

# Extreme Programming

## Un nuevo método de desarrollo de software

César F. Acebal y Juan M. Cueva Lovelle

*Universidad de Oviedo*

### ¿Qué es XP?

✍ Extreme Programming (XP) es un **método ligero** de desarrollo de software pensado para **equipos pequeños o medianos** que se enfrentan a **requisitos vagos o cambiantes**

- Ligero en contraposición a otros métodos pesados o burocráticos como Métrica
- Equipos de como máximo 10 o 12 personas
  - La comunicación es primordial
- El cliente no puede saber exactamente qué es lo que quiere hasta que no ha visto varias iteraciones del producto

## ¿Por qué "extremo"?

✍ Porque lleva un conjunto de técnicas y principios de sentido común a niveles extremos:

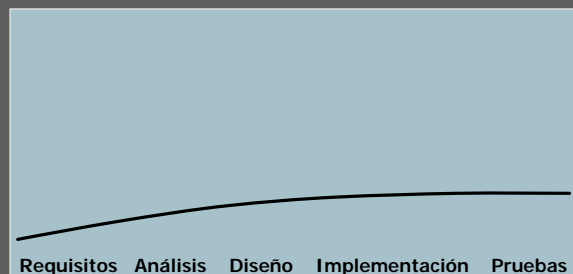
- El código será revisado continuamente, mediante la **programación en parejas** (dos personas por máquina)
- Se harán **pruebas** todo el tiempo, mediante *frameworks* de pruebas automatizadas, como el *xUnit*
  - De cada nueva clase (pruebas unitarias)
  - De integración, al añadir una clase o modificar cualquiera de las existentes
  - El cliente comprobará que se van satisfaciendo los requisitos (pruebas funcionales)
- Se rediseñará todo el tiempo (**refactoring**), dejando el código siempre en el estado más simple posible
- Las **iteraciones** serán mucho más cortas de lo habitual
  - Equipo y cliente se beneficiarán así de la retroalimentación

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

3

## Coste del cambio

- ✍ Hasta ahora, era un hecho asumido por todos que el coste del cambio se incrementaba exponencialmente en el tiempo
- ✍ Con una combinación de buenas prácticas de programación y tecnología, XP afirma que es posible obtener la siguiente curva para el coste del cambio:



Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

4

## Consecuencias del nuevo coste del cambio

### ✍ Si los costes no se incrementan drásticamente con el tiempo:

- Diseñaremos tan sencillo como sea posible
- Reduiremos la ambigüedad, ahorrando dinero
  - Sólo se implementará aquello que el cliente realmente necesita

### ✍ En resumen:

- Sabemos que los requisitos van a cambiar
- En vez de luchar contra el cambio, nos adaptamos a él

### ✍ Dificultad:

- Lograr que la curva sea cierta
- Es lo que XP pretende conseguir con una serie de buenas prácticas

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

5

## Las prácticas

*Estándares de  
codificación*

*Cliente en  
el sitio*

*Planificación*

*40 horas*

*Versiones  
cortas*

*Pruebas*

*Integración  
continua*

*Metáfora*

*Código colectivo*

*Diseño  
simple*

*Refactoring*

*Programación  
en parejas*

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

6

## La planificación

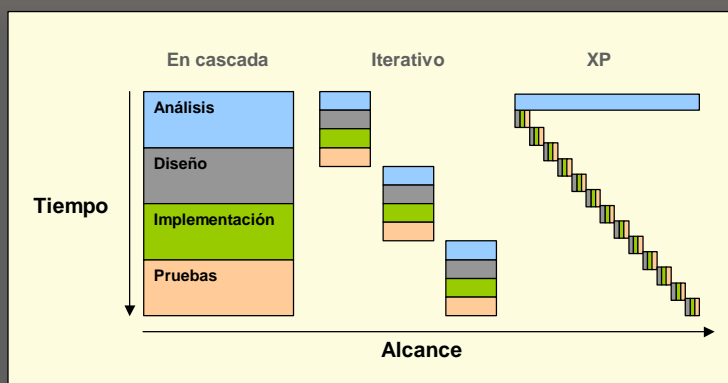


- ✍ Los clientes escriben *historias* con las funciones del proyecto
  - Debe poder estimarse fácilmente su *dificultad* (una o dos semanas)
  - Deben poder escribirse pruebas funcionales para cada una
- ✍ Se estima la duración total del proyecto basándose en *cuatro variables*:
  - *Coste, tiempo, calidad y alcance*
- ✍ El cliente escoge las historias a implementar en esta *versión*
- ✍ Cada versión se divide en *iteraciones* de 1 a 3 semanas
- ✍ Cada historia se divide en *tareas* de entre 1 y 3 días
  - Los programadores se asignan las tareas y estiman su duración
- ✍ XP distingue entre *tiempo de calendario* y *tiempo ideal*
  - *Velocidad del equipo = semanas ideales por iteración*

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

7

## El ciclo de vida



Evolución de los largos ciclos de desarrollo del modelo en cascada a los ciclos iterativos más cortos de, por ejemplo, el modelo en espiral y, finalmente, a la mezcla que XP hace de todas las actividades a lo largo de todo el ciclo de vida

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

8

## Versiones cortas

- ✍ Cada pocos meses
- ✍ El cliente elige las historias a implementar en la presente versión
- ✍ Puede añadir y quitar historias en cualquier momento
- ✍ Se deberá rehacer la planificación cuando:
  - El montón de historias retrasadas crece demasiado
  - Cambia la velocidad del equipo

### Ejemplo

*Si se están haciendo unas 8 semanas ideales por iteración y quedan 3 iteraciones para esta versión, el cliente selecciona historias por un valor de 24 semanas*



## Metáfora

- ✍ En XP, representa la arquitectura del sistema
- ✍ Ayuda a comprender los elementos básicos del sistema y sus relaciones
  - Facilita la comunicación entre los miembros del equipo
- ✍ Expresa el sistema en términos de algo conocido y fácilmente comprensible por todos
  - "Es como..."
- ✍ Todo se expresará en términos de la metáfora
- ✍ La metáfora se irá refinando



## Diseño simple



### ✍ No trataremos de adivinar el futuro

- En vez de eso, nos adaptaremos al cambio
- De esta manera, XP proporciona más valor al software
  - No invierte en características que nunca serán utilizadas

### ✍ El diseño correcto es aquél que:

- Ejecuta satisfactoriamente todas las pruebas
- No tiene lógica duplicada
- Está autodocumentado, revela su intención
- Tiene el menor número posible de clases y métodos

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

11

## Pruebas

**Cualquier característica de un programa sin una prueba automatizada simplemente no existe.**

### ✍ Los programadores escriben pruebas unitarias

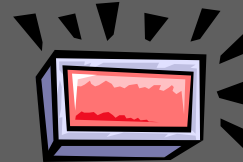
- Antes incluso que la propia clase a probar

### ✍ Los clientes escriben pruebas funcionales

### ✍ Las pruebas se ejecutan continuamente (xUnit)

### ✍ Resultado:

- A medida que el sistema crece aumenta nuestra confianza en él
- Es sencillo modificar el sistema (las pruebas aseguran que no “rompemos” nada)

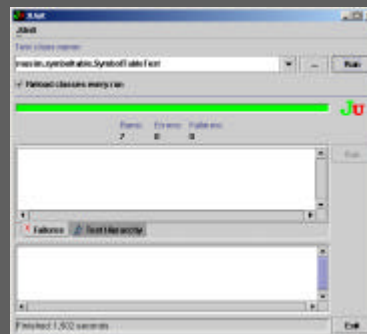


Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

12

## xUnit

- ✍ Es un *framework* de pruebas automatizadas
- ✍ Ejecutar todas las pruebas es tan sencillo como pulsar un botón
  - Se hará, por tanto, varias veces al día
- ✍ Asegura que las pruebas se satisfagan siempre al 100%
- ✍ Funciona a modo de aserciones, pero fuera del código
- ✍ Por cierto, es gratuito



Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

13

## Refactoring

- ✍ Consiste en mejorar la estructura interna del código sin alterar su comportamiento externo
- ✍ Trata de dejar el código en el estado más simple posible
  - Eliminar la lógica duplicada
- ✍ No añade características nuevas
- ✍ ¿Por qué modificar código que funciona correctamente?
  - Para que la siguiente característica sea fácil de añadir
- ✍ ¿Cuándo hacer *refactoring*?
  - Siempre que el sistema nos lo pida
  - Si un programador ve la posibilidad de simplificar alguna porción del código, deberá hacerlo inmediatamente, antes de añadir nuevas funcionalidades
- ✍ Ejemplos (catálogo de factorizaciones de Martin Fowler):
  - Extract Method, Replace Temp with Query, Replace Conditional with Polymorphism

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

14

## Programación en parejas



- ✍ Todo el código es desarrollado de dos en dos
  - Dos personas compartiendo un monitor y teclado
- ✍ Hay dos roles en cada pareja de programadores:
  - Uno piensa en la mejor manera de implementar el método actual
  - El otro lo hace de una manera más estratégica:
    - ¿Estamos siguiendo el enfoque correcto?
    - ¿Qué otros casos de prueba podrían seguir sin funcionar?
    - ¿Hay forma de simplificar el sistema completo?
- ✍ Ambos roles son intercambiables
- ✍ Las parejas cambiarán a lo largo del día
  - Según vayan siendo requeridos por otros programadores para ayudarles

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

15

## Propiedad colectiva del código

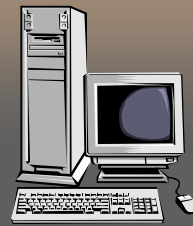
- ✍ Cualquiera puede modificar cualquier porción del código, en cualquier momento
- ✍ Hay que desterrar el acusado sentimiento de propiedad del código que tienen algunos programadores
- ✍ ¿En qué se apoya?
  - Estándares de codificación
  - Las pruebas aseguran que tras el cambio todo sigue funcionando
  - La programación en parejas reduce el riesgo de fallos
  - La integración continua reduce la posibilidad de conflictos

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

16

## Integración continua

- ✍ El código se integra y prueba en su totalidad cada pocas horas
- ✍ Hay una máquina así llamada, de integración
- ✍ Cuando una clase haya pasado sus pruebas unitarias, se añaden –clase y pruebas unitarias– al sistema
- ✍ El sistema debe quedar siempre funcionando al 100%
- ✍ Si alguna prueba falla, la pareja de programadores volverán a su sitio a refinar la clase
  - Si pasado un cierto tiempo no son capaces de arreglarlo, tirarán el código y volverán a empezar de cero



Máquina de integración

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

17

## 40 horas semanales

- ✍ Es imposible hacer software de calidad trabajando 60 horas semanales
- ✍ La gente debe ir cada mañana descansada y tranquila
  - Si sentimos un nudo en el estómago es que nuestro proyecto tiene graves problemas
- ✍ Nunca se deberá hacer horas *extra* más de una semana seguida
  - Estaríamos hablando de un problema que no se soluciona trabajando más horas



*Presencia ? Productividad*

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

18

## Cliente en el sitio

- ✍ El cliente debe ser un miembro más del equipo
- ✍ Estará a jornada completa con el equipo de desarrollo
  - Contestando preguntas
  - Resolviendo disputas
  - Estableciendo prioridades
- ✍ Si el trabajo de una o dos personas es más valioso que el proyecto, quizás éste no merezca ser realizado
- ✍ Si el cliente no accede, serán los programadores los que hagan suposiciones –erróneas– acerca del negocio

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

19

## Conclusiones

- ✍ XP es un método ligero que preconiza la adaptación al cambio (no trata de predecirlo)
- ✍ Son una serie de prácticas sinérgicas
  - Ninguna es nueva, lo que hace el método es juntarlas
- ✍ A nuestro juicio, dos son las piezas clave:
  - *Las pruebas*
  - *El diseño simple* (no *sobrediseñar*)
- ✍ Inconveniente:
  - Se necesita gente buena para llevarlo a cabo

Extreme Programming. Un nuevo método de desarrollo de software  
César F. Acebal y Juan M. Cueva Lovelle

20