

Entornos de Desarrollo

Actividad Práctica 4: Metodologías ágiles

¿Qué son las metodologías ágiles? ¿Para qué se utilizan? Exponga 3 de sus principios.

Las metodologías ágiles son un enfoque empleado para la gestión de proyectos, la optimización de procesos y especialmente en el desarrollo de software. Se centran en la flexibilidad, la colaboración y la entrega rápida de productos funcionales. Estas metodologías buscan adaptarse a cambios en los requisitos a lo largo del proyecto, en lugar de seguir un plan rígido.

Surgieron como una alternativa a las metodologías tradicionales, como el modelo en cascada (donde todas las etapas están claramente definidas y deben seguirse de manera secuencial)

Se basan en cuatro valores fundamentales:

- 1) Las personas son lo más importante, por encima de los procesos y las herramientas, por su capacidad para ser creativas e innovar.
- 2) Ver cómo se comportan las funcionalidades esperadas en software funcionando es más valioso que un documento muy detallado de requisitos.
- 3) Colaboración con el cliente por encima de negociación contractual.
- 4) Es más valiosa la capacidad de respuesta y adaptación a los cambios que la de seguir y asegurar el cumplimiento de los planes preestablecidos.

También se basan en una serie de principios. Para listar los tres primeros:

- 1) Satisfacer al cliente mediante la entrega temprana y continua: si la entrega de software se demora demasiado no satisfará las demandas de los clientes. Los usuarios cada día utilizan más software para infinidad de acciones y demandan cambios cada vez más inmediatos. No están dispuestos a esperar. Por lo que el equipo de desarrollo de software debe hacer entregas más rápidas que aporten valor a los usuarios.
- 2) Aprovechar el cambio como ventaja competitiva: En la gestión de proyectos tradicional, cambiar los requisitos al final del proyecto conllevaba aumentar el alcance y, por consiguiente, elevar los costes. Sin embargo, en los enfoques ágiles, los equipos son capaces de reconocer el valor que pueden tener estos cambios para los clientes y de responder a ellos de forma efectiva.

3) Entregar valor frecuentemente: Concretamente se refiere a la necesidad de entregar actualizaciones del software más pequeñas cada menos tiempo.

¿Cuáles son las metodologías ágiles más conocidas? Nombre alguna de ellas.

Scrum: Recibe este nombre del Rugby , está enfocada en la gestión de proyectos mediante iteraciones cortas llamadas *sprints*, con roles definidos como Scrum Master, Product Owner y el equipo de desarrollo. Promueve reuniones diarias (daily stand-ups) para revisar el progreso y sigue todos los principios del “Manifiesto Ágil”. La alegoría del Rugby se refiere al equipo entero empujando en una misma dirección.

Kanban: Basada en la visualización del flujo de trabajo a través de un tablero dividido en columnas que representan las distintas etapas del proceso. Se centra en la gestión del trabajo en curso (WIP) y la mejora continua.

Extreme Programming (XP): Orientada al desarrollo de software, enfatiza prácticas técnicas como la programación en pareja (pair programming), pruebas automáticas frecuentes y la integración continua.

Lean: Inspirada en el sistema de producción de Toyota, se enfoca en eliminar desperdicios, maximizar el valor entregado y optimizar la eficiencia del equipo.

Crystal: Una familia de metodologías ágiles enfocadas a la comunicación y la flexibilidad que varían según el tamaño del equipo y la criticidad del proyecto.

Feature-Driven Development (FDD): Enfocada en desarrollar características específicas y medibles del producto, siguiendo un proceso iterativo y planificado.

¿Qué es un sprint en Scrum?

Por así decirlo, es un ciclo de trabajo cuyas tareas, entrega y revisión vienen comprometidos dentro de la integridad del mismo, con una revisión tanto al principio como al final para asesorar posibles mejores.

¿Qué diferencias hay entre Scrum y Kanban?

En cuanto a estructura, Scrum tiene una serie de roles y eventos definidos, mientras que Kanban no. El flujo de trabajo en Kanban es continuo sin interacciones mientras que Scrum sí tiene iteraciones predefinidas con ciclos llamados “sprints” de dos a cuatro semanas. Igualmente la planificación toma una forma muy similar, donde Scrum se revisa al principio

y final de cada sprint mientras que en Kanban la revisión es continua y en función de la necesidad, al igual que las entregas. El control de trabajo en Scrum se basa en los compromisos de cada sprint, mientras que en Kanban en WIP (Work In Progress) está limitado para evitar sobrecarga en cada etapa del flujo. Scrum tiene menos flexibilidad que Kanban a la hora de realizar cambios de tarea, ya que estas vienen ya decididas en cada Sprint.

¿Cuál es la principal característica de la metodología Extreme Programming?

La principal característica de esta metodología ágil es poner más énfasis en la adaptabilidad que en la previsibilidad. El Extreme Programming considera que los cambios de requisitos sobre la marcha son acciones naturales e inevitables en el desarrollo de un proyecto.

¿Qué es el Backlog?

Es una lista priorizada de tareas, requisitos o funcionalidades pendientes por desarrollar en un proyecto. Es un elemento fundamental en metodologías ágiles (como Scrum y Kanban de las que hemos estado hablando en esta práctica) ya que ayuda a organizar y gestionar el trabajo necesario para entregar valor al cliente de manera progresiva.

Existe el “Product Backlog” (Backlog del Producto) con todas las funcionalidades, mejoras, correcciones de errores y requisitos del proyecto. Este es gestionado principalmente por el *Product Owner* y está en constante evolución según las necesidades del cliente o el mercado.

Durante los Sprint existe el “Sprint Backlog”, que incluye las tareas necesarias para completar las funcionalidades comprometidas, y es gestionado por el equipo de desarrollo.

¿Cuál es la función del product owner? ¿Debe pertenecer al equipo del proyecto?

El Product Owner o dueño del producto es el responsable de maximizar el valor del producto y asegurar que el equipo de desarrollo trabaje en lo que genera el mayor impacto para el negocio o los usuarios. Actúa como el puente entre los stakeholders (clientes, usuarios, negocio) y el equipo de desarrollo. Debido a que es básicamente el cliente final del producto, sí debe formar parte del equipo aunque con un rol diferente al de los desarrolladores, que es el de colaborar con el equipo y asegurar que las prioridades del negocio sean reflejadas correctamente en el desarrollo del producto.

Es fundamental para definir la visión del producto, gestionar el Product Backlog y asegurar la priorización del valor en cada Sprint.

Parte práctica: Jira

LOGIN 4 DE 4	CESTA COMPRA 3 DE 3	PRODUCTOS 3 DE 3 ✓
<div>Crear Usuario</div> <div><input checked="" type="checkbox"/> GDF-10 4 </div>	<div>Crear estructura de la cesta de la compra</div> <div><input checked="" type="checkbox"/> GDF-12 12 </div>	<div>Crear producto y mostrarlo a pie de página</div> <div><input checked="" type="checkbox"/> GDF-13 ✓ 15 </div>
<div>Registrar Usuario</div> <div><input checked="" type="checkbox"/> GDF-11 4 </div>	<div>Añadir y eliminar de la cesta de la compra</div> <div><input checked="" type="checkbox"/> GDF-15 12 </div>	<div>Mostrar todas las características del precio y del producto</div> <div><input checked="" type="checkbox"/> GDF-16 ✓ 15 </div>
<div>Login y Logout del usuario</div> <div><input checked="" type="checkbox"/> GDF-14 8 </div>	<div>Crear pasarelas de pago y dirección de envío</div> <div><input checked="" type="checkbox"/> GDF-18 12 </div>	<div>Crear botón de añadir al carrito ...</div> <div><input checked="" type="checkbox"/> GDF-19 ✓ 15 </div>
<div>Crear el "olvidé mi contraseña"</div> <div><input checked="" type="checkbox"/> GDF-17 8 </div>		