

Módulo 3: Metodologías ágiles.

eXtreme Programming

Programa Transformación Digital e Industrias 4.0



Módulo 3: Metodologías ágiles.

eXtreme Programming

eXtreme Programming definida en inglés por su creador Kent Beck generando la sigla y marca registrada XP como se conoce a esta metodología ágil.

En términos generales **no existen grandes diferencias con Scrum de hecho se puede mirar a Scrum como un marco de gestión de proyectos ágiles y a XP como un conjunto de prácticas y técnicas orientadas a garantizar la calidad del resultado de los productos elaborados.**

En general se recomienda la utilización combinada de ambas metodologías en los proyectos.

En palabras del propio Kent Beck se reconocen y distinguen fundamentalmente los siguientes conceptos cómo destacados:

- Retroalimentación temprana, concreta y continua de ciclos cortos.
 - Enfoque de planificación incremental, que genera rápidamente un plan general que se espera evolucione a lo largo de la vida del proyecto.
 - Capacidad para programar de manera flexible la implementación de la funcionalidad, respondiendo a las necesidades del negocio aún cuando cambien.
 - Generación de confianza mediante la ejecución de pruebas automatizadas escritas por programadores y clientes para monitorear el progreso del desarrollo, permitir que el sistema evolucione y detectar defectos de forma temprana.
 - Basado en la comunicación oral, las pruebas y el código fuente para comunicar la estructura y la intención del sistema.
 - garantía de calidad a través de un proceso de diseño evolutivo que dura tanto como dura el sistema.
 - Estrecha colaboración de programadores con habilidades técnicas acordes a los proyectos.
- XP es una disciplina porque hay ciertas actividades y premisas que se deben garantizar y realizar en XP. Por ejemplo, no se puede elegir si se escribirán o no pruebas automatizadas; si no se hacen, no se puede considerar que se está bajo la metodología XP.

Módulo 3: Metodologías ágiles.

eXtreme Programming

módulo 3



Módulo 3: Metodologías ágiles.

eXtreme Programming

A diferencia de Scrum que plantea que no exista una división de roles dentro del equipo, eXtreme Programming reconoce la **división de capacidades de las personas de acuerdo a la capa en la que se desarrolla.**

eXtreme Programming reconoce los **siguientes roles:**

- **Programador:** El programador es el corazón de XP. Para XP el programador debe contar con habilidades que se deben destacar respecto de otros estilos de desarrollo. Algunas de estas habilidades requieren estar orientadas a las capacidades técnicas de los individuos, contando con apego a la calidad de programación, buenas prácticas y al código limpio, comprendiendo los principios de deuda técnica y ser capaz de refactorizar, para ello es importante que el programador cuente con la capacidad de escribir y de realizar pruebas unitarias del código, lo que, al igual que la refactorización, requiere gusto y juicio para que se aplique bien.

- **Cliente:** El cliente es la otra mitad de la dualidad esencial de XP. El programador sabe programar. El cliente sabe qué programar (el negocio). Ambos programadores y clientes basados en la relación empírica de la solución a desarrollar aprenden mutuamente y de forma colaborativa el uno del otro.

Al igual que los programadores, los integrantes del cliente requieren habilidades especiales para que funcione XP. Hay habilidades particulares que deben ser aprendidas antes de comenzar el proyecto cómo escribir buenas historias de usuario y contar con una actitud proactiva que hará exitoso el proyecto.

Sin embargo, sobre todo, el cliente debe sentirse cómodo influyendo en el proyecto sin poder controlarlo. Las fuerzas fuera de su control darán forma a lo que realmente se construye tanto como las decisiones que tome. Los cambios en las condiciones del negocio, la tecnología, la composición y la capacidad del equipo, todo esto tiene un gran impacto en el producto resultante (software) que se entrega.

Módulo 3: Metodologías ágiles.

eXtreme Programming

- **Tester:** Dado que gran parte de la responsabilidad de las pruebas recae sobre los hombros de los programadores, el papel de tester en un equipo de XP se centra principalmente en el cliente. El equipo es responsable de colaborar con el cliente para elegir y escribir pruebas funcionales. Si las pruebas funcionales no forman parte del paquete de integración, usted es responsable de ejecutar las pruebas funcionales regularmente y publicar los resultados en un lugar accesible por todas las personas del proyecto.

Un tester en XP no es una persona aislada, dedicada a romper el sistema y humillar a los programadores. Sin embargo, alguien tiene que ejecutar todas las pruebas con regularidad (si no puede ejecutar su unidad y las pruebas funcionales juntas), transmitir los resultados de las pruebas y asegurarse de que las herramientas de prueba funcionen bien.

- **Entrenador (Coach):** El entrenador es un rol homónimo al del Scrum Master en la metodología Scrum. El entrenador es responsable del proceso en su totalidad. Evalúa cuando las personas se están desviando del proceso y se lo comunica al equipo. Mantiene la calma cuando todos los demás entran en pánico, recordando que en las próximas dos semanas solo puede hacer el trabajo de dos semanas o menos, y el valor de dos semanas es suficiente o no lo es.

Si bien todos los integrantes de un equipo XP son responsables de comprender su aplicación, el rastreador es responsable de comprenderlo con mayor profundidad: qué prácticas alternativas podrían ayudar al conjunto de problemas actual; cómo otros equipos están usando XP; cuáles son las ideas detrás de XP; y cómo se relacionan con la situación actual.

Módulo 3: Metodologías ágiles.

eXtreme Programming

- **Consultor:** Los proyectos realizados bajo la metodología de XP en general no generan especialistas. Debido a que se promueve a que todos se emparejan con todos los demás, y las parejas flotan tanto, y cualquiera puede aceptar la responsabilidad de una tarea si así lo desea, hay pocas posibilidades de que se desarrollen huecos donde solo una o dos personas entienden exclusivamente de ese negocio o parte del sistema. Esta es una fortaleza, porque el equipo es extremadamente flexible, pero también es una debilidad, porque de vez en cuando el equipo necesita un conocimiento técnico profundo. El énfasis en la simplicidad del diseño reduce la aparición de la necesidad de un experto, pero esto sucederá de vez en cuando. Cuando esto sucede, el equipo necesita la participación de un consultor. Lo más probable es que, si el consultor, no está acostumbrado a trabajar dentro de la metodología ágil, es probable que vea lo que hace el equipo con cierto escepticismo. Para ello el equipo debe tener muy claro el problema que debe resolver el consultor.

- **Gran Jefe (Big Boss):** El rol de gran jefe sumado al del cliente es lo que en Scrum denominamos Dueño de Producto. Al igual que el dueño del producto el gran jefe debe articular entre las necesidades del negocio, las capacidades del equipo y los recursos disponibles. Al igual que el equipo el gran jefe requiere de coraje para acompañar las decisiones innovadoras propuestas por el equipo. Encarar en proyecto de desarrollo de sistemas informáticos, no pensando en la innovación en los procesos que permite la aplicación de nuevas tecnologías es un error de concepto que se comete muy frecuentemente.

Es importante la confianza y la transparencia entre todos los integrantes del equipo para asumir y provocar correctamente la aplicación de las innovaciones requeridas. Aquí es donde el rol del gran jefe se vuelve necesario para el proyecto. El equipo debe sentir su acompañamiento y comprensión en la aplicación de nuevas tecnologías para resolver su negocio

Gracias

Módulo 3: Metodologías ágiles.
eXtreme Programming

