4. Métricas de software

Algunas definiciones: medidas, mediciones y métricas

- Una **medida** provee una indicación **cuantitativa** del grado, cantidad, dimensión, capacidad o tamaño de alguno de los atributos del producto o proceso de Software
- Una **medición** es la actividad (o acto) de determinar una medida
- Una **métrica** es una medida cuantitativa del grado en el cual un producto o proceso posee un determinado atributo
- Ejemplo:
  - **Medida**: cantidad de defectos encontrados durante la fase de pruebas
  - **Medición**: la forma en la cual se registran esos defectos
  - **Métrica**: el número promedio de defectos encontrados por hora en la fase de pruebas

Métricas de software

- ¿Para qué sirven? ¿Cómo pueden ser usadas?
  - Pueden ser usadas para responder:
    - ¿Cuál es el tamaño del programa?
    - ¿Cuál es el costo y duración estimada del proceso de desarrollo?
    - ¿El requerimiento es verificable?
    - ¿Cuándo es el momento correcto para detener las pruebas?
    - ¿Cuál es el esfuerzo invertido en la fase de mantenimiento?
    - ¿Cuántos defectos han sido corregidos vs. reportados en la fase de mantenimiento?
    - ¿Cuál es la complejidad de un módulo?
    - ¿Cuál es el costo estimado de corregir un defecto determinado?
    - ¿Cuál es la productividad de las personas que trabajan en el proyecto?
    - ¿Qué técnica o proceso es más efectivo que otro?

Métricas de software (cont.)

- Métricas de software (Goodman 1993): “La aplicación continua de técnicas basadas en la medición sobre los productos y procesos de software, para proveer información oportuna y útil para mejorar dichos procesos y productos”

“No puedes controlar lo que no mides”
(DeMarco 1982)

Aplicación de métricas de software

- Pueden ser usadas en varios dominios de aplicación
- Algunos ejemplos:
  - *Estimación de costo*
  - *Estimación de esfuerzo*

  Los modelos de estimación se derivan utilizando datos históricos
  - Efectividad (procesos, actividades, técnicas)
  - Tamaño, acoplamiento, cohesión, herencia
  - Predicción de atributos de calidad del software
    - Fiabilidad
    - Usabilidad
    - Eficiencia
    - Facilidad de mantenimiento
  - Efectividad de un conjunto de casos de prueba (distintos tipos de cubrimiento, # defectos detectados, etc)
  - Métricas de gestión (calidad del producto, eficiencia de procesos, productividad)

Características deseables de las métricas de software

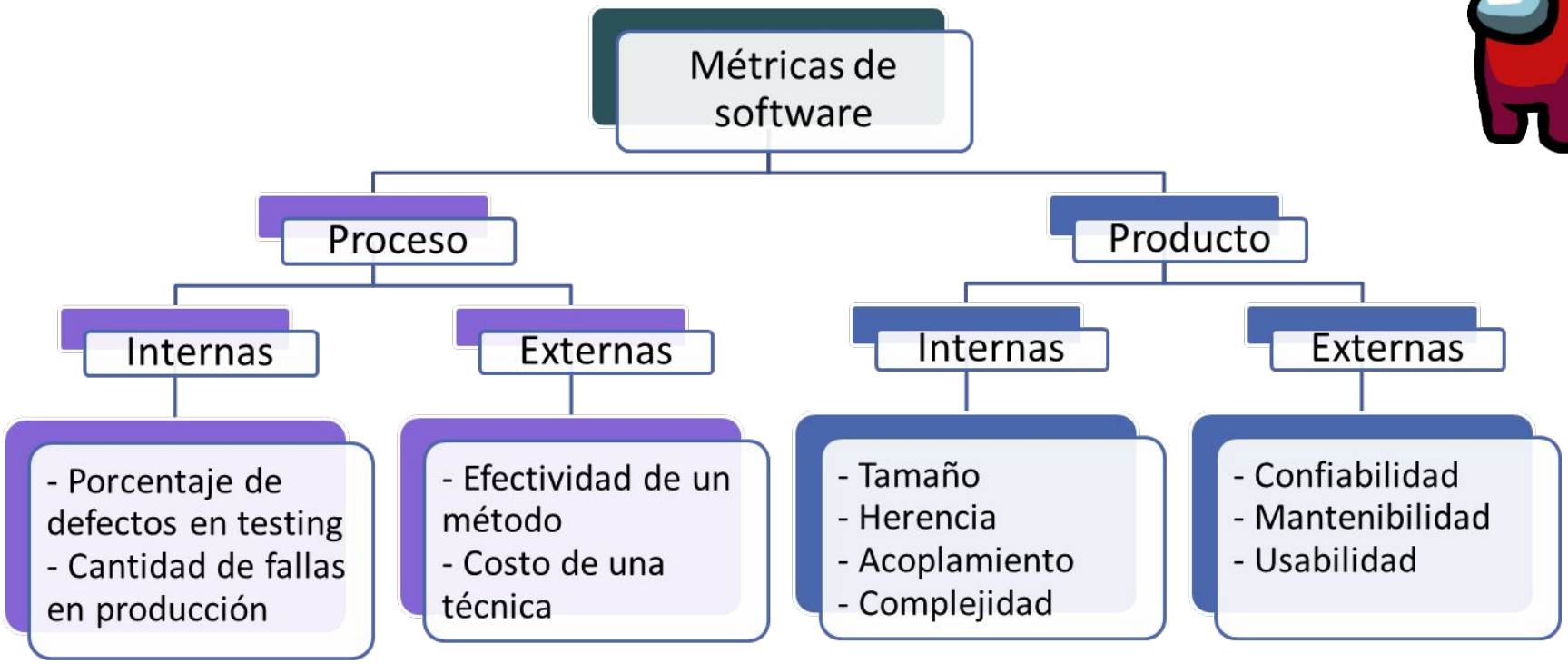
- Una métrica es relevante únicamente si es fácil de entender, de calcular, válida y económica
  - **Cuantitativa**: las métricas deben poder expresarse en valores
  - **Entendible**: la forma en la cual se mide la métrica debe ser fácil de comprender
  - **Validable**: debe poder capturar el mismo atributo para el cual fue diseñada
  - **Económica**: debe ser económico poder medir la métrica
  - **Repetible**: los valores deberían ser los mismos si se repite la medición, esto es, puede ser repetida consistentemente
  - **Comparable**: la métrica debería poder correlacionarse con otra métrica, capturando la misma funcionalidad o concepto

Características deseables de las métricas de software (cont.)

- Deben preservar la correspondencia de las relaciones empíricas con las relaciones numéricas para las entidades de la vida real
- Ejemplo:
  - Relación empírica: “Más alto que”
  - Relación numérica: “>”

Pasos en la medición de software

Categorías de entidades/atributos en IS



Escalas de Medida

- **Escala Nominal**: es la menos poderosa de las escalas, mapea el atributo de la entidad en un nombre o símbolo
  - El mapeo puede verse como una clasificación de las entidades acorde al atributo
  - Ejemplos: clasificaciones, etiquetados, entre otras
- **Escala Ordinal**: categoriza las entidades según un criterio de ordenación
  - Es más poderosa que la escala nominal
  - Ejemplos de criterios de ordenación son “mayor que”, “mejor que” y “más complejo”. Ejemplos: grados, complejidad del software, etc.
- **Escala de intervalo**: se utiliza cuando la diferencia entre dos medidas es significativa
  - Ordena los valores de la misma forma que la escala ordinal, pero existe la noción de “distancia relativa” entre dos entidades.
  - Es más poderosa que la ordinal
  - Ejemplos de escala de intervalo son la temperatura medida en Celsius o Fahrenheit
- **Escala ratio (cociente de dos números)**: si existe un valor cero significativo que indica ausencia de escala y la división entre dos medidas es significativa, se puede utilizar una escala ratio
  - Ejemplos de escala ratio son distancia, temperatura medida en Kelvin, etc.

Nota: el libro menciona otra escala adicional, la absoluta para conteos simples (la contemplamos dentro de la de ratio)

Escalas de medida

- Los datos pueden categorizarse en dos tipos:
  - Categóricos (*Nonmetric scale*): escala ***nominal*** y ***ordinal***
  - Continuos (*Metric scale*): escala de ***intervalo*** y ***ratio***

Ejercicio:

Considere la cantidad de defectos detectados durante una inspección

1. ¿Cuál es la escala de medición?
2. En caso de que la escala de medición de cantidad de defectos fuera clasificar entre 1 y 5 en donde, 1 = muy alta, 2 = alta, 3 = medio, 4 = bajo y 5 = muy bajo ¿cuál sería la escala de medición?

Métricas de calidad de software basadas en defectos

- Mantener la calidad del software es una parte esencial del desarrollo del mismo y debe ser medida en todas las fases del proceso de desarrollo
- Las métricas basadas en defectos pueden utilizarse para medir aspectos tanto del producto como del proceso
- **Densidad de defectos**: $\frac{Cantidad\ de\ defectos}{KLOC}$
  - ¿Información sobre qué nos podría dar esta métrica?
    - Calidad del diseño/desarrollo
    - Efectividad de una técnica de verificación

Métricas de calidad de software basadas en defectos (cont.)

- **Densidad de defectos por fase**: es una extensión de la métrica anterior, la cual se calcula en varias fases del proceso del ciclo de vida del software.
  - Esta métrica provee información acerca de los procedimientos y estándares usados en cada fase del desarrollo de software
- **Efectividad en la remoción de defectos**:

$$\frac{\text{\# defectos removidos en X fase}}{\text{\# defectos latentes}}$$

  - La cantidad de defectos latentes no es conocida, por lo que se toma la cantidad de defectos detectados en fases posteriores
- Las métricas presentadas son **algunas** de las que más se usan, pero hay muchas más (e incluso cada empresa podría crear las suyas en base a sus necesidades)

Métricas de usabilidad

- Involucra: facilidad de uso, amigabilidad, facilidad de aprendizaje, satisfacción del usuario, entre otras
- Propuesta de Bevan (1995) para medir atributos de usabilidad:
  - Efectividad de las tareas = $\frac{1}{100}(cantidad \times calidad)\%$
    - Cantidad = tareas completadas por el usuario (%)
    - Calidad = grado en el cual se satisface el objetivo del usuario para una determinada tarea con el resultado de la misma (%)
  - **Eficiencia temporal** = $\frac{Efectividad}{Tiempo\ de\ la\ tarea}$
  - **Productividad de un período** = $\frac{Tiempo\ de\ la\ tarea - Tiempo\ improductivo}{Tiempo\ de\ la\ tarea} \times 100$
  - **Eficiencia de usuario relativa** = $\frac{Eficiencia\ del\ usuario}{Eficiencia\ de\ un\ experto} \times 100$

Métricas de usabilidad (cont.)

- Otras medidas que pueden usarse:
  - Tiempo de aprendizaje de un sistema
  - Incremento de la productividad a través del uso del sistema
  - Tiempo de respuesta

- ¿De qué otras formas se les ocurre se podría medir la usabilidad?

Métricas de usabilidad (cont.)

- Otras medidas que pueden usarse:
  - Tiempo de aprendizaje de un sistema
  - Incremento de la productividad a través del uso del sistema
  - Tiempo de respuesta

- ¿De qué otras formas se les ocurre se podría medir la usabilidad?
  - Encuesta conteniendo preguntas que indiquen el nivel de satisfacción del usuario respecto del uso del sistema
    - ¿Qué tan fácil es encontrar los temas sobre los que necesita asistencia en el menú de "Ayuda"?
    - ¿Los títulos y descripciones de las pantallas son fácilmente entendibles?
    - ¿Le resulta intuitivo navegar entre las distintas pantallas del sistema?
    - ¿La organización de los tópicos y funcionalidades en el sistema es acorde? ¿Le ayuda a encontrar las funcionalidades que necesita fácilmente?

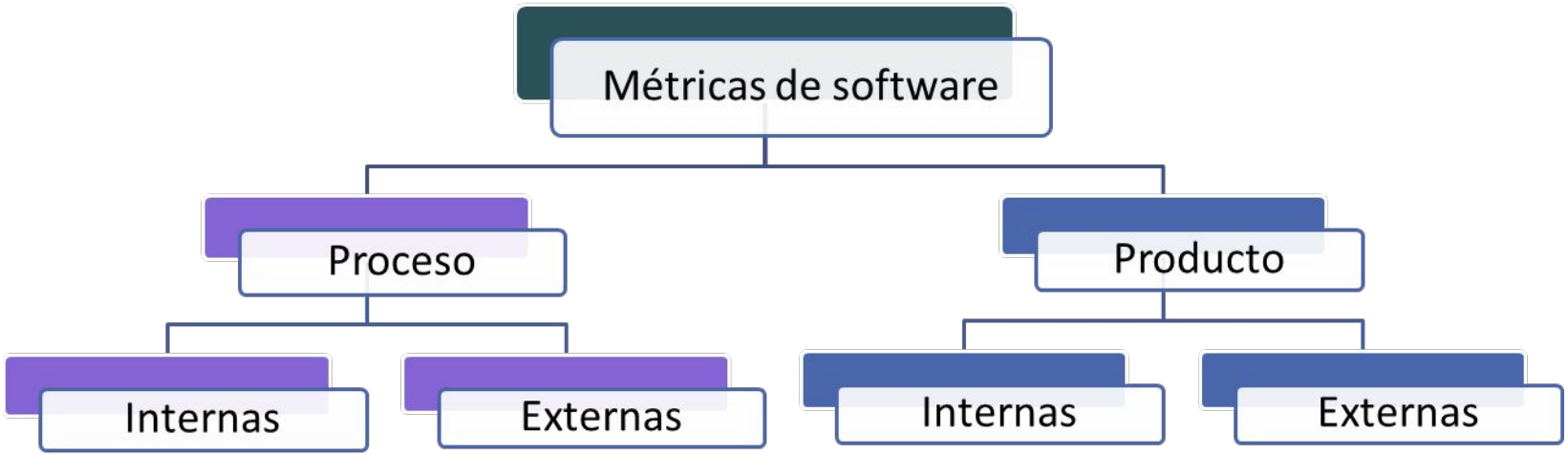
Métricas de testing

- Son usadas para medir el progreso y nivel de pruebas realizadas en el software
- La “cantidad” de testing realizado se mide a través de métricas de cubrimiento (en porcentaje)
  - Cubrimiento de sentencias
  - Cubrimiento de decisión (*branch coverage*)
  - Cubrimiento de operaciones
  - Cubrimiento de condición
  - Cubrimiento de caminos (en un grafo de control)
  - Cubrimiento de ciclos
  - Cubrimiento de condición múltiple

Métricas de testing (cont.)

- **NASA metrics**:
  - Test Focus (TF): razón (cociente) del esfuerzo invertido en encontrar y remover defectos sobre la cantidad total de defectos reportados
  - $TF = \frac{\#\ Defectos\ corregidos\ y\ cerrados}{Total\ de\ defectos\ reportados}$
  - Cubrimiento de defectos $\mathrm{FCM} = \frac{\#\ Defectos\ abordados \times severidad\ de\ los\ defectos}{Total\ defectos \times severidad\ de\ los\ defectos}$
- Métricas de proceso:
  - # casos de prueba (CP) diseñados/ejecutados/exitosos/fallidos
  - Tiempo de ejecución de un CP
  - Tiempo total de ejecución
  - Tiempo de desarrollo de un CP
  - Esfuerzo relacionado al testing
  - Tiempo total de desarrollo de CP

Pregunta: ¿Qué tipos de métricas hemos visto hasta ahora?



Métricas de tamaño

- **Propósito principal**: medir el tamaño del software para ser tomado como entrada de modelos empíricos de estimación de costo y esfuerzo
- Medida más popular: LOC's (*Lines of Code*)
  - Ejecutables
  - No ejecutables (comentarios, líneas en blanco)
- En general se cuentan las líneas de código ejecutable, ya que tomar en cuenta las no ejecutables nos puede dar una falsa medida de alta productividad

Métricas evolutivas

- Los sistemas de software pueden ser analizados a través de métricas de **cambio** o de **evolución** del sistema
  - Proveen información útil para entender la evolución y el historial de liberaciones de un sistema
- Métricas de revisión, *refactoring* y *bug-fixing*
  - **Revisiones**: cantidad de revisiones de un archivo en el repositorio de software
  - **Refactorings**: cantidad de veces que ha cambiado un archivo
  - **Bug-fixes**: cantidad de veces que un archivo ha sido asociado a una corrección de un bug
  - **Autores**: cantidad de diferentes autores que han realizado commit o check en el repositorio de software
- Basadas en código:
  - LOC agregadas, Max LOC agregadas, Avg LOC agregadas
  - LOC borradas, Max LOC borradas, Avg LOC borradas

Métricas de uso del software

Relevancia Práctica y uso de métricas de software en Investigación

- Investigación
  - Desarrollar modelos de predicción de comportamiento en base a ciertas métricas (variables independientes) las cuales denoten cierta correlación sobre otras variables
  - Conocer los valores de umbral (pico) de algunas métricas ayuda a controlar los procesos y productos, y enfocar el esfuerzo sobre aquellos que superen un nivel aceptable de riesgo
- Industria
  - Los diseñadores de software pueden usar métricas para obtener parámetros de calidad (*benchmarks*) para validar y comparar varios productos de software
  - Jefes de proyecto pueden usar métricas de software para controlar y auditar la calidad del software durante el ciclo de vida del desarrollo

Relevancia Práctica y uso de métricas de software en Investigación

- Existen muchas propuestas sobre cómo medir características del proceso/producto de software
  - Algunas propuestas son reconocidas como estándares
  - Otras propuestas están en proceso de validación/estudio
- Es importante definir correctamente cómo se mide un determinado atributo: la escala y el proceso de medición
  - Permite que los resultados de diferentes estudios puedan ser comparables
  - Reduce amenazas a la validez dentro del estudio de investigación
  - Ayuda a la replicación del estudio