

Clase 18: Métricas de calidad

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

6 de octubre de 2016

Contenidos

1 Repaso Clase Anterior

- ¿Qué es la calidad?
- Métricas de calidad

2 Métricas de calidad

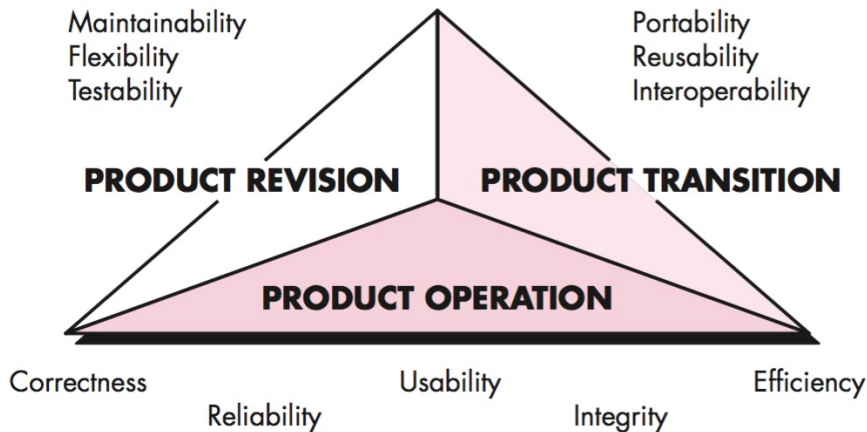
- CK metrics suite

Calidad según [Pressman, 2009]:

"Un desarrollo de *software* efectivo, aplicado de una manera que crea un producto útil que provee valor cuantificable para aquellos que lo producen y aquellos que lo utilizan."

Calidad en el *software*

Factores de Calidad [McCall, 1977]:



Características de una buena métrica:

- Simple y computable
- Intuitiva
- Consistente y objetiva
- Unidades de medición expresivas
- Independiente del lenguaje de programación
- Reflejar recomendaciones para mejorar la calidad del *software*

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

Propuestas por Chidamber & Kemerer [Chidamber, 1994]

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.

DIT: Depth of Inheritance Tree

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

NOC: Number Of Children

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

CBO: Coupling Between Object classes

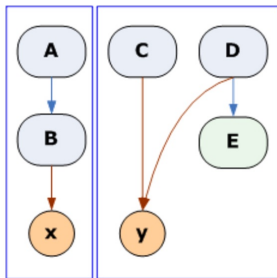
- Cantidad de clases con las que colabora una clase
- Altos niveles reflejan alto acoplamiento, lo que dificulta modificar el código

RFC: Response For a Class

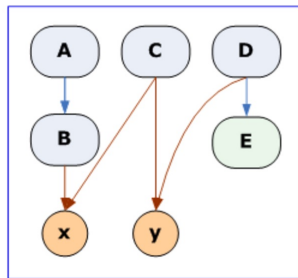
- Cantidad de métodos únicos invocados desde una clase (propios y los que utiliza directamente)
- Altos niveles reflejan alta complejidad, lo que dificulta entender y probar el código

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

Referencias



Pressman, R. S. (2009)

Software Engineering: A Practitioner's Approach

7th ed., *McGraw-Hill Education*



McCall, J., Richards, P., and Walters, G. (1977)

Factors in Software Quality

three volumes, NTIS AD-A049-014, 015, 055, November 1977.



Chidamber, S. R., and Kemerer, C. F. (1994)

A Metrics Suite for Object-Oriented Design

IEEE Trans. Software Engineering, vol. SE-20, no. 6, June 1994, pp. 476–493.

Fin