Universidad del Cauca ®

CALIDAD DE SOFTWARE

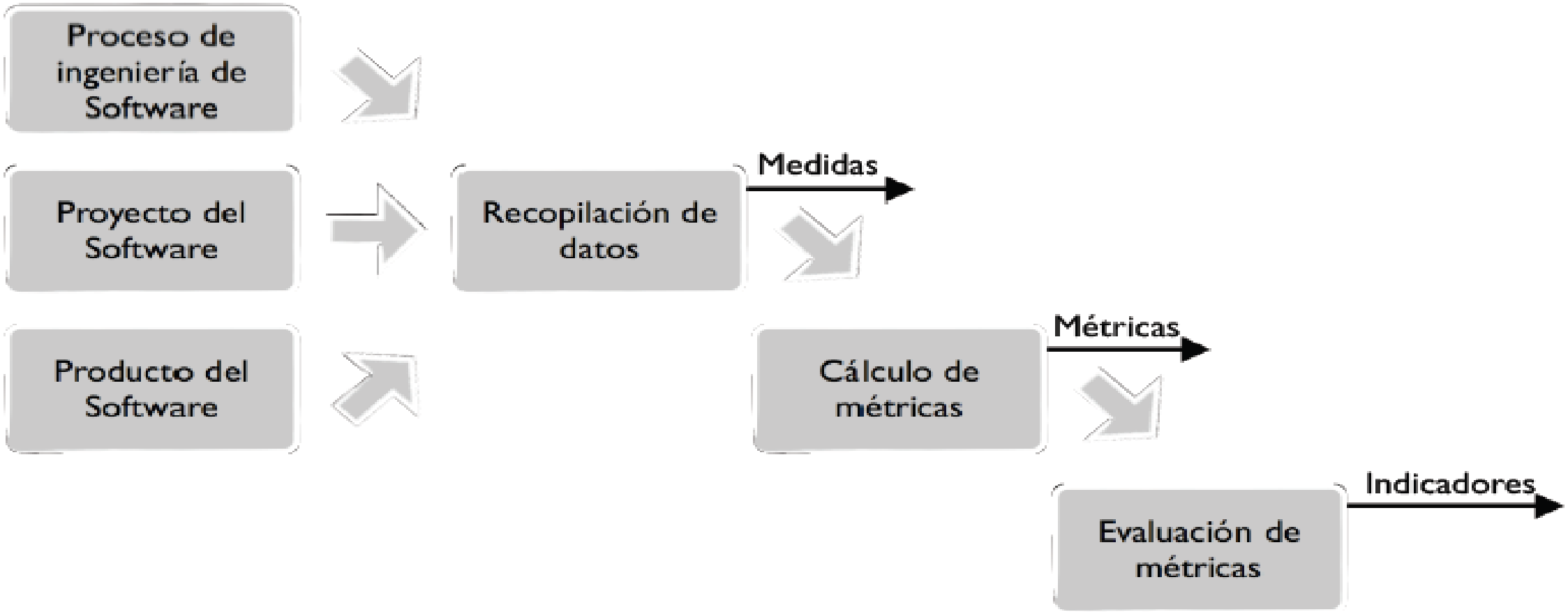
Mag. Carlos Alberto Ardila Albarracín

CAPÍTULO 5

MEDICIÓN EN EL PROCESO DE DESARROLLO DE SOFTWARE

5.5. Métricas de Producto y Proceso

PROCESO DE RECOPILACIÓN DE MÉTRICAS



Clasificación de las métricas de software según criterios:

de complejidad	Métricas que definen la medición de la complejidad: volumen, tamaño, anidaciones, y configuración.
de calidad	Métricas que definen la calidad del software: exactitud, estructuración o modularidad, pruebas, mantenimiento.
de competencia	Métricas que intentan valorar o medir las actividades de productividad de los programadores con respecto a su certeza, rapidez, eficiencia y competencia
de desempeño	Métricas que miden la conducta de módulos y sistemas de un software, bajo la supervisión del SO o hardware.
estilizadas	Métricas de experimentación y de preferencia: estilo de código, convenciones, limitaciones, etc.

Según el contexto:

Proceso:

Se recopilan de todos los proyectos, y durante un largo periodo de tiempo

Caracterizadas por:

- Control y ejecución del proyecto.
- Medición de tiempos de las fases.

Proyecto:

Permiten evaluar el estado del proyecto.

Permiten seguir la pista de los riesgos.

Producto:

Se centran en las características del software y no en cómo se fabricó.

Se consideran productos a todos los artefactos: documentos, modelos y código.

Se miden MAGNITUDES como el tamaño, la calidad, la totalidad, la volatilidad y el esfuerzo.

Ejemplo. Modelo de McCall.

Reusabilidad = (C1 * GE) + (C2 * IH) + (C3 * MD) + (C4 * AD) + (C5 * IS)

Criterio: Generalidad (GE--Generality)
Métrica 1. GE.1: UR Unit Referencing
Métrica 2. GE.2: UI Unit Implementation

Criterio: Modularidad (MO--Modularity)
Métrica 1. MO.1: MI Modular Implementation

Criterio: Auto Descriptividad (SD--Self-Descriptiveness)
Métrica 1. SD.2: EC Effectiveness of Comments

Criterio: Independencia de la Aplicación (AP--Application Independence)
Métrica 1. AP.3: AS Architecture Standardization

Tomadas del Reporte Técnico de la Federal Aviation Administration Technical Center.
(página 246 del archivo digital).
Indican cuáles son las métricas a tener en cuenta para valorar los atributos que conforman la reusabilidad.

Ejemplo. Modelo de McCall.

A modo de guía se muestran las especificaciones de 2 métricas asociadas a 2 criterios diferentes:

Criterio: Independencia de la Aplicación.
Métrica 1. AP.3: Estandarización de la Arquitectura (AS)

$$AS = \frac{\text{cantidad de líneas de código no-HOL en el módulo}}{\text{cantidad total de líneas de código fuente del módulo}}$$

REGLA:	Para cada módulo de un CSCI (Computer Software Configuration Item), responda la pregunta: "Cuántas líneas de código no-HOL (High Order Language) hay en el módulo?" (ejemplo de no-HOL es el lenguaje ensamblador). "Divida la cantidad de líneas de código no-HOL entre la cantidad total de líneas de código." (Bowen, Wigle y Tsai 1985).
RANGO:	Esta métrica genera un número real entre cero y uno.
INTERPRETACIÓN:	Los valores "útiles" para este atributo son los que se acerquen más a cero.

Ejemplo. Modelo de McCall.

Criterio: Independencia de la Aplicación
Métrica 1. AP.3: Estandarización de la Arquitectura (AS)

La razón de fijarse en lenguaje que no sea HOL es que esos lenguajes son dependientes de la máquina y en la medida que haya una menor cantidad de esas líneas, esto aumenta la valoración del atributo y por consiguiente podría incidir en un aumento de la reusabilidad.

Aporte: para ratios como este cuyos valores "buenos" para la reusabilidad son los que se acercan a cero, habría necesidad de efectuar un cálculo como este: **ASr = 1 – AS**. De modo que cuando el valor de AS se acerque más a cero, el valor de ASr será más cercano a 1 de modo que todas las métricas se puedan llevar a valores con sentido positivo (o creciente) y se puedan llevar todas a una expresión general para cuantificar la reusabilidad.

Ejemplo. Modelo de McCall.

Criterio: Auto Descriptividad
Métrica 1. SD.2: Efectividad de los Comentarios (EC)

$$EC = \frac{\text{cantidad de módulos con comentarios que cumplen estándar}}{\text{cantidad de módulos evaluados}}$$

REGLA:	Para cada módulo de un CSCI, responda la pregunta: "¿Tiene comentario de encabezado que contenga toda la información acorde con el estándar establecido?" (Bowen, Wigle, and Tsai 1985). Divida el total de respuestas afirmativas entre el número de módulos que se estén evaluando.
RANGO:	Esta métrica genera un número real entre cero y uno.
INTERPRETACIÓN:	Los valores "útiles" para este atributo son los que se acerquen más a 1.

Ejemplo. Modelo de McCall.

Criterio: Auto Descriptividad
Métrica 1. SD.2: Efectividad de los Comentarios (EC)

La razón de fijarse en que los comentarios cumplan con un estándar establecido es que podrían facilitar la comprensión de lo que hace el módulo. Esto aumenta la valoración del atributo ya que si se puede comprender, se puede evaluar con menos dificultad si se podría re-usar, y por consiguiente podría incidir en un aumento de la reusabilidad.

Aporte: para ratios como este cuyos valores “buenos” para la reusabilidad son los que se acercan a 1, no hay necesidad de calcular una expresión adicional como se hizo con la métrica anterior.

Ejemplo. Modelo de McCall.

Cuantificación de la Reusabilidad. Aún teniendo en cuenta lo expresado, ahora se plantea una primera aproximación para calcular un valor (o probabilidad) para la reusabilidad, a partir de las 2 métricas revisadas:

$$\textbf{REUSABILIDAD} = (C_1 * AS\textbf{r}) + (C_2 * EC)$$

De esta manera el rango de valores resultantes está entre 0 y 1.

$\mathbf{C_i}$ son coeficientes de regresión ($C_1 + C_2 = 1.0$) que representan la ponderación o el "peso" que se le podría asignar a cada métrica si se cree que alguna de ellas podría tener una mayor incidencia en la cuantificación de la reusabilidad.

De no ser así, en este ejemplo $C_1 = C_2 = 0.5$.

Ejemplo. Modelo de McCall.

Generalizando,
se muestra una expresión que podría cuantificar una valoración para "Reusabilidad":

Por lo tanto, se muestran los símbolos originales,
ya que no se tiene forma de saber cuáles de ellas tienen "valores buenos" cerca de cero.

REUSABILIDAD = $(C_1 * UR) + (C_2 * UI) + (C_3 * ASr) + (C_4 * MI) + (C_5 * EC)$

$\mathbf{C_i}$ son coeficientes de regresión ($C_1 + C_2 + C_3 + C_4 + C_5 = 1.0$)

Si la ponderación es equitativa $\mathbf{C_i} = 0.2$

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.

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 valor, previa consulta a un responsable de personal.

INDIRECTAS

HPT (horas-programador totales).

La función de cálculo es una sumatoria de las HPD de cada día: HPT = $\sum$HPD
[Métrica indirecta definida con base en sólo 1 métrica directa].

LCFH (líneas de código fuente por hora de programador).
La función de cálculo es una simple división: LCFH = LCF / HPD
[Métrica indirecta definida con base en 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 con base en 2 métricas, una directa y otra indirecta].

CLCF (coste por línea de código fuente).
La función de cálculo establece que el CLCF es el resultado de dividir el coste total actual del proyecto -en unidades monetarias- **entre** la cantidad de líneas de código fuente escritas: CLCF = CTP/ LCF.
[Métrica indirecta definida con base en 2 métricas, una indirecta y otra directa].

INDICADOR: **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** → valor medio de líneas de código fuente por hora de programador) y **CTP**

del subconjunto de ***proyectos similares*** (aquellos que tienen **LCF** entre el 80% y el 120%).

Y la organización establece el siguiente **CRITERIO DE DECISIÓN**:

LCFH / LCFHvm < 0.70	se interpreta como PROD='muy baja'.
0.70 <= LCFH / LCFHvm < 0.90	se interpreta como PROD='baja'.
0.90 <= LCFH / LCFHvm < 1.10	se interpreta como PROD='normal'.
1.10 <= LCFH / LCFHvm < 1.30	se interpreta como PROD='alta'.
1.30 <= LCFH / LCFHvm	se interpreta como PROD='muy alta'.

De modo que cuando se calcule el valor para el indicador (a partir de las métricas mencionadas)
se podrá evaluar y entregar un concepto sobre la situación que dicho indicador pretende reflejar.

TAMBIÉN SE PUEDEN UTILIZAR

- Número de defectos generados por desarrollador por hora
- Número de cambios a los requisitos
- Número de versiones con correcciones (patch) realizadas después de lanzar el producto
- Horas disponibles y ejecutadas por programador por semana
- Defectos descubiertos durante las pruebas
- Número de defectos introducidos al realizar una modificación.

----------FIN DEL DOCUMENTO