

# Tecnología de la Programación

## Software Quality

David Cabrero Souto

Facultad de Informática  
Universidade da Coruña

Curso 2007/2008



*In the context of software engineering, software quality measures how well software is designed (quality of design), and how well the software conforms to that design (quality of conformance), although there are several different definitions.*

*Wikipedia*

*The degree to which a set of inherent characteristics fulfills requirements.*

*ISO9001-00*



## Otros términos y referencias:

- “market-drive quality” (satisfacción total del cliente)  
*IBM.*
- “customer-driven quality”  
*National Institute of Standards and Technology*  
<http://www.quality.nist.gov>
- “Guide to the Software Engineering Body of Knowledge”  
<http://www.swebok.org/>



Concepto con una definición compleja.

- Ausencia de errores
- Calidad de un software de contabilidad vs. control de ABS vs. revelado fotográfico vs. tipografía vs. . . .
- Varía con la persona (desarrollador, usuario, cliente, director del proyecto, personal de marketing, . . .)



- Técnicas estáticas.
  - No precisan la ejecución del software.
  - Metodologías y procedimientos (Quality Assurance).
    - Estándares. P.e. ISO9001-00
    - Teorías de managment. P.e. Gestión del riesgo
    - Metodologías ágiles. P.e. Programación extrema
  - Herramientas de verificación formal y análisis del código.
- Técnicas dinámicas.
  - Precisan la ejecución del software o análisis del código.
  - Herramientas y tecnologías de validación o testing.



Es frecuente buscar un compromiso entre las características y niveles de calidad deseados y el coste de la implantación o no de las medidas necesarias en los procesos de ingeniería.

- Costes de prevención
- Costes de evaluación
- Coste interno del fallo
- Coste externo del fallo
- ...



- **Producto software**
  - Conforme con requerimientos o especificaciones
  - Escalabe, extensible, mantenible
  - Correcto, completo
  - Confiable (baja probabilidad de fallos)
  - Sin errores
  - Tolerante a fallos
  - Documentado
- **Código fuente**
  - Legible
  - Fácil de mantener, probar, depurar, corregir, portar y modificar
  - Complejidad reducida
  - Consumo de recursos bajo (cpu, memoria, e/s)
  - Número de avisos de compilación, lint, ...



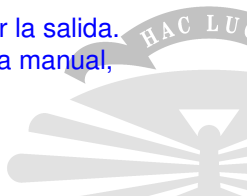
Comprobar corrección y búsqueda de errores.

- Verificación.

- Técnica estática. No se ejecuta el código.
- Análisis del código fuente
  - Manual
  - Automática o semi-automática.  
Métodos estadísticos, heurísticas, métodos formales.

- Validación (testing).

- Técnica dinámica. Se ejecuta el código.
- Ejecutar con unos datos de entrada y comprobar la salida.  
Los datos de entrada/salida se generan de forma manual, automática o mixta.





## Datos entrada/salida

a = ... -4 -3 -2 -1 0 1 2 3 4 5 6 ...  
b = ... -4 -3 -2 -1 0 1 2 3 4 5 6 ...  
...

## Propiedades

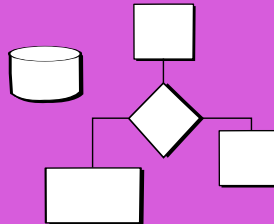
$\text{sqrt}(x)^2 = x$   
 $a + b = (2a + 2b) / 2$   
`write(file,x);read(file)=x`

seleccionar

Pruebas

ejecutar  
y  
comprobar

## Software



Formulación  
de las propiedades

Propiedades

$\text{sqrt}(x)^2 = x$   
 $a + b = (2a + 2b) / 2$   
`write(file,x);read(file)=x`

comprobar

Modelo  
del  
software

Software

