# Informe Ejecutivo

María Esther Ruilova Rojas

21 de abril de 2008

Métricas del Producto para el Software (Ingeniería de software Enfoque Práctico)

## 1 Introducción

Las métricas del software permiten medir de forma cuantitativa la calidad de sus atributos internos del producto, esto permite al ingeniero evaluar la calidad antes de su construcción. Es importante establecer ¿qué es la calidad del software?, ¿quién lo hace?, ¿Por qué es importante?, ¿Cuáles son los pasos? Para determinar la calidad, ¿Cuál es el producto obtenido?, ¿Cómo estar seguro de hacerlo correctamente? Todas estas interrogantes se determinarán a lo largo del desarrollo del presente informe. Aspectos a considerar tales como hacer una distinción entre medida, métrica e indicador, qué factores de calidad se toman en cuenta.

## 2 Desarrollo del tema

La medición, considerada como un elemento clave en cualquier proceso de ingeniería.

Medición: Proceso mediante el cual se asignan números o símbolos a los atributos reales para definirlas de acuerdo con reglas claramente establecidas.

### 2.1 Calidad General

Calidad del Software es el cumplimiento de los requisitos de funcionalidad y desempeño explícitamente establecidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas que se esperan de todo software desarrollado profesionalmente.

Con esta definición se destacan tres puntos importantes:

1. Los requisitos del software son la base de las medidas de calidad. La falta de concordancia con estos requisitos es una falta de calidad.

2. Los estándares especificados definen un conjunto de criterios de desarrollo que guían la ingeniería del software. Si no se siguen los criterios, el resultado será, casi seguramente, la falta de calidad.

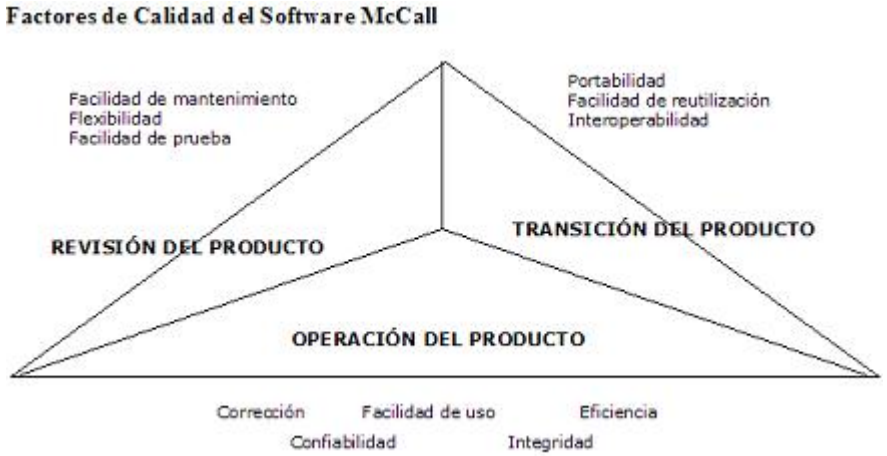
3. A menudo se soslaya un conjunto de requisitos implícitos. Si el software cumple con sus requisitos explícitos pero no con los implícitos, la calidad del software estará en duda.

### 2.1.1 Factores de Calidad de McCall

Estos factores se dividen en dos grupos muy importantes:

1. Los que se miden directamente.

2. Los que solo se miden indirectamente. McCall, Richards y Walters propusieron unos factores los cuales se concentran en tres aspectos importantes de un producto de software: sus características operativas, su capacidad para experimentar cambios y su capacidad para adaptarse a nuevos entornos.




*Corrección:* Grado en que cumple el programa con su especificación y satisface los objetivos que propuso el cliente.

*Confiabilidad:* Grado en que se esperaría que un programa desempeñe su función con la precisión requerida.

*Eficiencia:* Cantidad de código y de RR. De cómputo necesarios para que un programa realice su función.

*Integridad:* Grado de control sobre el acceso al S/W o los datos por parte de personas no autorizadas.

*Facilidad de uso*: Esfuerzo necesario para prender, operar y preparar los datos de entrada de un programa e interpretar la salida.

*Facilidad de mantenimiento:* Esfuerzo necesario para localizar y corregir un error en un programa.

*Flexibilidad*: Esfuerzo necesario para modificar un programa en operación.

*Facilidad de prueba:* Esfuerzo que demanda probar un programa con el fin de asegurar que realiza su función.

*Portabilidad:* Esfuerzo necesario para transferir el programa de un entrono de hardware o software a otro.

*Facilidad de reutilización:* grado en que un programa puede reutilizarse en otras aplicaciones.

*Interoperabilidad:* Esfuerzo necesario para acoplar un sistema con otro.

Muchas de estas métricas solo se miden subjetivamente.

### 2.1.2 Factores de calidad del estándar ISO 9126

Se desarrolló como un intento por identificar los atributos de calidad para el software de computadora. El estándar identifica 6 puntos:

- Funcionalidad
- Confiabilidad
- Facilidad de uso
- Eficiencia
- Facilidad de mantenimiento
- Portabilidad

## 2.2 Un marco conceptual para las métricas del producto

### 2.2.1 Medidas, métricas e indicadores

Aunque estos tres términos suelen utilizarse de manera intercambiable, es necesario especificar las diferencias entre éstos.

*Medida:* proporciona indicación cuantitativa de la extensión, la cantidad, la dimensión, la capacidad o el tamaño de algún atributo de un producto o proceso.

*Medición:* acto de determinar una medida.

*Métrica:* Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo determinado.

Un ingeniero de software recopila medidas y desarrolla métricas para obtener los indicadores.

*Indicador:* métrica o combinación de métricas que proporcionan conocimientos. Estos conocimientos le permiten al jefe de proyecto o a los ingenieros de software ajustar el proceso, el proyecto o el producto para que las cosas mejores.

### 2.2.2 El reto de las métricas del producto

El peligro de tratar de encontrar medidas que caractericen tantos atributos diferentes es que inevitablemente las medidas tienen que satisfacer objetivos que entran en conflicto entre sí. Esto se opone a la teoría de que cada medición debe ser representativa. Aunque la afirmación de Fenton es correcta, muchas personas argumentan que la medición del producto realizada durante las primeras etapas del proceso de software proporciona a los ingenieros un mecanismo consistente y objetivo para evaluar la calidad.

### 2.2.3 Principios de medición

Roche sugiere un proceso de medición al que caracterizan cinco actividades:

- *Formulación.* Derivación de medidas y métricas apropiadas para la representación del software que se considera.
- *Recolección.* Mecanismo con que se acumulan los datos necesarios para derivar las métricas formuladas.
- *Análisis.* Cálculo de las métricas y la aplicación de herramientas matemáticas.
- *Interpretación.* Evaluación de las métricas en un esfuerzo por conocer mejor la calidad de la representación.
- *Retroalimentación.* Recomendaciones derivadas de la interpretación de las métricas del producto transmitidas al equipo del software.

Existen principios que son representativos de muchos otros que podrían proponerse para caracterizar y validar las métricas:

- Una métrica debe tener propiedades matemáticas deseables.
- Cuado una métrica representa una característica de software que aumenta cuando se presentan rasgos positivos o que disminuye al encontrar rasgos indeseables, el valor de la métrica debe aumentar o disminuir en el mismo sentido.
- Cada métrica debe validarse empíricamente en una amplia variedad de contextos antes de publicarse o aplicarse a la toma de decisiones.

### 2.2.4 Medición del Software Orientado a Objetos

El paradigma *objetivo/pregunta/métrica* (OPM) desarrollado por Basili y Weiss es considerado una técnica para identificar significativa métricas las cuales son aplicables en cualquier parte del proceso de software; destaca la necesidad de: 1. Establecer un objetivo de medición que sea específico para la actividad del proceso o las características del producto que se está evaluando. 2. Definir un conjunto de preguntas que deben responderse con el fin de alcanzar el objeto. 3. Identificar métricas bien formadas que ayuden a responder esas preguntas.

### 2.2.5 Los atributos de las métricas efectivas del software

Ejiogu define un conjunto de atributos que toda métrica efectiva del software debe abarcar:

- Simples y calculable
- Consistentes y objetivas
- Consistentes en el uso de unidades y dimensiones
- Independientes del lenguaje de programación
- Mecanismos efectivos para la retroalimentación de alta calidad

### 2.2.6 Panorama de las métricas del producto

**Métricas para el modelo de análisis**

Incluyen aspectos como:

- Funcionalidad entregada
- Tamaño del sistema
- Calidad de la especificación

**Métricas para el modelo de diseño**

- Métricas arquitectónicas
- Métricas al nivel de componente
- Métricas de diseño de la interfaz
- Métricas especializadas en diseño orientado a objetos

**Métricas para el código fuente**

Se usan para evaluar su complejidad, además la facilidad con que se mantiene y prueba.

- Métricas de Halstead

- Métricas de complejidad
- Métricas de longitud

**Métricas para pruebas**

Ayudan a diseñar casos de prueba efectivos y evaluar la eficacia de las pruebas.

- Métricas de cobertura de instrucciones y ramas
- Métricas relacionadas con los defectos
- Efectividad de la prueba
- Métricas en el proceso

En muchos modelos las métricas de un modelo pueden aplicarse en actividades posteriores a la ingeniería del software.

## 2.3 Métricas para el modelo de análisis

### 2.3.1 Métricas basadas en la función

La métrica de punto de función (PF) es para medir la funcionalidad que entrega un sistema. Se usa para:

1. estimar el costo o el esfuerzo requerido para diseñar, codificar y probar el software
2. predecir el número de errores que se encontrarán durante la prueba
3. pronosticar el número de componentes, de líneas de código proyectadas, o ambas, en el sistema implementado.

Los valores del dominio de la información se definen de la siguiente manera:
Número de entradas externas (EE).
Número de salidas externas (SE)
Número de consultas externas (CE)
Número de archivos lógicos internos (ALI)
Número de archivos de interfaz externos (AIE)

Para calcular los puntos de función se usa la siguiente relación:

$$PF = ConteoTotal * [0.65 + 0-01 * \sum(F_i)]$$

## 2.4 Métricas para el modelo de diseño

**Métricas del diseño arquitectónico**

Consideradas métricas de caja negra ya que no requieren ningún conocimiento del funcionamiento interno de un componente de software en particular.

Card y Glass definen tres medidas de la complejidad del diseño del software:

- *Complejidad Estructural* en el caso de arquitecturas jerárquicas

$$S(i) = f^2 out(i)$$

- *Complejidad de datos* indica complejidad de la interfaz interna de un módulo

$$D(i) = v(i)/\left[f_{out}(i)+1\right]$$

- *Complejidad del sistema* es la suma de las complejidades estructural y de datos.

$$C(i) = S(i) + D(i)_i) + 1\Big|$$

**Métricas para el diseño orientado a objetos**

Whitmire describe nueve características distintivas y mensurables de un Diseño OO:

- Tamaño
- Complejidad
- Acoplamiento
- Suficiencia
- Grado de avance
- Cohesión
- Primitivismo
- Similitud
- Volatilidad

### 2.5 Métricas para el código fuente

Estás métricas asignadas como cuantitativas por Halstead, se derivan después de que se ha generado el código o se estima una vez que el diseño esté completo. Las medidas son:

n1 = el número de operadores distintos que aparecen en un programa.
n2 = el número de operandos distintos que aparecen en un programa.
N1= el número total de veces que aparece el operador.
N2= el número total de veces que aparece en operando.

Halstead demuestra que la longitud N se puede estimas de la siguiente manera:

$$N = n_1 \log_2 n_1 + n_2 \log_2 n_2$$

El volumen del programa se define como:

$$V = N \log_2 (n_1 + n_2)$$

## 3 Conclusiones

Las métricas proporcionan los conocimientos necesarios para crear modelos efectivos de análisis y diseño, un código sólido y pruebas exhaustivas. Las métricas se enfocan al proceso de software en varios aspectos tales como métricas del producto, para el modelo de análisis, para el modelo de diseño, para el código fuente, para pruebas y para mantenimiento, las cuales permiten el control de calidad en cada uno de estos procesos.

## 4 Bibliografía

## References

[1] ROGER S. PRESSMAN, INGENIERÍA DEL SOFTWARE, un enfoque práctico.