan evaluar la eficacia de diferentes métodos

MEDICIÓN EN IS

Ingenieros:

- **Necesitan monitorizar la calidad de la evolución de los sistemas a través de la medición de los procesos. Esto incluye los cambios hechos durante el diseño o los errores detectados durante diferentes fases de testeo.**
- **Necesitan especificar requisitos de calidad y realización en términos medibles estrictamente. De forma que estos requisitos sean testeables. Por ejemplo, el requisito de que un sistema sea fiable debe ser sustituido por: el tiempo medio hasta un fallo debe ser mayor de 15 horas de CPU.**
- **Necesita medir atributos de proceso y producto para certificación. Por ejemplo, la certificación puede requerir propiedades medibles del producto, como menos de 20 errores detectados por cada sitio de beta-testeo o, no habrá módulos con más de 100 líneas.**
- **Necesita medir atributos de productos existentes o**

MEDICIÓN EN IS

- Por tanto, tenemos mediciones utilizadas:
  - Por un lado para la evaluación
  - Por otro para predecir características importantes de los proyectos.
- Alcance de las métricas del Software
  - medidas y modelos de estimación de coste y esfuerzo
  - modelos y medidas de productividad
  - recolección de datos
  - medidas y modelos de calidad
  - modelos de fiabilidad
  - complejidad computacional o algorítmica.

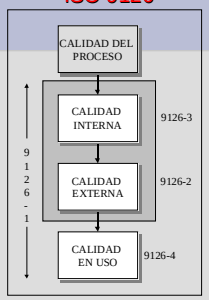
CONTENIDOS

- Introducción
- Formalización de métricas
- Medición en IS
- La ISO9126
- Conclusiones

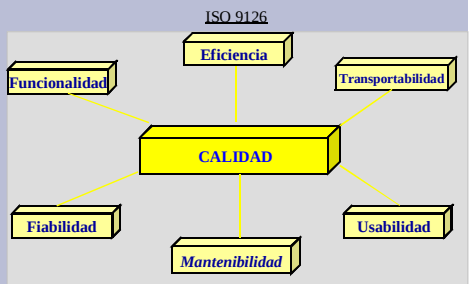
ISO 9126

- Este estándar está pensado para los desarrolladores, adquirentes, personal de aseguramiento de calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software.
- Por tanto, puede servir para validar la completitud de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, criterios de aseguramiento de la calidad, etc.
- La calidad de cualquier proceso del ciclo de vida del software (estándar ISO 12.207) influye en la calidad del producto software que, a su vez, contribuye a mejorar la calidad en el uso del producto.
- La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del

ISO 9126



ISO 9126



ISO 9126

- **Funcionalidad:** capacidad del producto software para proporcionar funciones que satisfagan las necesidades especificadas e implícitas.
- **Fiabilidad**: capacidad del producto software para mantener un nivel especificado de rendimiento.
- **Usabilidad**: la capacidad del producto software de ser entendido, aprendido, utilizado y atractivo al usuario.
- **Eficiencia:** la capacidad del producto software para proporcionar el rendimiento apropiado, relativo a la cantidad de recursos utilizados.
- **Mantenibilidad:** la capacidad del producto software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, en los requisitos o en las especificaciones funcionales.

ISO 9126

La **funcionalidad** se subdivide en cinco subcaracterísticas:

- Adecuación: la capacidad del producto software para proporcionar un conjunto apropiado de funciones para tareas específicas y objetivos de los usuarios.
- Exactitud: la capacidad del producto software para proporcionar los resultados o efectos correctos y con el grado de precisión acordado.
- Interoperabilidad: la capacidad del producto software para interactuar con uno o más sistemas especificados.
- Seguridad: referido a la capacidad del producto software para proteger la información y los datos.
- Conformidad: la capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativos a la funcionalidad.

ISO 9126

La **fiabilidad** se subdivide en cuatro subcaracterísticas:

- Madurez: la capacidad del producto software para evitar fallos provocados por errores en el software.
- Tolerancia a fallos: la capacidad del producto software para mantener un nivel de rendimiento determinado en caso de defectos en el software o incumplimiento de su interfaz.
- Recuperabilidad: la capacidad del producto software para restablecer un determinado nivel de rendimiento y recuperar los datos afectados directamente en caso de ocurrir un fallo.
- Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.

ISO 9126

La **usabilidad** se subdivide en cinco subcaracterísticas:

- Comprensibilidad: la capacidad del producto software para permitir al usuario que entienda si el software es adecuado, y como debe utilizarse para determinadas tareas y bajo ciertas condiciones de uso.
- Facilidad de aprendizaje: la capacidad del producto software para permitir al usuario aprender su aplicación.
- Operabilidad: la capacidad del producto software para permitir que el usuario lo opere y lo controle.
- Atracción: la capacidad del producto software para atraer al usuario.
- Conformidad: la capacidad del producto software para adaptarse a estándares, convenciones, guías de estilo y regulaciones relacionadas con la usabilidad.

ISO 9126

La **eficiencia** se subdivide en tres subcaracterísticas:

- Comportamiento temporal: la capacidad del producto software para proporcionar tiempos de respuesta y de procesamiento apropiados cuando realiza sus funciones bajo condiciones determinadas.
- Utilización de recursos: la capacidad del producto software para utilizar cantidades y tipos de recursos apropiados cuando el software realiza su función bajo determinadas condiciones.
- Conformidad: la capacidad del producto

ISO 9126

La **mantenibilidad** se subdivide en cinco subcaracterísticas:

- Analizabilidad: Capacidad del producto software de diagnosticar sus deficiencias o causas de fallos, o de identificar las partes que deben ser modificadas.
- Cambiabilidad: Capacidad del producto software de permitir implementar una modificación especificada. La implementación incluye los cambios en el diseño, el código y la documentación.
- Estabilidad: Capacidad del producto software de evitar los efectos inesperados de las modificaciones.
- Facilidad de prueba: Capacidad del producto software de permitir validar las partes modificadas.
- Conformidad: Capacidad del producto software de

ISO 9126

La **portabilidad** se subdivide en cinco subcaracterísticas:

- Adaptabilidad: la capacidad del producto software para ser adaptado para ambientes determinados sin realizar acciones o aplicar medios, más que los proporcionados para este propósito para el software considerado.
- Facilidad de instalación: la capacidad del producto software para ser instalado en un ambiente determinado.
- Coexistencia: la capacidad del producto software para coexistir con otro software independiente en un ambiente común compartiendo recursos.
- Reemplazabilidad: la capacidad del producto software para ser utilizado en lugar de otro producto de software para el mismo propósito en el mismo ambiente.
- Conformidad: la capacidad del producto software para

CONTENIDOS

Conclusiones

- Importancia de las métricas
- Utilidad de la formalización
- Aplicabilidad de la medición en IS
- Importancia de la ISO9126