

MODULO 7 - Taller 1

Matías Espinoza

1. Descripción de los Fundamentos de las Metodologías Ágiles (2 puntos):

Explicar los orígenes de las metodologías ágiles y las diferencias clave con los modelos tradicionales de desarrollo como Waterfall y RUP.

Identificar las diferencias entre Agile, Agilidad y Agilismo, describiendo cómo se aplican en el contexto actual del desarrollo de software.

Tip: Explica con claridad cómo las metodologías ágiles promueven la flexibilidad y la entrega de valor continuo en comparación con modelos más rígidos.

Las metodologías ágiles surgieron como respuesta a las limitaciones de los enfoques tradicionales de desarrollo de software, como el modelo Waterfall. Durante la década de 1990, se observó que los modelos rígidos y secuenciales no eran adecuados para proyectos donde los requisitos cambiaban con frecuencia. En 2001, un grupo de 17 desarrolladores se reunieron para crear el Manifiesto Ágil, que marcó el inicio formal de Agile como metodología de desarrollo. Este manifiesto buscaba mejorar la entrega de software enfocándose en la colaboración, la flexibilidad y la entrega continua de valor.

Las metodologías ágiles difieren notablemente de los modelos tradicionales de desarrollo en varios aspectos:

- I. Waterfall (Cascada): Es un modelo secuencial donde cada fase debe completarse antes de avanzar. Los cambios son difíciles de manejar una vez que se pasa de fase. En contraste, Agile acepta que los requisitos cambien, incluso en etapas tardías del desarrollo.
- II. RUP (Rational Unified Process): Aunque es iterativo, RUP sigue un enfoque más estructurado y dirigido por casos de uso, mientras que Agile es más flexible y adaptable a los cambios. Los modelos tradicionales a menudo tardaban demasiado en entregar valor, seguían un plan rígido sin adaptarse a los cambios, generaban requisitos extensos y documentación pesada antes de empezar a desarrollar, y los equipos de desarrollo tenían poca interacción con el cliente.
- III. PMBOK: Es un enfoque que prioriza la planificación detallada y el control estricto del proyecto. Agile, en cambio, promueve una mayor flexibilidad y adaptación continua a los requisitos.

Es importante distinguir entre Agile, Agilidad y Agilismo:

- I. Agile se define de forma específica como el Manifiesto Ágil (Scrum, XP, Kanban) y los doce principios que lo acompañan. Es un marco de referencia con valores y directrices concretas para el desarrollo de software. Imagínalo como una constitución que establece las bases filosóficas.
- II. Agilidad, por otro lado, es la capacidad de adaptarse y responder al cambio de manera rápida y efectiva. Es una cualidad o una habilidad que puede tener una organización o un equipo. Las metodologías ágiles, como Scrum y Kanban, son formas de alcanzar la agilidad. Piensa en la agilidad como la meta, la capacidad de ser flexible.
- III. Agilismo se refiere al movimiento o la comunidad de personas que adoptan y promueven los valores y principios ágiles. Es la cultura, la filosofía y las prácticas

que rodean la aplicación de Agile en el desarrollo de software y otros campos. Es la adhesión a la filosofía Agile.

En el contexto actual del desarrollo de software, las metodologías ágiles promueven la flexibilidad al permitir que los equipos se adapten a los requisitos cambiantes a lo largo del proyecto, en lugar de seguir un plan rígido (como en el modelo Cascada). La entrega de valor continuo se logra a través de ciclos de desarrollo cortos (sprints) y la entrega frecuente de incrementos de software funcional. Esto asegura una entrega continua de valor al cliente y permite incorporar la retroalimentación temprana para realizar ajustes y mejoras a lo largo del proyecto. Esto contrasta con los modelos más rígidos, donde la entrega de valor al cliente puede tardar mucho más. La retroalimentación temprana y la adaptación constante son pilares fundamentales de las metodologías ágiles.

2. Valores y Principios del Manifiesto Ágil (1.5 puntos):

Identificar los 4 valores y los 12 principios del Manifiesto Ágil.

Explicar cómo estos principios guían la toma de decisiones en equipos de desarrollo ágiles.

Tip: Relaciona los principios ágiles con ejemplos concretos en proyectos reales o experiencias vividas.

El Manifiesto Ágil, creado en 2001, se basa en cuatro valores fundamentales:

- I. Individuos e interacciones sobre procesos y herramientas: Se prioriza la comunicación y la colaboración entre las personas involucradas en el proyecto.
- II. Software funcionando sobre documentación extensiva: Se da más importancia a tener un software que realmente funcione y entregue valor, que a generar grandes cantidades de documentación.
- III. Colaboración con el cliente sobre negociación contractual: Se busca una colaboración estrecha y continua con el cliente para asegurar que el producto cumpla con sus necesidades.
- IV. Respuesta al cambio sobre seguir un plan: Se valora la capacidad de adaptarse a los cambios que puedan surgir durante el proyecto, en lugar de seguir un plan predefinido rígidamente.

Estos valores se complementan con doce principios que guían la implementación de Agile y la actuación de los equipos ágiles en proyectos de desarrollo:

- I. Satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- II. Aceptar que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para el beneficio competitivo del cliente.
- III. Entregar software funcional frecuentemente (idealmente cada pocas semanas).
- IV. La colaboración cotidiana entre responsables de negocio y desarrolladores durante todo el proyecto.
- V. Construir proyectos en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en que harán el trabajo.
- VI. La conversación cara a cara como el método más eficiente y efectivo para comunicar información.
- VII. El software funcionando como la principal medida de progreso.
- VIII. Promover un ritmo de desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.

- IX. La atención a la excelencia técnica y el buen diseño permiten tomar mejores decisiones, reducir el desperdicio y evitar riesgos mejorando la agilidad.
- X. La simplicidad (el arte de maximizar la cantidad de trabajo no realizado) es esencial, es decir, elegir la opción que aporte más valor empleando el menor esfuerzo posible.
- XI. Las mejores soluciones emergen de equipos autoorganizados.
- XII. Reflexionar regularmente sobre cómo ser más efectivo y ajustar el comportamiento en consecuencia.

Estos principios guían la toma de decisiones en equipos de desarrollo ágiles al fomentar una mentalidad de priorización del cliente (principio 1), adaptabilidad (principio 2 y 11), entrega temprana y frecuente (principio 3), colaboración (principio 4 y 6), confianza en el equipo (principio 5 y 10), medición del progreso a través del software funcional (principio 7), sostenibilidad del desarrollo (principio 8), excelencia técnica (principio 9) y mejora continua (principio 12).

Ejemplo concreto: Si durante un sprint surge un nuevo requisito del cliente que se considera de alto valor, un equipo ágil, guiado por el principio de "Aceptar que los requisitos cambien", no lo descartará automáticamente por no estar en la planificación inicial. En cambio, evaluará su impacto y, en colaboración con el Product Owner, podría ajustar el alcance del sprint actual o planificarlo para un sprint futuro. Esta decisión se toma priorizando la entrega de valor al cliente por encima de la adherencia estricta a un plan predefinido.

3. Explicación del Marco de Trabajo Scrum y Roles (2 puntos):

Explicar los principios y valores de Scrum y cómo este marco de trabajo ágil ayuda a organizar el desarrollo de proyectos.

Describir los roles clave en Scrum (ScrumMaster, ScrumDeveloper, ProductOwner) y sus responsabilidades dentro de un equipo ágil.

Tip: Utiliza ejemplos que demuestren cómo los roles trabajan en conjunto en un entorno ágil.

Scrum es un marco de trabajo ágil que permite a los equipos desarrollar, entregar y mantener productos de manera eficiente mediante iteraciones cortas (sprints) y ciclos de retroalimentación continua. Se basa en la entrega incremental de valor a través de sprints que suelen durar entre 2 y 4 semanas.

Los principios y valores de Scrum se alinean con los fundamentos del Manifiesto Ágil y se centran en la colaboración, transparencia y mejora continua. Los cinco valores clave de Scrum son:

- I. Compromiso: El equipo se compromete a alcanzar los objetivos del sprint.
- II. Enfoque: Los miembros del equipo se concentran en el trabajo del sprint y sus objetivos.
- III. Apertura: El equipo mantiene una comunicación transparente sobre los desafíos y el progreso.
- IV. Respeto: Los miembros del equipo se respetan mutuamente como profesionales.
- V. Valentía: El equipo tiene la valentía de tomar decisiones difíciles y comprometerse a entregar valor.

Scrum ayuda a organizar el desarrollo de proyectos mediante la definición de roles, eventos y artefactos específicos que facilitan la planificación, ejecución, seguimiento y adaptación del trabajo. El enfoque iterativo permite entregar valor de forma temprana y frecuente, obteniendo retroalimentación que se utiliza para mejorar el producto en iteraciones futuras.

Roles Clave en Scrum y sus Responsabilidades:

- I. **Scrum Master:** Es responsable de facilitar el proceso Scrum y asegurarse de que el equipo siga los principios y prácticas ágiles. Actúa como un coach, ayudando al equipo a eliminar impedimentos, promoviendo la mejora continua y asegurando que los principios de Scrum se mantengan. Por ejemplo, el Scrum Master podría ayudar a eliminar un bloqueo técnico que impide que el equipo avance o facilitar la reunión de retrospectiva para identificar áreas de mejora.
- II. **Scrum Developers (Equipo de Desarrollo):** Son los miembros del equipo responsables de realizar el trabajo técnico. Se encargan de desarrollar, probar y entregar el incremento funcional del producto al final de cada sprint. Los developers trabajan de manera colaborativa, autoorganizada y multidisciplinar. Por ejemplo, durante un sprint, los desarrolladores trabajan juntos para diseñar, programar y probar las funcionalidades seleccionadas del Sprint Backlog.
- III. **Product Owner:** Es el responsable de maximizar el valor del producto que el equipo está desarrollando. Administra el Product Backlog y se asegura de que las tareas más importantes y que aportan mayor valor al producto se prioricen. También es el punto de contacto principal con los stakeholders. Por ejemplo, el Product Owner define las historias de usuario en el Product Backlog, las prioriza según el valor que aportan al negocio y al cliente, y aclara dudas del equipo de desarrollo sobre los requisitos.

Cómo los Roles Trabajan en Conjunto: Estos roles colaboran estrechamente durante todo el ciclo de vida del proyecto. En la Sprint Planning, el Product Owner presenta las prioridades del Product Backlog, y el Equipo de Desarrollo selecciona los elementos que se comprometen a completar en el sprint. El Scrum Master facilita esta reunión. Durante el Daily Scrum, el equipo de desarrollo comparte su progreso y los impedimentos, y el Scrum Master ayuda a eliminar estos últimos. En la Sprint Review, el equipo presenta el incremento completado al Product Owner y a los stakeholders para recibir retroalimentación. Finalmente, en la Sprint Retrospective, el equipo reflexiona sobre el sprint y planifica mejoras para el futuro, con la facilitación del Scrum Master.

4. Identificación de las Prácticas y Artefactos en Scrum (1.5 puntos):

Describir las prácticas principales de Scrum: SprintPlanning, DailyScrum, SprintReview, SprintRetrospective.

Explicar los artefactos de Scrum: ProductBacklog, SprintBacklog, Incremento, y cómo contribuyen al progreso iterativo del proyecto.

Tip: Relaciona cada artefacto con su rol dentro del ciclo de vida de un proyecto Scrum.

Scrum incluye prácticas específicas que ayudan a organizar el trabajo, mantener la transparencia y garantizar la mejora continua, las principales son:

- I. **Sprint Planning:** Es la reunión al inicio de cada sprint donde el equipo planifica el trabajo que se realizará durante ese sprint. El Product Owner presenta las prioridades del Product Backlog, y el Equipo de Desarrollo selecciona los elementos

que se comprometen a completar, creando así el Sprint Backlog. El Scrum Master facilita esta reunión.

- II. Daily Scrum: Es una reunión diaria de corta duración (no más de 15 minutos) donde el Equipo de Desarrollo sincroniza su progreso y discute posibles impedimentos. Cada miembro del equipo suele responder a tres preguntas: ¿Qué hice ayer?, ¿Qué haré hoy?, ¿Hay algún impedimento?. El objetivo es mantener a todos alineados con los objetivos del sprint.
- III. Sprint Review: Se lleva a cabo al final del sprint para presentar el Incremento completado a los stakeholders y al Product Owner. Se discute el progreso del sprint, lo que se completó y lo que no, y se recibe retroalimentación sobre el producto.
- IV. Sprint Retrospective: Es una ceremonia que se realiza al final del sprint, donde el Equipo de Desarrollo y el Scrum Master reflexionan sobre lo que funcionó bien y qué se puede mejorar. El objetivo es identificar e implementar mejoras continuas en el proceso para los siguientes sprints.

Scrum cuenta con tres artefactos principales que ayudan a organizar y visualizar el trabajo del equipo:

- I. Product Backlog: Es una lista priorizada de todas las características, funciones, mejoras y correcciones que se necesitan en el producto. El Product Owner es el responsable de mantener y priorizar este backlog para maximizar el valor del producto. Sirve como la fuente de trabajo para el equipo de desarrollo.
- II. Sprint Backlog: Es un subconjunto del Product Backlog que contiene los elementos que el Equipo de Desarrollo ha seleccionado para completar durante un sprint. Este backlog es gestionado y actualizado por el Equipo de Desarrollo durante el sprint y representa el trabajo planificado para lograr el objetivo del sprint.
- III. Incremento: Es el resultado tangible de cada sprint. Es la suma de todos los elementos del Sprint Backlog que se han completado y probado durante el sprint, y debe ser un producto funcional y potencialmente desplegable. Cada incremento se construye sobre los incrementos anteriores, contribuyendo al progreso iterativo del proyecto al entregar valor al cliente de forma continua.

Contribución al Progreso Iterativo: Las prácticas y artefactos de Scrum trabajan en conjunto para facilitar el progreso iterativo del proyecto proporcionando transparencia y un entendimiento compartido del proyecto en curso. El Product Backlog define la dirección general del producto y asegura que todos los requisitos estén documentados y priorizados, permitiendo un enfoque en las funcionalidades más valiosas. En cada Sprint Planning, se extrae un subconjunto priorizado al Sprint Backlog, enfocando el trabajo del equipo en un ciclo corto, facilitando la entrega de valor de forma regular. El Incremento al final de cada sprint representa un avance tangible hacia el producto final, permitiendo la obtención de retroalimentación temprana y frecuente de los stakeholders, lo que a su vez puede influir en la priorización del Product Backlog en futuros sprints. La Daily Scrum asegura que el equipo esté sincronizado y pueda abordar cualquier obstáculo rápidamente. Finalmente, la Sprint Review y la Sprint Retrospective proporcionan mecanismos de retroalimentación y mejora continua, permitiendo que el equipo se adapte y optimice su trabajo en cada iteración. Este ciclo iterativo de planificación, ejecución, revisión y adaptación es fundamental para el desarrollo ágil con Scrum.