



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# Clase 23

## Métodos estadísticos en ingeniería de software

IIC2143 - Ingeniería de Software  
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

17 de junio de 2018

# Principales desafíos

- Obtener datos de buena calidad
  - Es necesario recolectarlos durante el proceso
  - Los costos son claros pero los beneficios no tanto
- Darle significado a los datos
  - Creación de métricas

# Métricas

- A partir de datos se generan métricas
- Ejemplos vistos en el curso:
  - Puntos de función
  - Densidad de defectos
  - *Coverage de tests*
- Las métricas se deben generar a partir de necesidades

# Etapas de una métrica

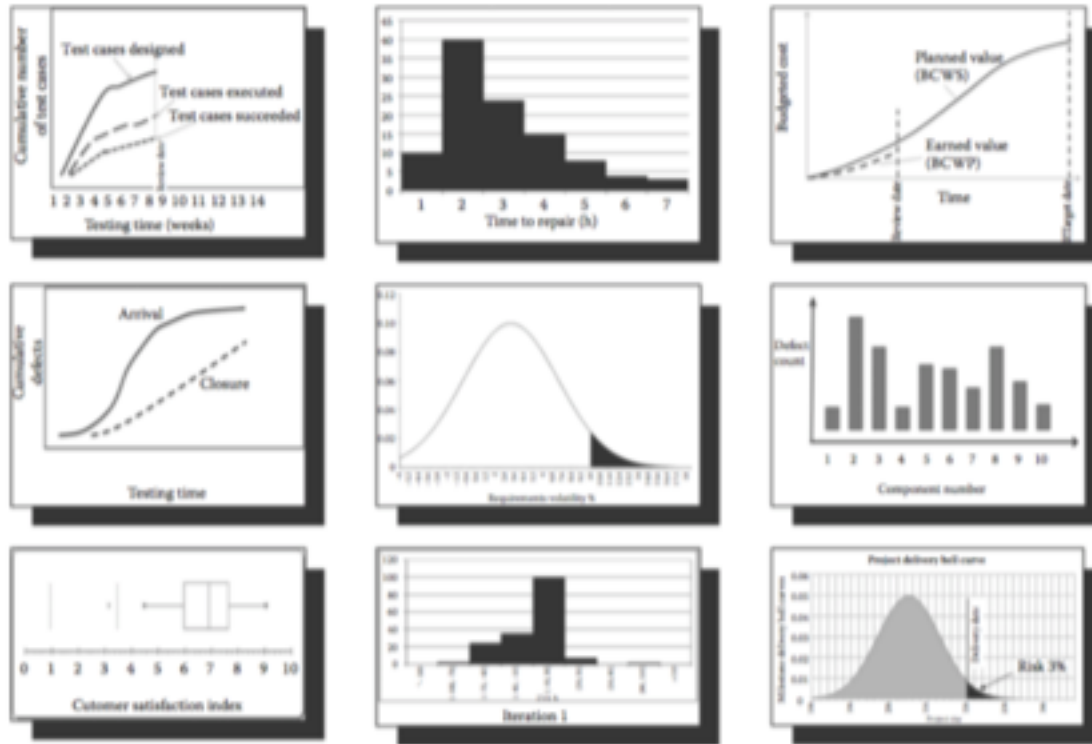
- **Formulación:** formalización de factores apropiados para representar el *software*
- **Recolección:** mecanismos para acumular datos a partir de la formulación
- **Análisis:** procesamiento de los valores recolectados para obtener información
- **Interpretación:** evaluación de la información para determinar mejoras
- **Retroalimentación:** recomendaciones para el equipo de desarrollo, derivadas de la interpretación

# Tipos de métricas

- **De negocios:** estrategia de la empresa
- **Del proyecto:** esfuerzo, calidad, productividad, satisfacción del usuario
- **Del proceso:** estabilidad de requisitos, complejidad, efectividad de los *tests*
- **De subprocesos:** esfuerzo de diseño, esfuerzo de revisión, esfuerzo de rediseño
- **Del producto:** tamaño del código, confiabilidad

# Visualización de métricas

- Las métricas son representadas con gráficos



# Características de una buena métrica

- Simple y computable
- Intuitiva
- Consistente y objetiva
- Unidades de medición expresivas
- Reflejar recomendaciones para mejorar
- Independiente del contexto (equipo, lenguaje de programación)

# Ejemplos de métricas

- Métricas de *testing*
- Métricas de diseño
- Métricas aplicaciones web



# Métricas de *testing*

## Estabilidad de requerimientos

$$RSI = \frac{\text{Original req} + \text{Req changed} + \text{Req added} + \text{Req deleted}}{\text{Original req}}$$

# Métricas de *testing*

## Efectividad de *tests*

$$\text{Test effectiveness} = \frac{\text{Defects found by tests}}{\text{Defects found by tests} + \text{Defects found by business users}}$$

# Métricas de *testing*

- Eficiencia en remover defectos

$$DRE = \frac{\text{Number of defects}}{\text{Detection time} + \text{Resolution time} + \text{Retesting time}}$$

- Tiempo de recuperación de defectos

$$TAT = \frac{1}{DRE}$$

# Métricas de diseño

## Complejidad ciclomática

- Es una métrica basada en el cálculo del número de caminos independientes que tiene el código
- Un buen valor de referencia es 11 (que representa métodos sencillos)

# Métricas de diseño

## **WMC: Weighted Methods per Class**

- Es la suma de la complejidad de cada método de una clase
- Es un indicador predictivo del esfuerzo necesario para mantener y extender una clase

# Métricas de diseño

## **DIT: Depth of Inheritance Tree**

- Es la distancia máxima de una clase base a una “hoja” de la jerarquía
- Refleja la complejidad del diseño

# Métricas de diseño

## **NOC: Number Of Children**

- Cantidad de subclases directas de una clase
- Refleja el nivel de abstracción del diseño

# Métricas de diseño

## **CBO: Coupling Between Object classes**

- Cantidad de clases con las que colabora una clase
- Altos niveles reflejan alto acoplamiento, lo que dificulta modificar el código



# Métricas de diseño

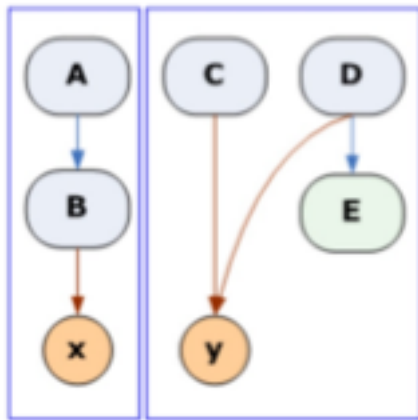
## **RFC: Response For a Class**

- Cantidad de métodos únicos invocados desde una clase (propios y los que utiliza directamente)
- Altos niveles reflejan alta complejidad, lo que dificulta entender y probar el código

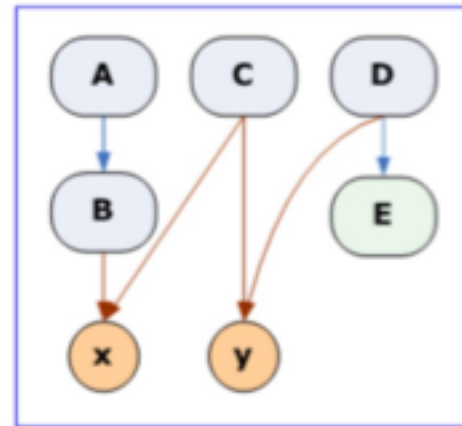
# Métricas de diseño

## LCOM: Lack of Cohesion Of Methods

- Cantidad de grupos de métodos que acceden a 1 o más atributos en común de una clase
- Altos niveles reflejan poca cohesión, lo que refleja que la responsabilidad de una clase no está bien definida



LCOM4 = 2



LCOM4 = 1



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# Clase 23

## Métodos estadísticos en ingeniería de software

IIC2143 - Ingeniería de Software  
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

17 de junio de 2018