

Programación extrema en pocos minutos:

planificando la transición

Por Ing. Santiago Luis Valdarrama del Pino
Departamento de Gestión y Administración de Aplicaciones Filial de Tecnología y Software,
Gerencia Territorial Camagüey, ETECSA
svpino@cmg.etcscu

Metodologías de Diseño

El desarrollo de software ha transitado por tres espacios bien definidos. El primero de ellos, conocido como codificar y corregir, fue factible mientras los programas eran pequeños y los requisitos cambiaban con poca frecuencia; pero los sistemas fueron creciendo, y la dificultad para agregar nuevas características y realizar modificaciones aumentó de forma exponencial.

Afortunadamente, surgió una nueva alternativa para contrarrestar los efectos negativos de codificar y corregir: las metodologías. Una metodología impone el proceso disciplinado en el desarrollo de un software para lograr que este sea lo más predecible y eficaz posible. La utilización de metodologías da lugar a un proceso que requiere del análisis detallado, haciéndose hincapié especialmente en la planificación del producto a desarrollar.

Al igual que el primer espacio, el constante cambio de los requisitos en un sistema provoca transformaciones significativas en los documentos desarrollados que hacen uso de las metodologías. Además, en la etapa de diseño es prácticamente imposible prever cada uno de los puntos con los que se deberá lidiar en el momento de programar, por lo tanto, después de comenzada la codificación, irán apareciendo, cada

vez con más frecuencia, evidencias que cuestionen el diseño realizado. Esto provoca, en el desarrollo de un proyecto, que la mayor parte del tiempo se concentre en la documentación burocrática del proceso en vez de en su codificación.

Las conocidas metodologías ligeras o de poco peso —*lightweight methodologies*— han ganado auge en los últimos tiempos. Surgen como contraparte de sus dos antecesores, en la búsqueda de un punto central entre la inexistencia de un proceso y un proceso exageradamente grande. Las metodologías ligeras —también conocidas como metodologías ágiles (*agile methodologies*)— realzan la importancia de la programación del código más que la documentación del proceso.

En Cuba, el desarrollo de software se ha convertido en una actividad fundamental como medio de soporte a la actividad interna de las entidades que conforman las empresas cubanas. En un mundo en constante revolución, la búsqueda de soluciones para potenciar el desarrollo de sistemas de computación ocupa un lugar preponderante en la agenda del personal involucrado. Por este motivo, con el objetivo de introducir su utilización en la empresa de software cubana, se analiza uno de los exponentes más populares de

las metodologías ágiles existentes: la Programación Extrema.

Programación Extrema

La Programación Extrema, conocida coloquialmente como XP —del inglés *eXtreme Programming*, y nada tiene que ver con el término utilizado por Microsoft en Windows XP que viene de la palabra *eXPerience*—, pone de manifiesto un grupo de principios y prácticas que aceleran el proceso de desarrollo de software, con la eliminación de las técnicas utilizadas innecesariamente por las metodologías pesadas. XP ha rejuvenecido la noción del diseño evolutivo con prácticas que permiten que dicha evolución se transforme en una estrategia de diseño viable y segura.

Durante años, uno de los principales problemas en el desarrollo de software ha sido el costo de realizar modificaciones en un sistema, pues durante su desarrollo se incrementa exponencialmente con el tiempo. Los requisitos de los usuarios o clientes cambian y serán diferentes cada año, cada mes o cada día. Si el sistema no puede cambiar rápidamente, en corto tiempo perderá la órbita prevista. La Programación Extrema acoge el cambio y permite transformar la dirección, violenta y frecuentemente, del desarrollo de

la aplicación sin afectar el producto final y el calendario previsto.

Por otra parte, un desarrollo de software exitoso es el producto de un esfuerzo de equipo, no solamente de sus desarrolladores, sino también de sus directivos y clientes. XP es un proceso de desarrollo de software simple que une a todas estas personas y las lleva juntas al éxito.

En su libro *“Extreme Programming Explained: Embrace the Change”*, Kent Beck subraya los valores fundamentales de la Programación Extrema. Estos valores pueden ser resumidos de la siguiente forma:

1. Comunicación: no puede existir un equipo de desarrollo exitoso sin la presencia de una buena comunicación entre sus miembros. La supervisión debe concentrarse menos en decirle a las personas qué hacer, y más en potenciar la comunicación para lograr que las personas lleguen a respuestas creativas por ellos mismos. El modelo de trabajo de la Programación Extrema imposibilita la existencia de una comunicación pobre.

2. Retroalimentación: para lograr el éxito es necesaria una retroalimentación constante que mantenga al equipo informado de la situación actual en la que se encuentra. Una buena retroalimentación logrará que clientes y desarrolladores avancen todo el tiempo por el camino de la victoria.

3. Simplicidad: dos de los principios más citados de la Programación Extrema son “Haga la cosa más simple que posiblemente funcione” —*Do the simplest thing that could*

possibly work— y “Usted no va a necesitarlo” —*You aren’t going to need it* (YAGNI)—. La idea fundamental que se persigue con estos principios es no malgastar esfuerzos con la adición de funcionalidad que no se necesitará en el producto planificado.

4. Coraje: si el equipo de proyecto no se mueve con la máxima velocidad, el trabajo fallará sin remedio alguno. El coraje es un arma fundamental en la Programación Extrema para hacer frente a situaciones difíciles donde una decisión puede significar el éxito o el fracaso del proyecto.

Algo realmente impresionante de XP es que entre las prácticas que propone hay poca novedad. Cada uno de sus principios ha sido utilizado y discutido durante mucho tiempo; sin embargo, puestos en conjunto, logran perfecta armonía que hacen de la Programación Extrema una disciplina organizada que ofrece una vía para incrementar velocidad y eficiencia en el desarrollo de software.

Para lograr entender qué es y cómo hacer XP, debe prestarse atención a cada una de las doce prácticas de la Programación Extrema.

Planeamiento del Juego

El Planeamiento del Juego —*The Planning Game*— es la etapa donde clientes y desarrolladores construyen un plan inicial para el desarrollo del proyecto y en la medida que este avanza, se refina hasta su culminación. El proceso de planificación puede verse como un juego, donde el cliente y los desarrolladores son los jugadores; las fichas, pequeños requisitos escritos sobre tarjetas, y una

serie de movimientos válidos que incluyen cada una de las responsabilidades de los jugadores. La planificación será realizada frecuentemente durante el proceso de desarrollo, permitirá que el equipo de trabajo perfeccione su concepción acerca del sistema, y brindará un excelente control al desarrollo de este.

Pruebas

Las Pruebas —*Testing*— son divididas en dos grupos durante la aplicación de la Programación Extrema; el primero de ellos es el que comprende la escritura de pruebas unitarias —*Test Units*—; el segundo es la escritura de pruebas de aceptación —*Acceptance Tests*—. Los desarrolladores escribirán las pruebas unitarias en la medida que escriban el código. El cliente escribirá las pruebas de aceptación mientras que defina los requisitos del sistema. Mantener un conjunto de pruebas durante la construcción del proyecto, y correrlas a intervalos regulares para validar los resultados obtenidos, asegurará mantener la calidad esperada del producto final.

Programación en Parejas

Una de las prácticas más discutidas es la Programación en Parejas —*Pair Programming*—. La producción de código en un proyecto con el empleo de Programación Extrema será realizada en parejas de desarrolladores, los cuales compartirán un mismo ordenador, monitor y teclado. Podría parecer ineficiente la utilización de dos programadores en la realización de una misma tarea, pero la realidad ha demostrado todo lo contrario. Investigaciones realizadas en este campo demuestran que dos programadores que trabajan en conjunto generan, en el mismo espacio de tiempo, códigos de mejor calidad que el producido

cuando se trabaja de forma separada.

Refactorización

La Refactorización —*Refactoring*— es el proceso de cambiar un sistema de software de modo que el comportamiento externo del código no sea alterado. En la aplicación de la Programación Extrema, los miembros del equipo de desarrollo refactorizarán el código del proyecto siempre y cuando sea necesario. El principal objetivo de la refactorización en XP es lograr un código más simple y legible, para eliminar cualquier vestigio de duplicados. La refactorización eliminará fuentes potenciales de errores y disminuirá posibles problemas a futuros desarrolladores guiándolos en la dirección correcta.

Diseño Simple

El Diseño Simple —*Simple Design*— establece que siempre deben ser realizadas las tareas de la forma más simple posible. Para lograr un diseño simple, nunca serán adicionadas características funcionales que no formen parte de la propuesta de requisitos analizada en la planificación del juego. En la Programación Extrema un buen diseño es esencial para asegurar el éxito.

Propiedad Colectiva del Código

La Propiedad Colectiva del Código —*Collective Code Ownership*— establece que cualquier miembro del equipo de desarrollo tendrá la autoridad y la capacidad de realizar cambios en el código del proyecto para lograr su mejoramiento. Cada uno de los miembros del equipo de desarrollo rotará en la implementación de cada módulo, y logrará una completa capacitación en todo el producto. Cada desarrollador estará informado de las modifica-

ciones realizadas en todas las secciones implementadas.

Integración Continua

La Integración Continua —*Continuous Integration*— estipula que deberán realizarse cada día varias integraciones del código del proyecto. Esta práctica evitará la mayor parte de los problemas que ocurren cuando un equipo integra el trabajo pasado largo tiempo, y comienzan los errores sin que nadie sepa dónde y por qué. La Integración Continua mantendrá al equipo de desarrollo moviéndose, para evitar congelamientos de código y una gran fuente potencial de errores. Una integración frecuente aumentará la posibilidad de descubrir desde el principio los errores que surjan en el proyecto y de esta forma corregirlos inmediatamente.

Cliente en el Lugar

El Cliente en el Lugar —*On-site Customer*— establece que para lograr el funcionamiento óptimo, el equipo de Programación Extrema deberá contar con el cliente a su lado durante todo el proceso de planificación para aclarar posibles dudas y tomar decisiones críticas en las reglas de negocio de la aplicación. Una comunicación cara a cara entre el cliente y el desarrollador elimina malentendidos, así como posibles cuellos de botella que retrasen el desarrollo del proyecto.

Entregas Pequeñas

Las Entregas Pequeñas —*Small Releases*— son la planificación y la entrega de versiones estables y funcionales del código del proyecto después de culminar cada iteración, de esta forma se logra una constante retroalimentación por parte del cliente que ayude a mantener al equipo de desarrollo al tanto de la actividad proyectada. La entrega de

versiones pequeñas conlleva al análisis más rápido y efectivo por parte del cliente, y evita malentendidos con los requisitos por parte de los desarrolladores. Además, podrán tomarse decisiones que no afecten o den al traste con el trabajo de varios días en caso de que el resultado no satisfaga las necesidades proyectadas.

40 Horas a la Semana

Las 40 Horas a la Semana —*40-hours Week*— establecen el tiempo promedio de trabajo del equipo de desarrollo del proyecto en 5 días hábiles. Una persona promedio, independientemente de su capacidad laboral, disminuye su rendimiento pasadas las 8 horas diarias de trabajo. Esto provoca que, paralelamente, disminuya la calidad del código producido. Una sobredosis en el tiempo de desarrollo no es la respuesta adecuada a los problemas en un proyecto, sino que es un síntoma de un problema aún más grave. La correcta dosificación del tiempo logrará que los recursos humanos se comporten a su máxima capacidad y mantengan sin irritaciones la calidad del producto final.

Estándares de Código

La aplicación de Estándares de Código —*Coding Standards*— logra un ambiente familiar en el código entre cada uno de los miembros del equipo de desarrollo. Sin la aplicación de estos será muy difícil la realización de refactorizaciones, construcción de pruebas unitarias y la comprensión por parte de otros programadores. El objetivo de esta práctica es lograr de cada pieza de código un espejo de sus vecinas en cuanto a claridad y simplicidad, sin importar la persona que la realice.

Metáfora

La Metáfora —*Metaphor*— es análoga o lo que la mayoría de las

metodologías llaman arquitectura. Brindará al equipo de desarrollo una visión común del funcionamiento del sistema a través de una evocativa y simple descripción. Si no se encuentra una variante para la construcción de la metáfora de un proyecto, el equipo de XP usará un sistema de nombres comunes para asegurarse de que cada miembro comprende el modo de funcionamiento general del sistema.

La Transición

La introducción de la Programación Extrema en nuestra empresa de desarrollo de software supone un paso de avance en materia de ingeniería, pero introduce dificultades con las que los directivos encargados de equipos de proyecto tendrán que lidiar. XP ha polarizado la industria de desarrollo de software, es difícil encontrar desarrolladores neutrales sobre temas relacionados con la Programación Extrema, debido a que muchas de sus prácticas difieren de las formas tradicionales de programación. Por ejemplo, la programación en parejas y la introducción de pruebas unitarias, en la medida en que se desarrolla el sistema, son prácticas que los desarrolladores que tratan de aplicar XP comentan las dificultades de llevarlas a cabo.

El éxito en la transición hacia el desarrollo de software guiado por la Programación Extrema se encuentra centrado en tres conceptos fundamentales: métodos, aprendizaje y dirección. El primero de ellos, se refiere a seguir los principios y estrategias de la Programación Ex-

trema. El segundo hace referencia a la necesidad de dejar a un lado los viejos conceptos y salir en busca de nuevos conocimientos que propicien el desarrollo; y el tercero resalta la necesidad de un líder en el equipo que vea el proceso como un todo y llame la atención de los miembros para impedir el desvío por caminos errados.


El hombre teme equivocarse. Cuando no conoce, prefiere mantenerse al margen; pero la equivocación es esencial para el aprendizaje. El error más grande que se comete es temer constantemente a equivocarse. Este comportamiento hace que los desarrolladores estén en constante defensiva, sin tener el coraje para probar otras alternativas y ser creativos y, de ese modo, no funciona XP. Para la utilización de XP simplemente deben aplicarse sus principios. Contar con buenos comunicadores y tener a los miembros del equipo en el lugar correcto es fundamental para la aplicación de la Programación Extrema. Con la ayuda del personal capacitado, XP debe convertirse en un gran desafío para, poco a poco, llegar a aplicar sus prácticas al más alto nivel.

Por último, y posiblemente lo más importante, contar con un equipo de personas que se relacionen bien entre sí, ayudará en gran medida a la adopción de las prácticas más difíciles de XP. La Programación Extrema es una forma social de desarrollo de software. Gran parte de su éxito se encuentra concentrado en las personas relacionadas con el producto que se pretende desarrollar. Todo esto es importante para la creación de una cultura donde los miembros del equipo no

sientan temor de ayudarse unos a otros para realizar las tareas que lleven el proyecto al éxito final.

¿Dónde encontrar más información?

Acerca de XP, varios autores han escrito diferentes artículos y libros. Estos materiales, conjuntamente con algunos sitios en Internet, son de vital importancia para la comprensión del proceso de desarrollo de software con el empleo de metodologías ágiles, específicamente, con la Programación Extrema.

Lograr un impulso en el desarrollo actual de la producción de software es responsabilidad de todo el personal involucrado en esta esfera que cada día es más importante. A menudo, las cosas pequeñas son las que logran a largo plazo grandes diferencias. 

Bibliografía

Auer, Ken; Miller, Roy. *Extreme Programming Applied*. USA: Addison-Wesley Professional, 2002.

Beck, Kent. *Extreme Programming Explained: Embrace the Change*. USA: Addison-Wesley Professional, 1999.

Collins, Chris; Miller, Roy. "XP Distilled". Disponible en: <http://www.rolemodelsoftware.com/moreaboutus/publications/xpdistilled.php>. 2001 (Consultado: 10-11-04).

Fowler, Martin. "The New Methodology" (julio, 2000). Disponible en: <http://www.martinfowler.com/articles/newmethodology.html> (Consultado: 5-11-04).

Jeffries, Ron. "What is Extreme Programming?" (noviembre, 2001). Disponible en: <http://www.xprogramming.com/xpmag/whatisxp.htm> (Consultado: 6-11-04).