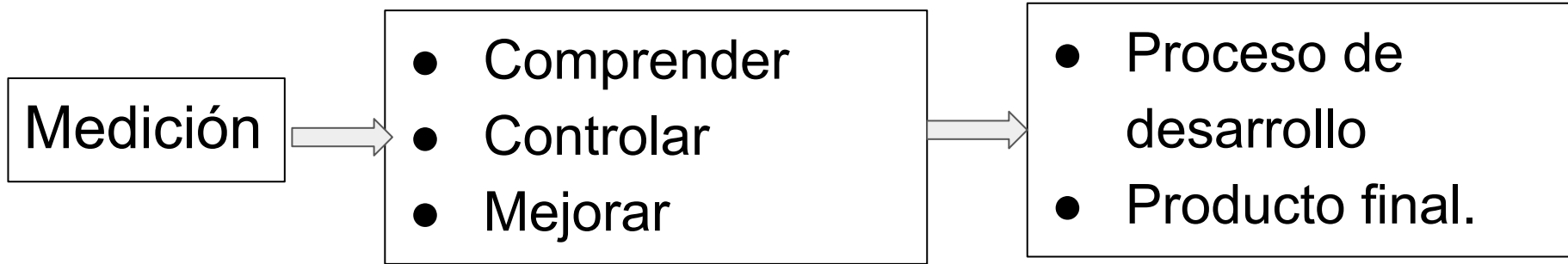


Medición de Sistemas de Información en Ingeniería de Software

Introducción



La teoría de la medición de software

La teoría de la medición de software se basa en los principios generales de la teoría de la medición

- **Medición**
- **Escalas de Medición**
- **Atributos y Entidades**
- **Validez y Confiabilidad**
- **Objetivos de la Medición (Entender - Evaluar - Controlar)**

Estándares y Metodologías de Medición

Guías y marcos de trabajo

- **ISO/IEC 9126 (ahora parte de ISO/IEC 25000):** (funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad).
- **ISO/IEC 15939:** Especifica un proceso para la medición del software.
- **CMMI (Capability Maturity Model Integration):** Modelo de mejora de procesos.
- **Goal-Question-Metric (GQM):** Metodología para definir un marco de medición basado en los objetivos del proyecto o la organización.
- **Practical Software and Systems Measurement (PSM):** Programa de medición alineado con las necesidades del negocio y los objetivos del proyecto.

Métricas de Software

Medidas cuantitativas de diversos aspectos del software, el proceso de desarrollo o los recursos utilizado

- **Métricas de Producto:** Describen las características del software en sí mismo:
 - **Tamaño:** Líneas de Código (LOC), Puntos de Función (FP), Puntos de Característica.
 - **Complejidad:** Complejidad Ciclomática, Profundidad de Anidamiento.
 - **Acoplamiento:** Medidas de dependencia entre módulos.
 - **Cohesión:** Medidas de la relación entre los elementos dentro de un módulo.
 - **Calidad:** Número de defectos, severidad de los defectos, tiempo medio entre fallos (MTBF).

Métricas de Software

Medidas cuantitativas de diversos aspectos del software, el proceso de desarrollo o los recursos utilizado

- **Métricas de Proceso:** Describen las características del proceso de desarrollo:
 - **Esfuerzo:** Horas-hombre, costo.
 - **Tiempo:** Duración de las fases, tiempo de ciclo.
 - **Productividad:** Puntos de Función por persona-mes, Líneas de Código por persona-día.
 - **Eficiencia:** Relación entre el esfuerzo invertido y el resultado obtenido.

Métricas de Software

Medidas cuantitativas de diversos aspectos del software, el proceso de desarrollo o los recursos utilizado

- **Métricas de Recurso:** Describen las características de los recursos utilizados en el desarrollo:
 - **Personal:** Experiencia del equipo, rotación de personal.
 - **Herramientas:** Utilización de herramientas, costo de las herramientas.

Métricas de Software

Medidas cuantitativas de diversos aspectos del software, el proceso de desarrollo o los recursos utilizado

- **Métricas de Recurso:** Describen las características de los recursos utilizados en el desarrollo:
 - **Personal:** Experiencia del equipo, rotación de personal.
 - **Herramientas:** Utilización de herramientas, costo de las herramientas.

Métricas adecuadas = función (objetivos de la medición, contexto del proyecto)

Métricas de Software

Se integran en cada etapa del ciclo de desarrollo

- **Herramientas de Análisis Estático de Código:** Analizan el código fuente sin ejecutarlo. Verifican problemas, complejidad, tamaño, acoplamiento y cohesión, estándares de codificación.
- **Herramientas de Análisis Dinámico:** Analizan el comportamiento del software durante la ejecución.
- **Herramientas de Gestión de Proyectos:** Permiten seguimiento de tiempo, esfuerzo, costo y progreso de las tareas.
- **Herramientas de Gestión de Defectos:** Registran, rastrean y analizan los defectos (Calidad).
- **Herramientas de Business Intelligence y Reporting:** Permiten consolidar datos de diferentes fuentes (Interpretación de Métricas).

Medición y Calidad de Modelos

Los modelos son representaciones abstractas del sistema y su calidad impacta directamente en la calidad del software final.

- **Complejidad del Modelo:** Complejidad estructural del modelo (número de elementos, relaciones y niveles de anidamiento)
- **Consistencia:** Evalúa Inconsistencias.
- **Compleitud:** Modelo representa todos los requisitos.
- **Claridad y Legibilidad:** Subjetivas (Entendible por interesados).
- **Acoplamiento y Cohesión en Modelos:** Dependencias entre diferentes partes del modelo y la relación entre los elementos dentro de una misma parte.

Teoría de la Medición

Cuantifica y analiza aspectos del software, los procesos de desarrollo y los recursos involucrados

Medición: Proceso de asignar números o símbolos a atributos de entidades del mundo real, siguiendo reglas definidas.

Entidades

- **Productos de Software:** El código fuente, la documentación, los modelos de diseño, las interfaces de usuario.
- **Procesos de Desarrollo:** Las actividades de análisis de requisitos, diseño, codificación, pruebas, despliegue.
- **Recursos:** El tiempo dedicado por los desarrolladores, el costo del proyecto, el número de personas en el equipo.

Atributos: son las características o propiedades de estas entidades que nos interesa medir.

Ejemplo:

- **Entidad:** Módulo código fuente.
- **Atributo:** Complejidad estructural.
- **Medición:** Uso métrica de Complejidad Ciclomática.
- **Regla definida:** algoritmo específico para calcular la Complejidad Ciclomática.

Teoría de la Medición

Escalas de Medición: El tipo de escala de medición determina qué operaciones estadísticas son válidas para los datos recolectados.

- **Escala Nominal:** Se utilizan números o símbolos para clasificar o identificar elementos.
- No hay un orden inherente.
 - **Ejemplo:** Categorizar
 - 1 = Defecto de lógica
 - 2 = Defecto de interfaz
 - 3 = Defecto de rendimiento
 - 4 = Defecto de usabilidad.
 - Solo son categorías distintas.

- **Escala Ordinal:** Números o símbolos representan categorías con un orden o jerarquía.
- Las diferencias entre los valores no son necesariamente iguales ni significativas.
 - **Ejemplo:** Severidad de un defecto.
 - 1 = Bajo,
 - 2 = Medio,
 - 3 = Alto,
 - 4 = Crítico.

Teoría de la Medición

Escala de Medición: El tipo de escala de medición determina qué operaciones estadísticas son válidas para los datos recolectados.

- **Escala de Intervalo:** Números representan valores con intervalos iguales entre ellos
- No existe cero absoluto significativo.
- Las operaciones de suma y resta son válidas, pero la multiplicación y la división no lo son de manera significativa.

- **Ejemplo:** Índice de "facilidad de mantenimiento"
- Escala: 0 a 100.

- **Escala de Razón:** Números representan valores con intervalos iguales
- Existe cero absoluto significativo.

- **Ejemplo:** Medir el tiempo dedicado a corregir un defecto (ej: minutos).
- 0: No se trabajó
- 15: Tiempo dedicado.
- 30: Doble tiempo anterior

Teoría de la Medición

Atributos y Entidades: Definir con precisión qué se va a medir (el atributo) y a qué objeto pertenece ese atributo (la entidad).

Ejemplo:

- **Atributo:** Acoplamiento entre módulos.
- **Entidad:** El sistema de gestión de Inventarios
- **Medición:** Calcular el número de dependencias que cada módulo tiene con otros módulos dentro del sistema de gestión de Inventarios.

Teoría de la Medición

Validez y Confiabilidad: Determinan Calidad de una medición

- **Validez:** Medida es válida si realmente mide el atributo que se pretende medir.
 - **Ejemplo 1:** Medir la "complejidad de mantenimiento" de un sistema.
 - Número de líneas de código (LOC) No es del todo válida
 - Combinar Complejidad Ciclomática con medidas de acoplamiento y cohesión.

- **Confiabilidad:** Medida es confiable si es consistente y reproducible bajo las mismas condiciones.
 - **Ejemplo 1:** Definir un procedimiento claro y objetivo para medir
 - Por ejemplo la eficiencia de trabajo de dos desarrolladores.

Teoría de la Medición

Objetivos de la Medición: Definir propósito de la medición asegura que los datos recolectados sean relevantes y útiles.

- **Entender:** Comprender software, proceso o recursos.

- **Ejemplo 1:** Cómo se utiliza el tiempo del equipo de trabajo?
- Medir la distribución de tiempo del equipo entre las diferentes actividades que llevan a cabo.

- **Evaluar:** Se cumplen criterios, estándares o metas?

- **Ejemplo 1:** Criterios de cobertura de pruebas.
- Medir la cobertura de pruebas de un sistema después de la fase de pruebas.

Teoría de la Medición

Objetivos de la Medición: Definir propósito de la medición asegura que los datos recolectados sean relevantes y útiles.

- **Controlar:** Progreso y tomar acciones correctivas si es necesario.

- **Ejemplo 1:** Monitorear calidad del software
- Medir el número de defectos.
- Tomar medidas correctivas como revisar el proceso de desarrollo, asignar más recursos a las pruebas.

- **Predecir:** Estimar valores futuros basándose en datos históricos.

- **Ejemplo 1:** Predecir esfuerzo de desarrollo
- Utilizar datos históricos (tamaño del proyecto, experiencia del equipo, complejidad).

Teoría de la Medición

Objetivos de la Medición: Definir propósito de la medición asegura que los datos recolectados sean relevantes y útiles.

- **Mejorar:** Identificar áreas de oportunidad para optimizar procesos y productos.
 - **Ejemplo 1:** Identificar cuello de botella.
 - Medir el tiempo de ciclo para la resolución de defectos
 - Acción: automatizar pruebas.

Métricas de Software

Objetivo: Se utilizan para evaluar diversas características del software, el proceso de desarrollo y los recursos utilizados

Métricas de Producto

- **Tamaño:**
 - **Líneas de Código (LOC):** Cuenta el número de líneas de código fuente en un módulo o sistema.
 - **Puntos de Función (PF):** Mide la funcionalidad proporcionada por el software desde la perspectiva del usuario.
 - **Puntos de Historia:** Estima el esfuerzo necesario para implementar una historia de usuario.

Métricas de Producto

- **Complejidad:**
 - **Complejidad Ciclomática (CC):** Mide la complejidad del flujo de control de un programa.
 - **Profundidad de Anidamiento:** Mide la profundidad de las estructuras de control anidadas (bucles, condicionales).
- **Acoplamiento:** Mide el grado de dependencia entre módulos de software.
- **Cohesión:** Mide el grado en que los elementos dentro de un módulo están relacionados.

Métricas de Software

Objetivo: Se utilizan para evaluar diversas características del software, el proceso de desarrollo y los recursos utilizados

Métricas de Producto

- **Calidad:**
 - **Número de Defectos:** Cuenta el número de defectos encontrados en el software.
 - **Densidad de Defectos:** Número de defectos por unidad de tamaño (por ejemplo, defectos por KLOC).
 - **Tiempo Medio Entre Fallos (MTBF):** Mide el tiempo promedio entre fallos del software.
 - **Cobertura de Pruebas:** Mide el grado en que el código ha sido ejercitado por las pruebas.

Métricas de Software

Objetivo: Se utilizan para evaluar diversas características del software, el proceso de desarrollo y los recursos utilizados

Métricas de Proceso

- **Esfuerzo:**
 - **Horas-hombre:** Cantidad de trabajo realizado por el equipo.
 - **Costo:** Costo total del proyecto, incluyendo salarios, infraestructura y otros gastos.
- **Tiempo:**
 - **Duración del Ciclo:** Tiempo necesario para completar una iteración o fase del desarrollo.
 - **Tiempo de Comercialización:** Tiempo necesario para lanzar un producto al mercado.

Métricas de Proceso

- **Productividad:**
 - **Líneas de Código por Persona-Mes:** Cantidad de código producido por un desarrollador en un mes.
 - **Puntos de Función por Persona-Mes:** Cantidad de funcionalidad entregada por un desarrollador en un mes.
- **Eficiencia:**
 - **Retorno de la Inversión (ROI):** Beneficio obtenido en relación con el costo del proyecto.
 - **Índice de Velocidad:** Cantidad de trabajo completado por un equipo ágil en un sprint.

Métricas de Software

Objetivo: Se utilizan para evaluar diversas características del software, el proceso de desarrollo y los recursos utilizados

Métricas de Recursos

- **Personal:**
 - **Tamaño del Equipo:** Número de desarrolladores, testers y otros miembros del equipo.
 - **Experiencia del Equipo:** Años de experiencia promedio de los miembros del equipo.
 - **Rotación de Personal:** Tasa a la que los empleados abandonan la organización.

Métricas de Recursos

- **Herramientas:**
 - **Utilización de Herramientas:** Grado en que se utilizan las herramientas de desarrollo y prueba.
 - **Costo de las Herramientas:** Inversión en licencias de software, hardware y otras herramientas.

Métricas de Software

Objetivo: Se utilizan para evaluar diversas características del software, el proceso de desarrollo y los recursos utilizados

Uso de las Métricas de Software

Las métricas de software se utilizan para:

- **Planificación y Estimación:** Estimar el tamaño del proyecto, el esfuerzo, el costo y el cronograma.
- **Control y Seguimiento:** Monitorear el progreso del proyecto, identificar desviaciones del plan y tomar acciones correctivas.
- **Evaluación de la Calidad:** Medir la calidad del software, identificar áreas de mejora y asegurar el cumplimiento de los estándares.
- **Mejora de Procesos:** Identificar cuellos de botella, optimizar el flujo de trabajo y mejorar la eficiencia del desarrollo.
- **Toma de Decisiones:** Proporcionar datos objetivos para respaldar las decisiones de gestión.

Herramientas de Medición de Software

Objetivo: automatizan la recopilación, el análisis y la presentación de métricas

Tipos de Herramientas de Medición de Software

- **Herramientas de Análisis Estático de Código**
- **Herramientas de Análisis Dinámico**
- **Herramientas de Gestión de Proyectos**
- **Herramientas de Gestión de Defectos**
- **Herramientas de Business Intelligence y Reporting**

Herramientas de Medición de Software

Selección de herramientas

Factores de selección

- **Los objetivos de la medición:** ¿Qué aspectos del software y del proceso de desarrollo se van a medir?
- **El tipo de proyecto:** ¿Es un proyecto pequeño o grande? ¿Es un proyecto ágil o en cascada?
- **El presupuesto:** ¿Cuánto se puede invertir en herramientas?
- **La infraestructura existente:** ¿Qué herramientas ya están en uso en la organización?
- **Las habilidades del equipo:** ¿El equipo tiene la experiencia necesaria para utilizar las herramientas?

IMPORTANTE:

Herramientas deben integrarse con el proceso de desarrollo existente.

Deben proporcionar los datos necesarios para alcanzar los objetivos de la medición.