

Contenido

1. Defina en el contexto de la Ingeniería de Software: Medida, Medición, Métrica e Indicador..	1
2. Defina indicador de Proceso.	2
3. Defina Indicador de Proyecto.....	2
4. Enuncie el programa de métricas de proceso propuesto por Grady.	2
5. Enuncie las buenas prácticas del análisis de fallas.....	3
6. Defina Métricas orientadas al Tamaño y Métricas Orientadas a la Función. De Ejemplos.	3
7. Defina Puntos de Función.	4
8. Enuncie las características del dominio de información tenidas en cuenta en el cálculo de puntos de función.	4
9. Describa el cálculo de puntos de función.....	4
10. Enuncie medidas de calidad de un proyecto software.	5
11. Defina eficacia de eliminación de defectos.....	6
12. Fundamente el establecimiento de una línea base de métricas.	6
13. Caracterice los atributos deseables de una línea base.	6
14. De un esquema de proceso de recopilación de métricas del software.	6
15. Enuncie los tres componentes de un programa de medida exacto propuesto por Basili.	7
16. Enuncie las tres etapas del paradigma de mejora de calidad asociado al método OPM(Objetivo/Pregunta/Métrica).....	7
17. Enuncie los pasos para el establecimiento de un programa de métricas de software.....	7

1. Defina en el contexto de la Ingeniería de Software: Medida, Medición, Métrica e Indicador.

Hay cuatro razones para medir los procesos del software, los productos y los recursos:

- Caracterizar
- Evaluar
- Predecir
- Mejora
- Dentro del contexto
- de la ingeniería del software, una *medida* proporciona
- una indicación cuantitativa de la extensión,
- cantidad, dimensiones, capacidad o tamaño de algunos
- atributos de un proceso o producto.

Dentro del contexto de la ingeniería del software, Medida, Medición, Métrica e Indicador.

Medida : proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. **Por ejemplo:** el número de errores descubiertos en la revisión de un módulo

Medición: el acto de determinar una medida. Aparece como resultado de la recopilación de uno o varios aspectos de los datos. **Por ejemplo:** se investiga un número de revisiones de módulos para recopilar medidas del número de errores encontrados durante cada revisión.

Métrica: una medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. La métrica del software relata de alguna forma las medidas individuales sobre algún aspecto. **Por ejemplo:** el número medio de errores encontrados por revisión o el número medio de errores encontrados por persona y hora en revisiones.

Indicador: Un indicador es una métrica o una combinación de métricas que proporcionan una visión profunda del proceso del software, del proyecto de software o del producto en sí. Un indicador proporciona una visión profunda que permite al gestor de proyectos o a los ingenieros de software ajustar el producto, el proyecto o el proceso para que las cosas salgan mejor.

2. Defina indicador de Proceso.

Los indicadores de proceso permiten a una organización de ingeniería del software tener una visión profunda de la eficacia de un proceso ya existente (por ejemplo: el paradigma, las tareas de ingeniería del software, productos de trabajo e hitos). También permiten que los gestores evalúen lo que funciona y lo que no

3. Defina Indicador de Proyecto

Los indicadores de proyecto permiten al gestor de proyectos del software

- (1) evaluar el estado del proyecto en curso;
- (2) seguir la pista de los riesgos potenciales:
- (3) detectar las 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 del proyecto en controlar la calidad de los productos de trabajo del software.

4. Enuncie el programa de métricas de proceso propuesto por Grady.

- Utilice el sentido común y una sensibilidad zativa al interpretar datos de métricas.
- Proporcione una retroalimentación regular para particulares y equipos que hayan trabajado en la recopilación de medidas y 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 o equipos.

- Los datos de métricas que indican un área de problemas no se deberían considerar. Estos datos son meramente un indicador de mejora de proceso.
- No se obsesione con una sola métrica y excluya otras métricas importantes.

5. Enuncie las buenas prácticas del análisis de fallas.

1. Todos los errores y defectos se categorizan por origen (por ejemplo: defectos en la especificación, en la lógica, no acorde con los estándares).
2. Se registra tanto el coste de corregir cada error como el del defecto.
3. El número de errores y de defectos de cada categoría se cuentan y se ordenan en orden descendente.
4. Se computa el coste global de errores y defectos de cada categoría.
5. Los datos resultantes se analizan para detectar las categorías que producen el coste más alto para la organización.
6. Se desarrollan planes para modificar el proceso con el intento de eliminar (o reducir la frecuencia de apariciones de) la clase de errores y defectos que sean más costosos.

6. Defina Métricas orientadas al Tamaño y Métricas Orientadas a la Función. De Ejemplos.

Métricas orientadas al tamaño: Proviene de la normalización de medidas de calidad y/o productividad considerando el tamaño del software producido. Algunas de ellas pueden ser:

- Errores por KLDC
- Defectos por KLDC,
- E por KLDC
- páginas de documentación por KLDC.

Además, se pueden calcular otras métricas interesantes:

- errores por persona-mes LDC por persona-mes
- por página de documentación

Métricas orientadas a la Función: Utilizan la <funcionalidad> como valor de normalización, y como la funcionalidad no se puede medir directamente, se deriva indirectamente de otras medidas directas

7. Defina Puntos de Función.

Es un método utilizado para medir el tamaño del software, que se derivan de una relación empírica según las medidas contables del dominio de información, y de las evaluaciones de complejidad del software

8. Enuncie las características del dominio de información tenidas en cuenta en el cálculo de puntos de función.

- Número de entradas de usuario: Se cuentan entradas que proporcionan datos de la aplicación
- Número de salidas de usuario: Se cuentan salidas que proporcionan información de la aplicación
- Número de peticiones: Son entradas interactivas que generan una respuesta del software
- Numero de Archivos: Se cuenta cada archivo maestro lógico
- Número de interfaces externas: Todas las interfaces utilizadas para transmitir Info de un sistema a otro

9. Describa el cálculo de puntos de función.

Parámetros de medición	Factor de ponderación				
	Cuenta	Simple	Medio	Complejo	
Número de entradas de usuario	<input type="text"/>	x 3	4	6	= <input type="text"/>
Número de salidas de usuario	<input type="text"/>	x 4	5	7	= <input type="text"/>
Número de peticiones de usuario	<input type="text"/>	x 3	4	6	= <input type="text"/>
Número de archivos	<input type="text"/>	x 7	10	15	= <input type="text"/>
Número de interfaces externas	<input type="text"/>	x 5	7	10	= <input type="text"/>
Cuenta total	<input type="text"/>				

FIGURA 4.5. Cálculo de puntos de función.

$$PF = \text{cuenta-total} \times (0,65 + 0,01 \times 6(F_i))$$

cuenta-total es la suma de las entradas en la figura de cálculo, y F_i son <valores de ajuste de complejidad>, que se obtienen respondiendo 14 preguntas. Cada respuesta afirmativa añade complejidad

1. ¿Requiere el sistema copias de seguridad? y de recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?

7. ¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualizan los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos o las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están incluidas en el diseño la conversión y la instalación?
13. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizada por el usuario?

10. Enuncie medidas de calidad de un proyecto software.

- Corrección: El grado en el que el software lleva a cabo su acción requerida
- Facilidad de mantenimiento: La facilidad con la que se puede corregir un programa si se encuentra un error
- Integridad: Mide la capacidad del sistema para resistir ataques contra su seguridad. Hay que definir amenaza (probabilidad de que un ataque ocurra en un tiempo determinado) y seguridad (probabilidad de poder repeler dicho ataque).
- Facilidad de uso: Si el programa no es amigable con el usuario, tiende a fracasar. Se mide por:
 - 1) Habilidad intelectual o física requerida para aprenderlo
 - 2) El tiempo que requiere ser eficiente en el uso del sistema
 - 3) Aumento de la productividad cuando alguien lo usa
 - 4) Valoración de los usuarios hacia el sistema

11. Defina eficacia de eliminación de defectos.

EED es una medida de la habilidad para filtrar las actividades de garantía de calidad, al aplicarse todas las actividades. Se define por la fórmula:

$$EED = E / (E + D)$$

E: Errores encontrados (antes de la entrega)

D: Defectos después de la entrega.

El valor ideal de EED es 1. No se han encontrado defectos en el software.

12. Fundamente el establecimiento de una línea base de métricas.

Estableciendo una línea base de métricas se obtienen beneficios a nivel de:

- Proceso
- Proyecto
- Producto

Esta línea base, comprende los siguientes pasos:

1. Recopilación de datos. (Medidas)

Requiere una investigación histórica de los proyectos para reconstruir los datos requeridos

2. Cálculo de métricas (Métricas)

Se hace el cálculo de métricas una vez se han determinado las medidas. Pueden abarcar una gran cantidad de métricas:

- LDC y PF
- De calidad
- Del proyecto

13. Caracterice los atributos deseables de una línea base.

1) Los datos deben ser exactos, sin conjeturas

2) Se debe reunir datos de la mayor cantidad de proyectos posibles

3) Las medidas deben ser consistentes, por ejemplo una LDC debe interpretarse de igual forma en todos los proyectos

4) Las aplicaciones deben ser semejantes a la estimación (no sirve usar una línea base de un Batch para una app en tiempo real)

14. De un esquema de proceso de recopilación de métricas del software.

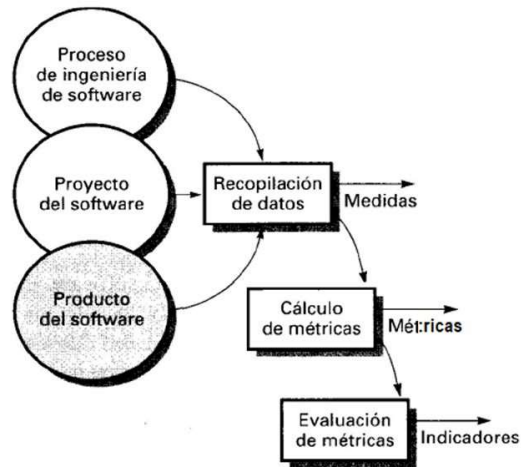


FIGURA 4.7. Proceso de recopilación de métricas del software.

15. Enuncie los tres componentes de un programa de medida exacto propuesto por Basili.

- 1) Un proceso donde se puedan articular metas y objetivos para sus proyectos
- 2) Un proceso donde esas metas se puedan traducir a los datos del proyecto que reflejen dichas metas u objetivos en términos de software
- 3) Un proceso que interprete los datos del proyecto, para entender los objetivos

16. Enuncie las tres etapas del paradigma de mejora de calidad asociado al método OPM(Objetivo/Pregunta/Métrica).

- 1) El proceso debe llevar a cabo una auditoria, estableciendo metas u objetivos de calidad, y usando herramientas que permitan que se cumplan esos objetivos.
- 2) El proceso debe ejecutar un proyecto y chequear los datos relacionados con esas metas u objetivos. Se debe llevar a cabo con un proceso de actuación sobre los datos cuando el proyecto corre peligro.
- 3) El proceso debe analizar los datos del segundo paso, para hacer sugerencias de mejora. Esto implica analizar problemas en las etapas de recolección de datos, en la implementación de directivas de calidad y en la interpretación de datos.

17. Enuncie los pasos para el establecimiento de un programa de métricas de software.

- 1) Identificar objetivos de negocio
- 2) Identificar lo que se desea saber o aprender

- 3) Identificar subobjetivos
- 4) Identificar entidades y atributos de los subobjetivos
- 5) Formalizar los objetivos de medición
- 6) Identificar preguntas que puedan cuantificarse y los indicadores a utilizar para obtener objetivos de medición
- 7) Identificar elementos de datos a recoger para construir indicadores que respondan a las preguntas planteadas
- 8) Definir medidas a usar para que las definiciones sean operativas
- 9) Identificar las acciones que serán tomadas para tomar las medidas indicadas
- 10) Preparar un plan para implementar estas medidas