

## CAPÍTULO 2

### Conceptos básicos de Métricas

Empezaremos por definir los posibles términos que se encuentran encerrados en la palabra *métrica*, porque es muy común asociarla con las palabras medición y medida, aunque estas tres son distintas. La *medición* “es el proceso por el cual los números o símbolos son asignados a atributos o entidades en el mundo real tal como son descritos de acuerdo a reglas claramente definidas” [Fenton '91]. Una *medida* “proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto” [Pressman'98]. El IEEE “*Standard Glossary of Software Engineering Terms*” define como *métrica* como “una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado” [Len O. Ejiogo '91].

Muchos investigadores han intentado desarrollar una sola métrica que proporcione una medida completa de la complejidad del software. Aunque se han propuesto docenas de métricas o medidas, cada una de éstas tienen un punto de vista diferente; y por otro lado, aunque bien se sabe que existe la necesidad de medir y controlar la complejidad del software, es difícil de obtener un solo valor de estas métricas de calidad. Aun así debería ser posible desarrollar medidas de diferentes atributos internos del programa.

Aunque todos estos obstáculos son motivo de preocupación, no son motivo de desprecio hacia las *métricas*. Es por eso que se dice que la medición es esencial, si es que se desea realmente conseguir la *calidad en software*.

Es por eso que existen distintos tipos de métricas para poder evaluar, mejorar y clasificar al software final, en donde serán manejadas dependiendo del entorno de desarrollo del software al cual pretendan orientarse.

## 2.1 ¿Qué son las métricas de software?

Michael [‘99] define las métricas de software como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el administrador junto con el empleo de estas técnicas mejorará el proceso y sus productos”. Las métricas de software proveen la información necesaria para la toma de decisiones técnicas. En la figura 2.1 se ilustra una extensión de esta definición para incluir los servicios relacionados al software como la respuesta a los resultados del cliente:

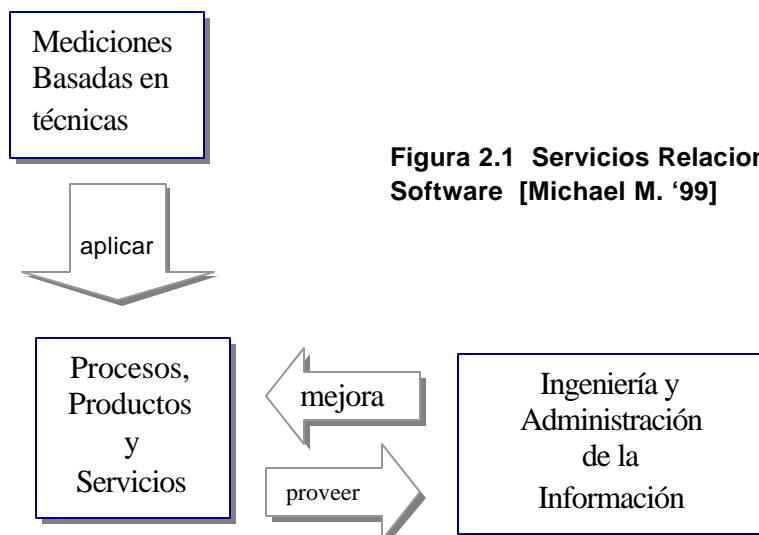


Figura 2.1 Servicios Relacionados al Software [Michael M. ‘99]

Las métricas son la maduración de una disciplina, que, según Pressman [’98] van a ayudar a la (1) evaluación de los modelos de análisis y de diseño, (2) en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y (3) ayudaran en el diseño de pruebas más efectivas; Es por eso que propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

- (1) *Formulación*: La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
- (2) *Colección*: El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- (3) *Análisis*: El cálculo de las métricas y la aplicación de herramientas matemáticas.
- (4) *Interpretación*: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
- (5) *Realimentación*: Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

Se conoce que no existe un cuerpo de principios en conjunto, puedan dirigir al desarrollo de métricas de software a que sean independientes del lenguaje, a ambientes y a metodologías de programación. Matemáticamente, estos principios son teorías e implementaciones críticas ya que una métrica, tiene ciertas propiedades matemáticas y atributos de ingeniería, así como también ciertas

realimentaciones de productividad. Es por eso que alcanzamos a responder tres preguntas fundamentales deseadas de una métrica [Fenton '91].

¿Cuánto mide? - la complejidad en la medida

¿Qué tan bien mide? - la calidad en la medida

¿Qué tanto tiempo mide? - la predicción

Las métricas de software incluyen otras varias actividades, tales como:

- Estimación de costo y el esfuerzo
- Medición de la productividad
- Acumulación de datos
- Realización de modelos y mediciones de la calidad
- Elaboración de modelos de seguridad
- Evaluación y modelos de desempeño
- Valoración de las capacidades y de la madurez
- Administración por métricas
- Evaluación del método y herramientas

## **2.2 Clasificación de Métricas**

La clasificación de una métrica de software refleja o describe la conducta del software. A continuación se muestra una breve clasificación de métricas de software, descritas por Lem O. Ejiogu ['91]:

*Métricas de complejidad:* Son todas las métricas de software que definen de una u otra forma la medición de la complejidad; Tales como volumen, tamaño, anidaciones, costo (estimación), agregación, configuración, y flujo. Estas son los puntos críticos de la concepción, viabilidad, análisis, y diseño de software.

*Métricas de calidad:* Son todas las métricas de software que definen de una u otra forma la calidad del software; Tales como exactitud, estructuración o modularidad, pruebas, mantenimiento, reusabilidad, cohesión del módulo, acoplamiento del módulo, etc. Estas son los puntos críticos en el diseño, codificación, pruebas y mantenimiento.

*Métricas de competencia:* Son todas las métricas que intentan valorar o medir las actividades de productividad de los programadores o practicantes con respecto a su certeza, rapidez, eficiencia y competencia. No se ha alcanzado mucho en esta área, a pesar de la intensa investigación académica.

*Métricas de desempeño:* Corresponden a las métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del sistema operativo o hardware. Generalmente tienen que ver con la eficiencia de ejecución, tiempo, almacenamiento, complejidad de algoritmos computacionales, etc.

*Métricas estilizadas:* Son las métricas de experimentación y de preferencia; Por ejemplo: estilo de código, indentación, las convenciones denominando de datos, las

limitaciones, etc. Pero estas no se deben confundir con las métricas de calidad o complejidad.

*Variedad de métricas:* tales como portabilidad, facilidad de localización, consistencia. Existen pocas investigaciones dentro del área.

Estas clasificaciones de métricas fortalecen la idea, de que más de una métrica puede ser deseable para valorar la complejidad y la calidad del software.

### **2.3 Diferentes enfoques de métricas**

Se han propuesto cientos de métricas para el software, pero no todas proporcionan suficiente soporte práctico para su desarrollo. Algunas demandan mediciones que son demasiado complejas, otras son tan esotéricas que pocos profesionales tienen la esperanza de entenderlas, y otras violan las nociones básicas intuitivas de lo que realmente es el software de alta calidad. Es por eso que se han definido una serie de atributos que deben acompañar a las métricas efectivas de software, por lo tanto la métrica obtenida y las medidas que conducen a ello deben cumplir con las siguientes características fundamentales:

- *Simple y fácil de calcular.* debería ser relativamente fácil de aprender a obtener la métrica y su cálculo no obligara a un esfuerzo o a una cantidad de tiempo inusuales.

- *Empírica e intuitivamente persuasiva*: la métrica debería satisfacer las nociones intuitivas del ingeniero de software sobre el atributo del producto en cuestión (por ejemplo: una métrica que mide la cohesión de un módulo debería aumentar su valor a medida que crece el nivel de cohesión).
- *Consistente en el empleo de unidades y tamaños*: el cálculo matemático de la métrica debería utilizar medidas que no lleven a extrañas combinaciones de unidades. Por ejemplo, multiplicando el número de personas de un equipo por las variables del lenguaje de programación en el programa resulta una sospechosa mezcla de unidades que no son intuitivamente concluyentes.
- *Independiente del lenguaje de programación*: las métricas deberían apoyarse en el modelo de análisis, modelo de diseño o en la propia estructura del programa. No deberían depender de los caprichos de la sintaxis o semántica del lenguaje de programación.
- *Un mecanismo eficaz para la realimentación de calidad*: la métrica debería suministrar al desarrollador de software información que le lleve a un producto final de superior calidad.

No obstante que la mayoría de las métricas de software compensan las características anteriores, algunas de las métricas usualmente empleadas no cumplen una o dos características. Un ejemplo es el punto funcional (PF, tratado en la sección 3.5), en donde se consigue argumentar que el atributo es

*consistente y objetivo* pero falla por que un equipo ajeno independiente puede no ser capaz de conseguir el mismo valor de PF que otro equipo que emplee la misma información del software. En tal caso la siguiente pregunta se manifiesta, ¿Debería entonces rechazar la medida PF?. La respuesta es por supuesto que No!., la PF aporta una visión interna útil y por lo tanto provee de un valor claro, incluso si no satisface un atributo perfectamente. Este es un ejemplo en donde el uso de las métricas dependen de los factores individuales del desarrollador o desarrolladores del software.

## **2.4 Las métricas en el proceso y dominio del proyecto**

Las métricas de *proceso* de software se emplean para fines estratégicos, y las métricas del *proyecto* de software son tácticas, éstas últimas van a permitir proporcionar al desarrollador de proyectos del software una evaluación al proyecto que sigue en continuo desarrollo, equivalentemente podrá ver los defectos que logren provocar riesgos a largo plazo (áreas problema); y observar si el área de trabajo (equipo) y las distintas tareas se ajustarán.

Existe una preocupación en la unificación de las métricas dentro del proceso del software, dado que los desarrolladores mexicanos de software no tienen la cultura de la medición. Es por eso que el uso de métricas demanda un cambio cultural, tanto en el recopilado de datos (investigación histórica), como en el cálculo de métricas (LDC, PF, métricas de calidad y orientadas a objetos) y asimismo en la evaluación (resultados obtenido), para poder comenzar un



programa de bases de métricas y justamente obtener una guía, asimismo una base de datos tanto de los procesos como de los productos y de este modo adquirir una visión más profunda de lo que se está elaborando.

Las métricas de software nos aportan una manera de estimar la calidad de los atributos internos del producto, permitiendo así al ingeniero de software valorar la calidad antes de construir el producto, así el tiempo invertido será identificando, examinando y administrando el riesgo, este esfuerzo merece la pena por muchas razones ya que habrá disminución de disturbios durante el proyecto, asimismo se podrá desarrollar una habilidad de seguir y controlar el proyecto y se alcanzará la seguridad que da planificar los problemas antes de que ocurran, además conseguiremos absorber una cantidad significativa del esfuerzo en la planificación del proyecto.

Del mismo modo existen diferentes tipos de métricas para poder evaluar, mejorar y clasificar al software desde sus inicios hasta el producto final, de las cuales se verán en los siguientes capítulos.