

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/318211906>

# La programación extrema

Technical Report · November 2013

DOI: 10.13140/RG.2.2.29359.43687

---

CITATIONS

0

---

READS

8,320

2 authors, including:



[Antonio Collazo Garcia](#)

CIT Santa Cruz - CONICET

6 PUBLICATIONS 1 CITATION

SEE PROFILE

# La Programación Extrema

**Autores:** Marycarmen Díaz Labrador  
Antonio Collazo Garcia

# Índice

Introducción.....	3
Desarrollo.....	4
Fases .....	4
Fase de exploración .....	4
Fase de planificación .....	4
Fase de iteraciones .....	4
Fase de puesta en producción .....	4
Etapas.....	4
Planificación.....	4
Diseño .....	6
Implementación .....	6
Pruebas .....	7
Valores en XP .....	7
Roles .....	7
Cliente .....	7
Programador .....	7
Encargado de Pruebas (Tester) .....	7
Encargado de Seguimiento (Tracker) .....	8
Entrenador (Coach) .....	8
Gestor (Big Boss) .....	8
Conclusiones.....	9

## **Introducción**

La programación Extrema o eXtreme Programming (XP) es una metodología de desarrollo de la ingeniería de software. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

# Desarrollo

## Fases

### Fase de exploración

Es la fase en la que se define el alcance general del proyecto. En esta fase, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuario”. Los programadores estiman los tiempos de desarrollo en base a esta información. Debe quedar claro que las estimaciones realizadas en esta fase son primarias (ya que están basadas en datos de muy alto nivel), y podrían variar cuando se analicen en más detalle en cada iteración. Esta fase dura típicamente un par de semanas, y el resultado es una visión general del sistema, y un plazo total estimado.

### Fase de planificación

La planificación es una fase corta, en la que el cliente, los gerentes y el grupo desarrolladores acuerdan el orden en que deberán implementarse las historias usuario, y, asociadas a éstas, las entregas. Típicamente esta fase consiste en una varias reuniones grupales de planificación.

### Fase de iteraciones

Esta es la fase principal en el ciclo de desarrollo de XP. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una un entregable funcional que implementa las historias de usuario asignadas a la iteración. Como las historias de usuario no tienen suficiente detalle como para permitir su análisis y desarrollo, al principio de cada iteración se realizan las tareas necesarias de análisis, recabando con el cliente todos los datos que sean necesarios. El cliente, por lo tanto, también debe participar activamente durante esta fase del ciclo. Las iteraciones son también utilizadas para medir el progreso del proyecto. Una iteración terminada sin errores es una medida clara de avance.

### Fase de puesta en producción

Si bien al final de cada iteración se entregan módulos funcionales y sin errores, puede ser deseable por parte del cliente no poner el sistema en producción hasta tanto no se tenga la funcionalidad completa. En esta fase no se realizan más desarrollos funcionales, pero pueden ser necesarias tareas de ajuste.

## Etapas

### Planificación

La metodología XP plantea la planificación como un dialogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una.

Si alguna de ellas tiene “riesgos” que no establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba (“spikes”), para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas (“Release Plan”) en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en dónde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios”.

Los planes en XP se diferencian de las metodologías tradicionales en tres aspectos:

1. Simplicidad del plan. No se espera que un plan requiera de un “gurú” con complicados sistemas de gerenciamiento de proyectos.
2. Los planes son realizados por las mismas personas que realizarán el trabajo.
3. Los planes no son predicciones del futuro, sino simplemente la mejor estimación de cómo saldrán las cosas. Los planes son útiles, pero necesitan ser cambiados cuando las circunstancias lo requieren. De otra manera, se termina en situaciones en las que el plan y la realidad no coinciden, y en estos casos, el plan es totalmente inútil.

### **Historias de usuario**

Las “Historias de usuarios” sustituyen a los documentos de especificación funcional, y a los “casos de uso”. Estas “historias” son escritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar. La diferencia más importante entre estas historias y los tradicionales documentos de especificación funcional se encuentra en el nivel de detalle requerido. Las historias de usuario deben tener el detalle mínimo como para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. Cuando llegue el momento de la implementación, los desarrolladores dialogarán directamente con el cliente para obtener todos los detalles necesarios.

Las historias de usuarios deben poder ser programadas en un tiempo entre una y tres semanas. Si la estimación es superior a tres semanas, debe ser dividida en dos o más historias. Si es menos de una semana, se debe combinar con otra historia.

### **Plan de entregas**

El cronograma de entregas establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto (cliente, desarrolladores, gerentes, etc.). Típicamente el cliente ordenará y agrupará según sus prioridades las historias de usuario. El cronograma de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores.

### **Plan de iteraciones**

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido. Al comienzo de cada ciclo, se realiza una reunión de planificación de la iteración. Cada historia de usuario se traduce en tareas específicas de programación. Asimismo, para cada historia de usuario se establecen las pruebas de aceptación. Estas pruebas se realizan al final del ciclo en el que se desarrollan, pero también al final de cada uno de los ciclos siguientes, para verificar que subsiguientes iteraciones no han afectado a las anteriores.

Las pruebas de aceptación que hayan fallado en el ciclo anterior son analizadas para evaluar su corrección, así como para prever que no vuelvan a ocurrir.

### **Reuniones diarias de seguimiento**

El objetivo de tener reuniones diarias es mantener la comunicación entre el equipo, y compartir problemas y soluciones. En la mayoría de estas reuniones, gran parte de los participantes simplemente escuchan, sin tener mucho que aportar. Para no quitar tiempo

innecesario del equipo, se sugiere realizar estas reuniones en círculo y de pie.

## **Diseño**

La metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes:

### **Simplicidad**

Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Características fundamentales del código: Testeable, legible, comprensible y explicable.

### **Metáforas**

Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo.

### **Solución “spike”**

Una solución “spike”, es una solución muy simple para plantear posibles soluciones, de manera, que solamente se aborda el problema en concreto y se aísla de otro tipo de preocupaciones.

### **Refactorización**

La recodificación consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de hacerlo más simple, conciso y/o entendible.

Muchas veces, al terminar de escribir un código de programa, pensamos que, si lo comenzáramos de nuevo, lo habríamos hecho en forma diferente, más clara y eficientemente.. Las metodologías de XP sugieren recodificar cada vez que sea necesario. Si bien, puede parecer una pérdida de tiempo innecesaria en el plazo inmediato, los resultados de ésta práctica tienen sus frutos en las siguientes iteraciones, cuando sea necesario ampliar o cambiar la funcionalidad. La filosofía que se persigue es, como ya se mencionó, tratar de mantener el código más simple posible que implemente la funcionalidad deseada.

## **Implementación**

### **Cliente disponible**

Uno de los requisitos de XP es tener al cliente disponible. No solo para ayudar al equipo de desarrollo, sino para ser parte del mismo. Todas las fases de XP requieren comunicación con el cliente.

Las historias de usuario son escritas por el cliente con la ayuda de los desarrolladores, además de establecer la prioridad de las mismas. Su presencia asegura que los desarrollos cubren toda la funcionalidad descrita.

### **Estándares de codificación**

Todos los programadores deben escribir y documentar el código en la misma manera. El código debe seguir los estándares de codificación. Las normas de codificación ayudan a mantener el código legible y fácil de mantener y refactorizar.

### **Implementación dirigida por las Pruebas Unitarias**

En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de los tests, se realiza al final del proyecto, o al final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, en el que, lo primero que se escriben los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. Las pruebas a las que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al

comienzo, condiciona y dirige el desarrollo.

### **Programación en parejas**

XP promueve que todo el código sea escrito en parejas trabajando en el mismo ordenador. La programación en parejas incrementa la calidad del código sin impactar en la fecha de entrega. En contra de lo que parece, dos personas que trabajan en un mismo equipo añadirán la misma funcionalidad que dos personas trabajando por separado, excepto que el código será de mucha mayor calidad.

## **Pruebas**

### **Pruebas Unitarias**

Las pruebas unitarias son una de las piedras angulares de XP. Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código ("Test-driven programming"). Que todo código liberado pase correctamente las pruebas unitarias es lo que habilita que funcione la propiedad colectiva del código. En este sentido, el sistema y el conjunto de pruebas debe ser guardado junto con el código, para que pueda ser utilizado por otros desarrolladores, en caso de tener que corregir, cambiar o recodificar parte del mismo.

## **Valores en XP**

Simplicidad: Hacer exactamente lo que se ha pedido, no más.

Comunicación: La comunicación entre los componentes del equipo XP es fundamental. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.

Retroalimentación: Siempre tener en cuenta la valoración del cliente una vez que se hace una entrega e intentar mejorar haciendo cambios en el proceso si es necesario.

Coraje: Se trata que el equipo asuma la responsabilidad de su trabajo, tanto si es un éxito como un fracaso, además de ser emprendedor a la hora de implementar cambios en la aplicación.

## **Roles**

### **Cliente**

El cliente es el responsable de conducir el proyecto. Define el proyecto y sus objetivos. Cuanto más preciso es su trabajo y cuanto mayor sea su involucración, mayores serán las oportunidades de éxito.

### **Programador**

Una vez que se han comprendido las historias de usuario, el XP adjudica a los programadores la responsabilidad de tomar decisiones técnicas. Los desarrolladores estiman el tiempo que les va a tomar cada historia.

### **Encargado de Pruebas (Tester)**

El encargado de pruebas ayuda al cliente a definir y escribir las pruebas de aceptación de las historias de usuario. Este rol en un equipo XP también es responsable realizar test periódicamente y informar de los resultados al equipo. A medida que el volumen de pruebas aumenta, el encargado de pruebas necesitará una herramienta para crear y



mantener la batería de pruebas, ejecutarlas y obtener los resultados más rápidamente.

### **Encargado de Seguimiento (Tracker)**

Hace el seguimiento de acuerdo a la planificación. La métrica más importante para XP es la velocidad del equipo, que se define como el tiempo ideal estimado para las tareas frente al tiempo real dedicado. Esta métrica ayuda a determinar si el proyecto está dentro del tiempo de la iteración.

### **Entrenador (Coach)**

No es un rol cubierto en todos los equipos de XP. Su papel es guiar y orientar al equipo, especialmente cuando un equipo comienza a trabajar siguiendo la metodología XP. Esto se debe a que no es fácil aplicar XP de forma consistente. Aunque son prácticas de sentido común, se necesita un tiempo para interiorizarlas. También hay situaciones especiales en las que se requiere la sabiduría de un especialista en XP para aplicar sus normas frente a un obstáculo en el proyecto.

### **Gestor (Big Boss)**

Es el gerente del proyecto, debe tener una idea general del proyecto y estar familiarizado con su estado. El cliente puede asumir este papel.

## Conclusiones

Una de las principales preocupaciones del XP es reducir costes y hacer del desarrollo adaptado a lo que realmente hace falta para mejorar el software. En cuanto a los valores positivos del XP, se destacan la importancia de la comunicación en el equipo, el feedback y la valoración constante del trabajo para determinar si vamos en la dirección correcta.