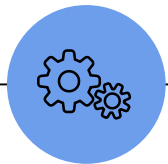


Introducción a la ingeniería de software

Proyectos de software – Clase 2





En la clase anterior...

Proyectos de sw

Definiciones.
Restricciones.
Descomposición del trabajo

Organización

Perfiles.
Roles.
Líder.
Organización del equipo.

Riesgos

Estimaciones
Plan RSGR - MMMR

Planificación

Dependencias.
Diagramas de red.
Camino crítico

Estimaciones

Técnicas.



Tarea: Duración vs Esfuerzo

- Duración:
 - Unidad de tiempo (horas, días, meses)
 - Generalmente lo que se le comunica al cliente
 - Visible externamente
- Esfuerzo
 - Unidad de tiempo / recursos usados (horas hombre)
 - Depende de la dificultad o complejidad de la tarea
 - Asignar más recursos disminuye la duración, pero no proporcionalmente. ¿Por qué?
 - No necesariamente es visible externamente
- La duración se calcula en base al esfuerzo
 - Se debe contemplar:
 - Factor de eficiencia
 - Experticia del recurso
 - Eventos no esperados o equivocaciones



Tarea: Duración vs Esfuerzo

- ⦿ Una tarea lleva 10 días hombre de esfuerzo
- ⦿ Es decir, si se le asigna 1 persona al 100% de su tiempo, entonces tendrá una duración de 10 días

- ⦿ si se le asignan 2 personas con el 100% de dedicación $10/2 = 5$ días de duración

- ⦿ si se le asigna 1 persona con el 50% de dedicación, entonces $10/0,5 = 20$, serán 20 días de duración



Temas de hoy



Métricas.

Métricas de producto.

Métricas de proceso, de proyecto y software

1

Mediciones, métricas, medidas

Introducción. Teoría de la medición.
Métricas para software

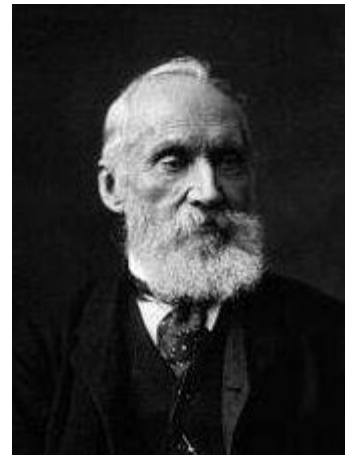


¿POR QUÉ medir?



Cuando puedes medir aquello de lo que hablas y expresarlo en números, sabes algo acerca de ello; pero cuando no puedes medir , cuando no puedes expresarlo en números, tu conocimiento es exiguo e insatisfactorio: puede ser el comienzo del conocimiento, sin embargo apenas habrás avanzado en tus pensamientos, hacia la etapa de una ciencia.

Lord Kelvin



“



Objetivos

○ Se desea:

- **comprender** lo que se está realizando en el proceso de desarrollo
- hacer más **visibles** determinados aspectos del proceso y del producto
- conocer la **complejidad** del producto y del proceso
- realizar **estimaciones** de costos y de esfuerzo
- **controlar** los procesos y los proyectos
- incorporar **mejoras** en los procesos y en los productos



Mediciones, medidas, métricas

⦿ ¿Cuál es la diferencia?



Mediciones, medidas, métricas

MEDICIÓN

Proceso mediante el cual se asignan números o símbolos a los atributos de entidades del mundo real, de manera tal que se los describa de acuerdo a reglas definidas.

- ⦿ **Entidad:** objeto o evento del mundo real
- ⦿ **Atributo:** característica o propiedad de una entidad



Mediciones, medidas, métricas

○ Cuantificación **directa** de los atributos: **medición**

○ Ej: altura de un árbol, duración de un evento.

○ Cuantificación **indirecta**, combinación de más de una medición en un elemento que refleja el atributo que intentamos comprender: **cálculo**

○ Ej: valuación de una casa, sensación térmica.

Mediciones, medidas, métricas

*una **MEDIDA**
proporciona un indicio cuantitativo de la extensión,
cantidad, dimensión, capacidad o tamaño de algún
atributo de un producto o proceso.*

- La medición es el acto de determinar una medida



Mediciones, medidas, métricas

MÉTRICA

Medida cuantitativa del grado en el que un sistema, componente o proceso posee un atributo determinado.

- Métrica: método de medición elegido junto con la escala de medición.



Indicadores

INDICADOR

Métrica o combinación de métricas que proporcionan comprensión acerca del proceso de software, el proyecto de software o el producto en sí.

- Los indicadores proporcionan comprensión para realizar ajustes, para incorporar cambios que permitan hacer mejor las cosas.



Teoría de la representación

- ⦿ Los datos obtenidos mediante mediciones deben representar los atributos observados
- ⦿ La manipulación de los datos debe preservar las relaciones observables entre las entidades
 - Medición: mapeo del mundo empírico al mundo formal, o relacional.
 - Medida: el número o símbolo asignado por el mapeo a una entidad para caracterizar un atributo.

Teoría de la representación

● **Condición de representación:** un mapeo M debe mapear entidades a números y relaciones empíricas a relaciones numéricas de manera que las relaciones empíricas preserven y sean preservadas por las relaciones numéricas.



Blancanieves:
1,70m

Dormilón: 1,10m



Escalas

Al mapeo de medición, junto con el sistema empírico y el sistema de relaciones numéricas lo llamamos **escala de medición**.

⦿ De acuerdo a sus características identificamos cinco tipos de escala:

- Nominal
- Ordinal
- De intervalos
- De proporciones (o ratios)
- Absoluta



Escala nominal

- ⦿ Se definen clases o categorías en las que se ubican las entidades de acuerdo al valor del atributo medido.
- ⦿ La escala **nominal** tiene dos características principales:
 - El sistema de relación empírico consiste sólo de diferentes clases, no hay noción de orden entre ellas
 - Cualquier representación simbólica o numérica para las distintas clases es aceptable. No hay noción de magnitud asociada a los números o símbolos

Escala nominal. Ejemplo.

Defecto en automotor. Atributo: ubicación

● M1(x) =

- 100 mecánico
- 200 eléctrico
- 600 chasis





Escala ordinal

- ⦿ Define clases o categorías, al igual que la escala nominal, pero agrega información de ordenamiento de las clases.
- ⦿ Características de la escala **ordinal**:
 - El sistema de relación empírico consiste de diferentes clases ordenadas con respecto al atributo
 - Cualquier mapeo que preserve el ordenamiento es aceptable
 - Los números sólo representan un ranking (no tiene sentido sumar, restar o cualquier operación aritmética)



Escala ordinal. Ejemplo.

Defecto en automotor. Atributo: gravedad

○ $M1(x) =$

- 1 trivial
- 2 moderado
- 3 complejo
- 4 irresoluble

○ $M2(x) =$

- 1 trivial
- 8 moderado
- 12 complejo
- 40 irresoluble





Escala de intervalos

- ⦿ Esta escala captura información sobre el tamaño de los intervalos que separan a cada una de las clases ordenadas
 - Captura el “salto” entre clase y clase
- ⦿ Las características de la escala **de intervalos** son:
 - Preserva el orden
 - Preserva las diferencias (pero no proporciones)
 - Se aceptan operaciones de suma y resta



Escala de intervalos. Ejemplo.

Defecto en automotor. Atributo: gravedad

○ M1(x) =

- 2,5 trivial
- 4,5 moderado
- 6,5 complejo
- 8,5 irresoluble

○ M2(x) =

- 10 trivial
- 20 moderado
- 30 complejo
- 40 irresoluble





Escala de intervalos. Ejemplo 2.

Proyecto. Atributo: duración

⦿ M1(x) = meses desde
1/4/2013 (producción)



⦿ M2(x) = años desde
1/1/2014 (finanzas)

- ⦿ Podemos decir de demorar 10 días más y cuantificarlo en ambas escalas sin problemas
- ⊘ No podemos decir de demorar el proyecto al “doble” y cuantificarlo en ambas escalas



Escala de proporciones

- Brinda información acerca de las diferencias de proporciones existentes entre cada clase.
- Características de la escala **de proporciones**:
 - Preserva el orden, el tamaño de los intervalos y las proporciones de los intervalos entre entidades
 - Existe el elemento cero, representa la ausencia total del atributo
 - El mapeo debe comenzar en cero e incrementarse en intervalos iguales llamados unidades
 - Todas las operaciones aritméticas son válidas



Escala de proporciones. Ejemplo.

Costo monetario de un producto. Atributo: valor

● $M1(x) =$

- Cualquier valor mayor o igual a cero es válido
- \$20 es más caro que \$15
- \$40 es el doble que \$20

● $M2(x) =$ Valor expresado en otra moneda

○ $M2(x) = M1(x) * k$





Escala absoluta

- ⦿ No admite transformaciones excepto la identidad.
- ⦿ Características de la escala **absoluta**:
 - La medición se realiza contando el número de elementos
 - Siempre toma la forma “número de ocurrencias de x en la entidad”
 - Solo hay un mapeo posible de medición: la cuenta
 - Todas las operaciones aritméticas son significativas



Escala absoluta. Ejemplo.

Monedas en un monedero. Atributo: cantidad

● $M1(x) =$

- el número resultante de contar la cantidad de monedas
- no importa el valor que simboliza cada moneda
- No existe otra métrica $M2$ que sea distinta a $M1$





Proceso de medición

- Puede caracterizarse mediante cinco actividades.
- **Formulación.** Derivación de medidas y métricas apropiadas.
- **Recolección.** Mecanismo para acumular los datos requeridos para obtener las métricas formuladas.
- **Análisis.** Cálculo de métricas y aplicación de herramientas matemáticas.
- **Interpretación.** Conocimiento ganado a partir de la representación usada.
- **Retroalimentación.** Recomendaciones derivadas de la interpretación y análisis de los resultados.

2

Métricas del producto

Aspectos medibles del producto



¿QUÉ medir?



- Basadas en funciones
 - Para calidad de la especificación
 - Del diseño arquitectónico
 - Del diseño orientado a objetos
 - De diseño a nivel de componente
 - Orientadas a la operación*
 - De la interfaz del usuario
 - Para el código
 - Para las pruebas
 - Para el mantenimiento
- Req.
- Diseño
- Implem.
- Testeo
- Mant.



Métrica basada en funciones

-Basadas en funciones

- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

- ⦿ La intención es predecir el “tamaño” del sistema que resultará.
- ⦿ El motivo es que el tamaño *puede** ser un indicador:
 - de la complejidad del diseño
 - del esfuerzo de codificación
 - del esfuerzo de integración
 - del esfuerzo necesario para las pruebas



Puntos de función (PF)

-Basadas en funciones

- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

- La métrica de puntos de función se utiliza para predecir la funcionalidad de un sistema.
- Utiliza datos para:
 - Estimar costo o esfuerzo requerido para diseñar, codificar y probar el software
 - Predecir el número de errores que se encontrarán durante las pruebas
 - Prever el número de componentes y/o de líneas de código del sistema a implementar



Para calidad de especificación

-Basadas en funciones

-Para calidad de la especificación

-Del diseño arquitectónico

-Del diseño orientado a objetos

-De la interfaz del usuario

-Para el código

-Para las pruebas

-Para el mantenimiento

● Características de evaluación de la especificación de requerimientos (RQs):

- Especificidad (falta de ambigüedad)
- Completitud
- Corrección
- Facilidad de comprensión
- Verificabilidad
- Consistencia (interna y externa)
- Factibilidad
- Concisión
- Trazabilidad
- Facilidad de modificación
- Precisión
- Reusabilidad



Para calidad de especificación

-Basadas en funciones

-Para calidad de la especificación

-Del diseño arquitectónico

-Del diseño orientado a objetos

-De la interfaz del usuario

-Para el código

-Para las pruebas

-Para el mantenimiento

○ Sugieren representar las características mediante una o más métricas.

○ Ejemplos:

○ Consistencia de la interpretación de los revisores de un requerimiento (Especificidad)

$$E1 = N_{ui} / N_t$$

○ Grado de validación de los requerimientos (Compleitud)

$$E2 = N_c / (N_c + N_{nv})$$



Métricas del diseño arquitectónico

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Orientadas a la operación
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

- Se enfocan en características de la arquitectura del programa.
- Son de “caja negra” porque no requieren conocimiento sobre el funcionamiento interior de los componentes.
- Si la complejidad arquitectónica aumenta, aumenta la complejidad del sistema. Es probable que aumenten:
 - Esfuerzo de integración
 - Esfuerzo de pruebas



Métricas del diseño arquitectónico

Tres medidas de complejidad propuestas: (Card y Glass)

● Complejidad estructural de un módulo i :

○ $S(i) = f_{out}^2(i)$

● Complejidad de datos

○ $D(i) = \frac{v^{(i)}}{(f_{out}(i) + 1)}$

● Complejidad del sistema

○ Es la suma de las complejidades estructurales y de datos

○ $C(i) = S(i) + D(i)$



Del diseño orientado a objetos

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

- Características medibles de un diseño OO
 - Tamaño
 - Complejidad
 - Acoplamiento
 - Suficiencia
 - Completitud
 - Cohesión
 - Primitivismo
 - Similitud
 - Volatilidad



Del diseño orientado a objetos

Tres grupos de métricas orientadas a objetos

- ⦿ Métricas orientadas a la clase (CK)
- ⦿ Métricas orientadas a la clase (MOOD)
- ⦿ Métricas de Lorenz y Kidd



Métricas CK

- Suite de métricas de Chidamber y Kemerer
 1. Métodos ponderados por clase (MPC)
 2. Profundidad del árbol de herencia (PAH)
 3. Número de hijos (NDH)
 4. Acoplamiento entre clases de objetos (ACO)
 5. Respuesta para una clase (RPC)
 6. Falta de cohesión en métodos (FCOM)



Métricas CK: MPC

- Métodos ponderados por clase
 - Clase con n métodos, de complejidad c_1, \dots, c_n
 - La complejidad se mide con alguna métrica elegida
 - $MPC = \sum_{i=1}^n c_i$
- Más métodos, más complejos:
 - Más esfuerzo de implementación y pruebas
 - Más complejo el árbol de herencia
 - Más específica es la aplicación
 - Más limitada la reutilización



De la interfaz del usuario

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

- Sears propone “Correctitud de la distribución de objetos visuales”
 - Posición de las entidades (gráficos, íconos, textos, menues, ventanas, etc.) en la distribución de la pantalla
 - Frecuencia con la que se usa
 - La dificultad para moverse de una entidad a otra



Para el código

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

Medidas de Halstead

- n_1 = nro de *operadores* distintos que aparecen en un programa
- n_2 = nro de *operandos* distintos que aparecen en un programa
- N_1 = nro total de ocurrencias del operador
- N_2 = nro total de ocurrencias del programa

Usa las medidas para:

- expresiones de longitud del programa, $N = n_1 \log_2(n_1) + n_2 \log_2(n_2)$
- volúmen de información $V = N \log_2(n_1 + n_2)$
- nivel del programa, nivel del lenguaje, etc.



Para las pruebas

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas**
- Para el mantenimiento

- La mayor parte se enfoca en el proceso de las pruebas, no en las características técnicas de las pruebas en sí.
 - Métricas de Halstead
 - Estimar esfuerzo de prueba mediante métricas derivadas de las de Halstead
 - Métricas para pruebas orientadas a objetos
 - Consideran aspectos de encapsulación y herencia



Para el mantenimiento

- Basadas en funciones
- Para calidad de la especificación
- Del diseño arquitectónico
- Del diseño orientado a objetos
- De la interfaz del usuario
- Para el código
- Para las pruebas
- Para el mantenimiento

● Pueden usarse todas las ya definidas, se proponen métricas nuevas, diseñadas explícitamente para actividades de mantenimiento

● Índice de madurez de software

○ M_T = nro de módulos en la versión actual

○ F_c = nro de módulos con cambios en la versión actual

○ F_a = nro de módulos que se agregaron

○ F_d = nro de módulos de la versión anterior borrados en la actual

○ $IMS = \frac{M_T - (F_c + F_a + F_d)}{M_T}$

3

Métricas de proceso, proyecto y software

Aspectos medibles del proceso y del proyecto



¿QUÉ medir?





Métricas del proceso y del proyecto

- ⦿ El objetivo es proporcionar indicadores para mejorar el proceso a largo plazo
- ⦿ Se recopilan durante mucho tiempo
- ⦿ Las métricas del proyecto permiten:
 - Valorar el estado de un proyecto en marcha
 - Rastrear riesgos potenciales
 - Descubrir áreas problemáticas antes que se vuelvan críticas
 - Ajustar el flujo de trabajo o las tareas
 - Evaluar la habilidad del equipo del proyecto para controlar la calidad de los productos



Métricas del proceso

- ⦿ Distintos usos para las métricas: privados y públicos
- ⦿ Usos privados
 - Tasa de defecto por individuo
 - Tasa de defecto por componente
 - Errores encontrados durante el desarrollo
- ⦿ Las métricas privadas deben manejarse con cuidado: utilizarlas como motor para la mejora



Métricas del proceso

- Usos públicos

- Tasas de defectos, esfuerzo, tiempos calendario, y cualquier otro dato relacionados

- Utilizarlas para mejorar el desempeño del proceso



Métricas del proyecto

- Son útiles durante la estimación.
- Resultados anteriores se usan como base de las estimaciones actuales.
 - Minimizar la planificación de desarrollo: ajustes para evitar demoras, mitigar potenciales problemas (riesgos)
 - Valorar la calidad del producto en marcha
 - Modificar el enfoque técnico para mejorar la calidad



Métricas del software

- Basarse en LOC (líneas de código)
- Errores por KLOC
- Defectos por KLOC
- \$ por KLOC
- Páginas de documentación por KLOC
- Errores por persona-mes
- KLOC por persona-mes
- \$ por página de documentación

Project	LOC	Effort	\$(000)	Pp. doc.	Errors	Defects	People
alpha	12,100	24	168	365	134	29	3
beta	27,200	62	440	1224	321	86	5
gamma	20,200	43	314	1050	256	64	6
•	•	•	•	•	•		
•	•	•	•	•	•		
•	•	•	•	•	•		



Métricas del software OO

○ Lorenz y Kidd

- Número de guiones de escenario (casos de uso)
- Número de clases clave
- Número de clases de apoyo
- Número promedio de clases de apoyo por clase clave
- Número de subsistemas

○ Métricas orientadas a casos de uso

- Los CUs describen las funciones y características visibles al usuario.
- Directamente proporcional a los LOC y casos de prueba



Resumen.

Métricas

Mediciones, medidas y métricas

Escalas

Proceso de medición

Métricas de producto, proceso, proyecto y software

Aspectos medibles de cada uno y
ejemplos de métricas



Bibliografía

- ⦿ *Ingeniería del software. Un enfoque práctico* – R. Pressman
Capítulo 23 – Métricas de producto.
Capítulo 25 – Métricas de proceso y de Proyecto
- ⦿ *Software metrics. A rigorous & practical approach* – N. Fenton, S. L. Pfleeger
Capítulo 2 – The basics of measurement.
- ⦿ *Introducción a la ingeniería. Grech, Pablo – 2da edición. 2013*
Capítulo 4 – Mediciones, cálculos y toma de decisiones



Template: www.slidescarnival.com

Mg. M. Clara Casalini. 2018.

Introducción a la ingeniería de Software – Ingeniería en Sistemas de Información

Departamento de Ciencias e Ingeniería de la Computación – Universidad Nacional del Sur