

Clase 19: Testing

Rodrigo Arturo Saffie Kattan

Pontificia Universidad Católica de Chile

rasaffie@ing.puc.cl

13 de octubre de 2016

1 Repaso Clase Anterior

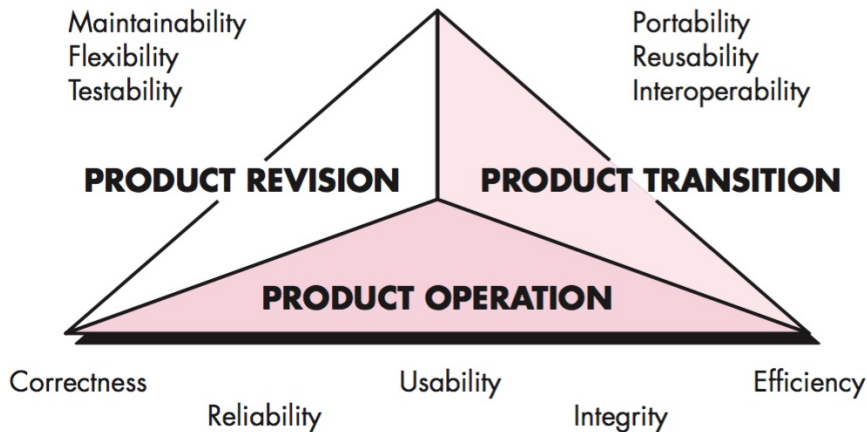
- Calidad
- CK metrics suite

2 Testing

- ¿Por qué falla el software?
- Validación y Verificación
- Unit Testing

Calidad en el *software*

Factores de Calidad [McCall, 1977]:



Propuestas por Chidamber & Kemerer [Chidamber, 1994]

- WMC: Weighted Methods per Class
- DIT: Deep of Inheritance Tree
- NOC: Number Of Children
- CBO: Coupling Between Object classes
- RFC: Response For a Class
- LCOM: Lack of Cohesion Of Methods

¿Por qué falla el software?

En los requisitos:

- Faltan requisitos
- Requisitos mal definidos
- Requisitos no realizables
- Diseño de software defectuoso

En la implementación:

- Algoritmos incorrectos
- Implementación defectuoso

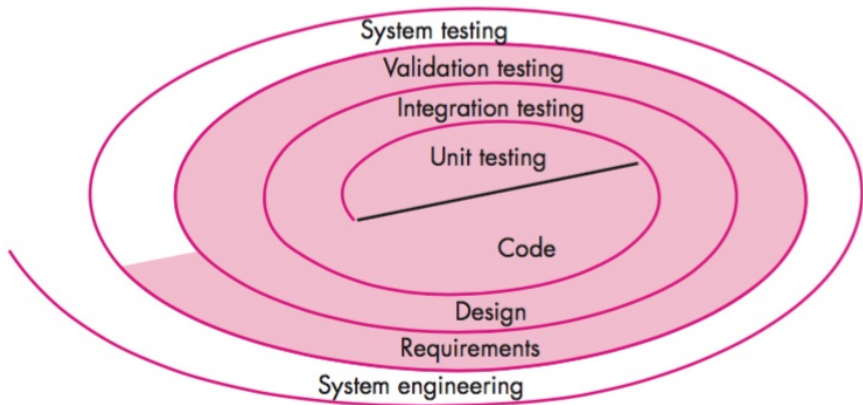
Validación:

¿Estamos construyendo el producto **correcto**?

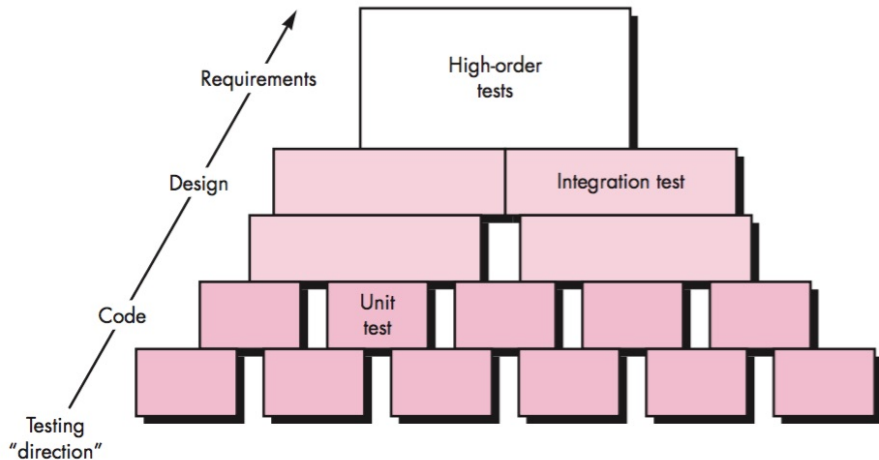
Verificación:

¿Estamos construyendo el producto **correctamente**?

Validación y Verificación



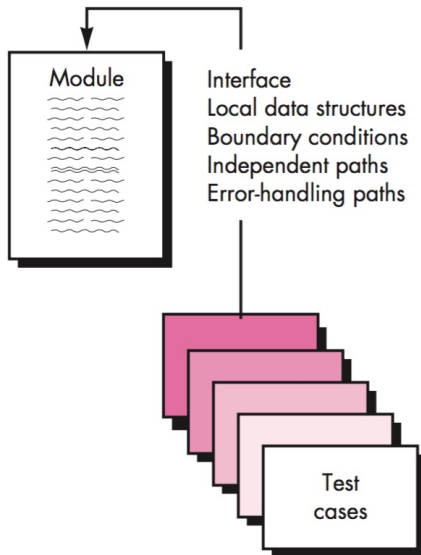
Validación y Verificación



Unit Testing

Se centran en verificar las unidades más pequeñas del software (componentes y módulos).

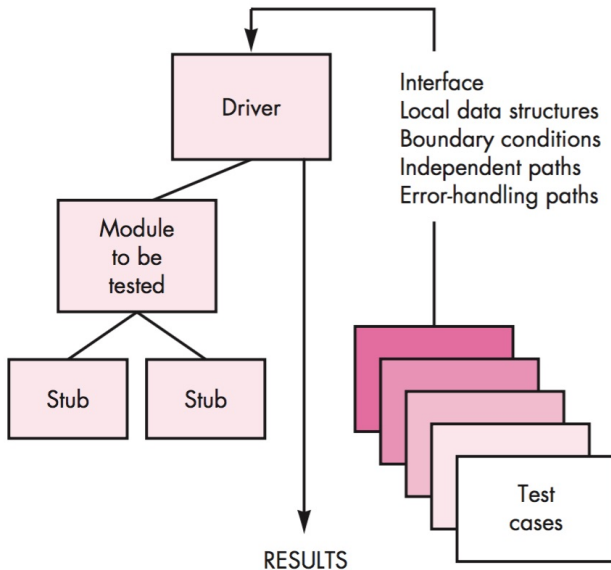
Unit Testing



Fallas en el manejo de errores:

- Descripción del error es confusa
- El error detectado es distinto al esperado
- El error genera una excepción que necesita intervención
- Manejo de la excepción es incorrecto
- El error registrado no es suficientemente descriptivo

Unit Testing



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