

# DESARROLLO DE SOFTWARE DE CALIDAD: *EXTREME PROGRAMMING* Y HERRAMIENTAS *OPENSOURCE*

M<sup>a</sup> Carmen Bartolomé  
mcbartolome@qualityobjects.com



**QUALITY OBJECTS**

## Índice

- ☞ Introducción a *eXtreme Programming* (XP)
- ☞ Herramientas *OpenSource*
  - Eclipse, CVS, Ant, JUnit y Cactus
- ☞ Ejemplo de Aplicación
  - Demostración
- ☞ Conclusiones
- ☞ Referencias

www.qualityobjects.com

SIMO-2003

## Introducción a XP

- ☞ Metodología para el desarrollo software
- ☞ Metodología "ágil"
- ☞ Basada en un conjunto de reglas y prácticas que se apoyan y se refuerzan entre ellas
- ☞ Tiene como objetivos principales:
  - La implicación y satisfacción del cliente
  - La calidad de código
  - El trabajo en equipo y la productividad

SIMO-2003

www.qualityobjects.com

## Filosofía de XP

**Personas** por encima de **herramientas**  
**Creatividad** por encima de **estandarización**  
**Calidad** por encima de **coste**  
**Agilidad** por encima de **procesos**  
**Iteraciones** por encima de **modelos**  
**Código** por encima de **diagramas**  
**Comunicación** por encima de **documentación**  
**Cliente** por encima de **contratos**  
**Flexibilidad** por encima de **planificaciones**

SIMO-2003

www.qualityobjects.com

## QOXP (I)

### Planificación

- Utilización de las *Historias de Usuario* como requisitos del proyecto.
- Planificación de versiones junto con el cliente.
- Medición de la *velocidad* de proyecto.
- Planificación de iteraciones.
- Asignar equipos de desarrollo por *Historias de Usuario* (asignación vertical).
- Mover la gente en el proyecto.
- Reuniones de pie.
- Todo el mundo codifica en un proyecto.
- Focalización en el código y pruebas, no en la documentación modelos, informes.
- No hay inspecciones de calidad.

### Desarrollo

- El cliente está involucrado en el proyecto.
- Estilos compartidos en el equipo.
- **Se codifica el sistema y sus pruebas al mismo tiempo.**
- La codificación se realiza en parejas y el conocimiento del código es colectivo.
- **Integración secuencial y continua.**
- Se piensa primero, se codifica después.

SIMO-2003

www.qualityobjects.com

## QOXP (II)

### Diseño

- Simplicidad.
- Creación de aplicaciones de pruebas cuando se detecta riesgo.
- Se desarrolla única y estrictamente lo que se pide.
- Refactorizar entre iteraciones (arreglar ventanas rotas).
- Dar libertad a los desarrolladores en la toma de decisiones.

### Testing

- Todo el código se prueba.
  - **Las pruebas deben estar lo más automatizadas posible y empaquetadas junto al código.**
- Los bugs se corrigen y se crean casos de prueba nuevos si es necesario.
- V&V se ejecuta continuamente por el cliente.

SIMO-2003

www.qualityobjects.com

## Herramientas (I)

☞ Las herramientas elegidas deben proporcionar:

- Automatización
  - “one-button testing”
  - Compilar, deploy y probar la aplicación
- Tests de Regresión
- Consistencia entre desarrolladores
- Integración continua
  - En una máquina dedicada: manualmente o utilizando una herramienta.

☞ *OpenSource*

☞ Lenguaje de Programación: Java

www.qualityobjects.com

SIMO-2003

## Herramientas (II)



www.qualityobjects.com

SIMO-2003

## Eclipse

### ¿Qué es Eclipse?

- Entorno de desarrollo (IDE)
- Plataforma de integración de herramientas

### Funcionalidad básica

- Se extiende con Plug-ins: unidad básica e independiente
- Plug-ins utilizados
  - Ant
  - CVS
  - JUnit
  - Cactus

www.qualityobjects.com

SIMO-2003

## Concurrent Version System (CVS)

- ✓ Herramienta de gestión de versiones
- ✓ Almacena ficheros y su historial en un **repositorio** compartido
- ✓ Funcionalidad básica: mantener y actualizar ficheros
- ✓ Principios
  - Tareas pequeñas
  - Sincronización con el repositorio
- ✓ Integrado con Eclipse

www.qualityobjects.com

SIMO-2003

## Another Neat Tool (ANT)

- ✓ Utilidad de building basada en Java
- ✓ Equivalente a *make*
- ✓ Controla el proceso de building de software y garantiza un entorno base común.
- ✓ Ficheros build.xml:
  - Análogos a los "Makefile" para make
  - Ant usa este fichero para decidir lo que va a hacer y cómo lo va a hacer
  - Estructura XML
  - Define targets e información de configuración.

www.qualityobjects.com

SIMO-2003

## Pruebas

- ✓ Filosofía XP: "Test-Driven"
  - Primero escribir el test y después escribir el código
  - Pruebas automatizadas
- ✓ Pruebas unitarias
  - JUnit
  - Cactus (aplicaciones Web)

www.qualityobjects.com

SIMO-2003

## Pruebas Unitarias: JUnit

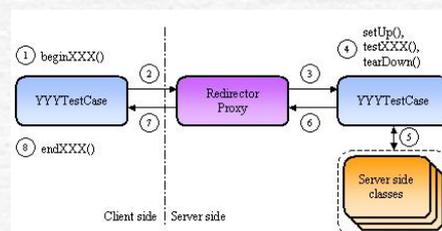
- JUnit es un *framework* a nivel de programador para escribir pruebas unitarias en Java
- Integrado en Eclipse y en Ant
- Los casos de prueba son subclases de *TestCase* y sobrescriben el método *runTest*
- Se ejecutan con herramientas *TestRunner* incluidas en JUnit

www.qualityobjects.com

SIMO-2003

## Cactus

- Framework* para probar aplicaciones web
- Basado en JUnit
- Implementa una estrategia *in-container*
  - Relación entre el código J2EE y su contenedor
- Replica el ciclo petición-respuesta HTTP



www.qualityobjects.com

SIMO-2003

## Ejemplo de aplicación

- ☞ Añadir una nueva funcionalidad al sistema.
- ☞ Desarrollo en local:
  1. Obtener los últimos ficheros del CVS
  2. Escribir una prueba unitaria con JUnit que intente ejecutar la nueva funcionalidad
  3. Ejecutar el test -> Fallo
  4. Escribir el código
  5. Ejecutar de nuevo la prueba
  6. Repetir los pasos 2-5
  7. Actualizar el código con el CVS
  8. Ejecutar los tests
  9. Introducir los cambios en el CVS
- ☞ Integración continua y secuencial en una máquina dedicada.

SIMO-2003

www.qualityobjects.com

## Conclusiones

- ☞ Bases para desarrollo de software de calidad según XP:
  - Importancia de las pruebas
  - Integración continua
- ☞ Necesidad de herramientas adecuadas
  - OpenSource
- ☞ Validez del escenario propuesto
  - Automatización del proceso de desarrollo: building, deploy y testing.
  - Integración continua

SIMO-2003

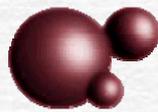
www.qualityobjects.com

## Referencias

- ☞ <http://www.extremeprogramming.org>
- ☞ Eclipse: <http://www.eclipse.org>
- ☞ CVS: <http://www.cvshome.org>
- ☞ Ant: <http://ant.apache.org>
- ☞ Junit: <http://www.junit.org>
- ☞ Cactus: <http://jakarta.cactus.org>
- ☞ "Java Tools for eXtreme Programming" R. Hightowe, N. Lesiecki, Wiley 2002
- ☞ "Java Extreme Programming Cookbook", E. Burke, B. Coyner, O'Reilly, 2003
- ☞ "eXtreme Programming Explained" Kent Beck, Addison Wesley, 2000

www.qualityobjects.com

SIMO-2003



**QUALITY OBJECTS**

**<http://www.qualityobjects.com>**

C/ Agastia, 44-46 – 1ª Planta  
28027 Madrid  
Tel.: 91 406 68 90  
Fax: 91 406 68 91  
e-mail: [info@qualityobjects.com](mailto:info@qualityobjects.com)