**SCRUM**

Scrum es un proceso en el que se aplican de manera regular [un conjunto de buenas prácticas](http://www.proyectosagiles.org/fundamentos-de-scrum)para trabajar colaborativamente, en equipo, y obtener [el mejor resultado posible](http://www.proyectosagiles.org/beneficios-de-scrum) de un proyecto.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado paraproyectos en **entornos complejos**, donde se necesita **obtener resultados pronto**, donde los**requisitos son cambiantes o poco definidos**, donde la **innovación**, la **competitividad**, la**flexibilidad** y la **productividad** son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

**El proceso**

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos ([iteraciones](http://www.proyectosagiles.org/desarrollo-iterativo-incremental) de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

[Planificación de la iteración](http://www.proyectosagiles.org/planificacion-iteracion-sprint-planning)

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1. Selección de requisitos (4 horas máximo). El cliente presenta al [equipo](http://www.proyectosagiles.org/equipo-team) la lista de requisitos priorizada del producto o proyecto.
2. Planificación de la iteración (4 horas máximo). El equipo elabora la [lista de tareas de la iteración](http://www.proyectosagiles.org/lista-tareas-iteracion-sprint-backlog) necesarias para desarrollar los requisitos a que se ha comprometido.

[**Ejecución de la iteración**](http://www.proyectosagiles.org/ejecucion-iteracion-sprint)

Cada día el equipo realiza una [reunión de sincronización](http://www.proyectosagiles.org/reunion-diaria-de-sincronizacion-scrum-daily-meeting) (15 minutos máximo). Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. En la reunión cada miembro del equipo responde a tres preguntas:

* ¿Qué he hecho desde la última reunión de sincronización?
* ¿Qué voy a hacer a partir de este momento?
* ¿Qué impedimentos tengo o voy a tener?

Durante la iteración el [Facilitador](http://www.proyectosagiles.org/facilitador-scrum-master) se encarga de que el equipo pueda cumplir con su compromiso y de que no se merme su productividad.

* Elimina los obstáculos que el equipo no puede resolver por sí mismo.
* Protege al equipo de interrupciones externas que puedan afectar su compromiso o su productividad.

**Inspección y adaptación**

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

1. [Demostración](http://www.proyectosagiles.org/demostracion-requisitos-sprint-review) (4 horas máximo). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo.
2. [Retrospectiva](http://www.proyectosagiles.org/retrospectiva-sprint-retrospective) (4 horas máximo). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

**XP**

Windows XP (cuyo nombre clave inicial fue el Whistler) es una versión de [Microsoft Windows](http://es.wikipedia.org/wiki/Microsoft_Windows), línea de [sistemas operativos](http://es.wikipedia.org/wiki/Sistemas_operativos) desarrollado por [Microsoft](http://es.wikipedia.org/wiki/Microsoft). Lanzado al mercado el [25 de octubre](http://es.wikipedia.org/wiki/25_de_octubre) de [2001](http://es.wikipedia.org/wiki/2001), en diciembre de 2013, tenía una cuota de mercado de 500 millones de ordenadores. Las letras "XP" provienen de la palabra eXPeriencia (eXPerience en inglés).

Dispone de versiones para varios entornos informáticos, incluyendo para los [PC](http://es.wikipedia.org/wiki/Computadora_personal) domésticos o de negocios, además de [equipos portátiles](http://es.wikipedia.org/wiki/Computadora_port%C3%A1til), "[netbooks](http://es.wikipedia.org/wiki/Netbook" \o "Netbook)", "[tabletas](http://es.wikipedia.org/wiki/Tableta_(computadora))" y "[centros multimedia](http://es.wikipedia.org/wiki/Centro_multimedia)". Sucesor de [Windows 2000](http://es.wikipedia.org/wiki/Windows_2000) junto con [Windows ME](http://es.wikipedia.org/wiki/Windows_ME), y antecesor de [Windows Vista](http://es.wikipedia.org/wiki/Windows_Vista), es el primer sistema operativo de Microsoft orientado al consumidor que se construye con un núcleo y arquitectura de [Windows NT](http://es.wikipedia.org/wiki/Windows_NT) disponible en versiones para plataformas de [32](http://es.wikipedia.org/wiki/X86) y [64 bits](http://es.wikipedia.org/wiki/X86-64).

A diferencia de versiones anteriores de Windows, al estar basado en la [arquitectura de Windows NT](http://es.wikipedia.org/wiki/Arquitectura_de_Windows_NT) proveniente del código de [Windows 2000](http://es.wikipedia.org/wiki/Windows_2000), presenta mejoras en la estabilidad y el rendimiento. Tiene una [interfaz gráfica de usuario](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario) (GUI) perceptiblemente reajustada (denominada Luna), la cual incluye características rediseñadas, algunas de las cuales se asemejan ligeramente a otras GUI de otros sistemas operativos, cambio promovido para un uso más fácil que en las versiones anteriores. Se introdujeron nuevas capacidades de gestión de software para evitar el "[DLL Hell](http://es.wikipedia.org/wiki/DLL_Hell)" (infierno de las DLLs) que plagó las viejas versiones. Es también la primera versión de Windows que utiliza la [activación](http://es.wikipedia.org/w/index.php?title=Activaci%C3%B3n&action=edit&redlink=1) del producto para reducir la [piratería](http://es.wikipedia.org/wiki/Pirater%C3%ADa) del software.

El desarrollo de Windows XP parte desde la forma de Windows Neptune. Windows XP fue desarrollado en poco más de 18 meses, desde diciembre de 1999 hasta agosto de 2001. Windows XP fue lanzado el 25 de octubre de 2001.

Durante la década de 1990, Microsoft producía dos líneas separadas de [sistemas operativos](http://es.wikipedia.org/wiki/Sistema_operativo). Una línea estaba dirigida a las computadoras domésticas basada en un [núcleo](http://es.wikipedia.org/wiki/N%C3%BAcleo_(inform%C3%A1tica)) [MS-DOS](http://es.wikipedia.org/wiki/MS-DOS) y representada por [Windows 95](http://es.wikipedia.org/wiki/Windows_95), [Windows 98](http://es.wikipedia.org/wiki/Windows_98) y[Windows Me](http://es.wikipedia.org/wiki/Windows_ME), mientras que la otra, basada en un núcleo "[NT](http://es.wikipedia.org/wiki/Arquitectura_de_Windows_NT)" y representada por [Windows NT](http://es.wikipedia.org/wiki/Windows_NT) y [Windows 2000](http://es.wikipedia.org/wiki/Windows_2000), estaba pensada para el mercado corporativo y empresarial e incluía versiones especiales para servidores. Windows XP implicó la fusión de ambas líneas en un sistema operativo único basado enteramente en la arquitectura NT y contando con la funcionalidad y compatibilidad de la línea doméstica; con él, se eliminó definitivamente el lastre de seguridad y estabilidad que involucraba llevar el código del ya vetusto MS-DOS junto con el sistema operativo.

**Introducción a la Programación Extrema (XP)**

La programación extrema o XP es una metodología de desarrollo que se englobaría dentro de las denominadas metodologías Ágiles en la que se da máxima prioridad a la obtención de resultados y reduce la burocracia que se produce al utilizar otras ‘metodologías pesadas’.

Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.

**Kent Beck.**

El autor de la XP es Kent Beck, entre otros, que con su larga experiencia como programador eligió las mejores características de las metodologías y profundizó en las relaciones de éstas y como se reforzaban unas a otras. Por tanto, la XP no se basa en principios nuevos, sino que todas, o casi todas, sus características ya se conocen dentro de la ingeniería del software, las cuales se complementan para minimizar los tópicos problemas que pueden surgir en todo desarrollo de proyectos software.

**Objetivos de la programación extrema**

El objetivo principal de la XP es la satisfacción del cliente. Se le trata de dar al cliente lo que quiere y cuando quiere. Por tanto, se debe responder rápidamente a las necesidades del cliente, aunque realice cambios en fases avanzadas del proyecto. Como metodología Ágil que es, se pueden producir modificaciones de los requisitos del proyecto a lo largo de su desarrollo, sin que esto produzca un buen dolor de cabeza.

Otro de los objetivos es el trabajo en grupo. Tanto los jefes del proyecto, clientes y desarrolladores forman parte del equipo y deben estar involucrados en el desarrollo.

**Valores de la programación extrema**

Para garantizar el éxito de un proyecto, los autores de XP han considerado como fundamentales cuatro valores:

Comunicación. Muy importante. La XP ayuda mediante sus prácticas a la comunicación entre los integrantes del grupo de trabajo: jefes de proyecto, clientes y desarrolladores.

**Sencillez.** Los programas deben ser los más sencillos posibles y tener la funcionalidad necesaria que se indican en los requisitos. No hay que añadir algo que no se necesite hoy. Si se necesita añadir más funcionalidad mañana pues ya se hará entonces.

**Retroalimentación.** Las pruebas que se le realizan al software nos mantiene informados del grado de fiabilidad del sistema.

Valentía. Asumir retos, ser valientes ante los problemas y afrontarlos. El intentar mejorar algo que ya funciona. Aunque gracias a las pruebas unitarias no existe el riesgo de ‘meter la pata’.

Algunas voces, añaden además un quinto valor: la humildad. Con la compartición de código, la refactorización y el trabajo de equipo tan estrecho una buena dosis de humildad siempre es de agradecer.

Prácticas de la programación extrema

**12 son las prácticas de la XP:**

El juego de la planificación (the planning game). Es un permanente diálogo entre las partes empresarial (deseable) y técnica (posible).

Pequeñas entregas (small releases). Cada versión debe de ser tan pequeña como fuera posible, conteniendo los requisitos de negocios más importantes, las versiones tiene que tener sentido como un todo, me explico no puedes implementar media característica y lanzar la versión.

Es mucho mejor planificar para 1 mes o 2 que para seis meses y un año, las compañías que entregan software muy voluminoso no son capaces de hacerlo con mucha frecuencia.

Metáfora (metaphor). Una metáfora es una historia que todo el mundo puede contar a cerca de cómo funciona el sistema. Algunas veces podremos encontrar metáforas sencillas “Programa de gestión de compras, ventas, con gestión de cartera y almacén”. Las metáforas ayudan a cualquier persona a entender el objeto del programa.