

CARRERA DE COMPUTACIÓN

INFORME

1. Datos informativos

1.1.Módulo: 1

1.2.Nivel: Séptimo

1.3.Fecha: 04 de marzo de 2025

1.4.Nombres: Aupas Narvaez Cleber Antony

1.5.Tema: Metodologías: SCRUM, XP y Kanban

2. Objetivo

Comprender y analizar las metodologías ágiles Scrum, XP (Extreme Programming) y Kanban, comparando sus características, ventajas, desventajas y aplicabilidad en distintos contextos de desarrollo de software.

3. Contenido

3.1.Descripción de las metodologías

3.1.1. Scrum

Según DoneTunic (2022) menciona que: Scrum es una metodología ágil que se ve involucrada en la gestión interactiva e incremental de proyectos de desarrollo de software, donde la colaboración del equipo es muy importante y así mismo una entrega de valor continuo.

3.1.1.1. Roles principales

- **Propietario del producto:** En este aspecto se encuentra los intereses del cliente y la gestión del producto.
- **Scrum Master:** Facilitador encargado de eliminar los diferentes aspectos que se presentan y garantiza una buena aplicación de la metodología Scrum.
- **Equipo de desarrollo:** Aquí se encuentran los profesionales multifuncionales que ayudan a ejecutar de una mejor forma el trabajo

3.1.1.2. Salve de eventos

- **Reuniones de planificación del Sprint (Sprint Planning Meeting)**

Ayuda a definir el objetivo del Sprint (Sprint Goal) y el trabajo a realizar en el transcurso del sprint (Sprint Backlog).

Durante el desarrollo de Sprint Planning se debe responder dos preguntas clave:

1. ¿Qué incremento de valor puede entregarse como resultado del Sprint qué comienza?
 2. ¿Cómo se desarrollará el trabajo necesario para entregar ese incremento?
- **Responsabilidades de los diferentes roles en el Sprint Planning:**
 1. **Product Owner:** Define el Sprint Goal y el Sprint Backlog, esto quiere decir el objetivo del sprint y las actividades que se van a realizar cuando se ejecute el Sprint.
 2. **Scrum Master:** Individuo encargado de ayudar a los restantes del grupo de trabajo, dando aseguramiento de que el evento se cumpla, además que los asistentes comprendan su objetivo y que exista el cumplimiento de llevar a cabo en el tiempo indicado.
 3. **Equipo de desarrollo:** Vela en la proyección de la funcionalidad que se desarrollará en el Sprint.

- **Sprint**

Nombre que recibe cada ciclo que compone el desarrollo de un proyecto con Scrum, también llamado corazón del método Scrum. Su tiempo está entre 2 y 4 semanas, donde el Scrum Team debe trabajar para construir un crecimiento de producto “Terminado” siendo útil y desplegable.

Por otro lado, se debe tener en cuenta que durante el Sprint no se debe realizar cambios que pueda ser perjudicial al Sprint Goal. Así mismo que los objetivos de calidad no pueden decaer.

- **Scrum Diarios (Daily Scrum Meeting)**

Definida como una reunión que se la realiza diariamente y con el mismo tiempo (no debe ser superior a los 15 min), en esta cita se ven involucrados los equipos de desarrollo y puede que exista colaboración por parte del Scrum Master y el Product Owner, los cuales ayudan a agilizar el proceso de desarrollo.

- **Revisión del Sprint**

Considerada una reunión informal al final de cada Sprint donde el equipo de desarrollo con el Product Owner e personal externo, proyectan los diferentes avances logrados forma de Incremento del Producto. Aquí es donde se resaltan los objetivos alcanzados, se realiza un análisis del Product Backlog y hacer nuevos ajustes si es de hacerlo en los plazos del proyecto.

- **Retroalimentación del Sprint (Sprint retrospective)**

Sirve como un medio para realizar una inspección del Scrum Team, y así mismo proponer nuevas mejoras para aplicar a los siguientes Sprint. ¿Qué fue bien y qué fue mal en el ultimo Sprint?, y así mismo reconocer las relaciones interpersonales. Con respecto a la duración de esta fase se ve reflejado en el Sprint, si este dura 4 semanas el Sprint Restrospective debe durar 3 horas como máximo, y si es de 3 semanas debería durar 2,25 horas, etc.

3.1.1.3. Artefactos Scrum

Para Miro (2025) los artefactos Scrum son documentos e informaciones generados y tienen su uso como un aspecto importante de la metodología de trabajo Scrum.

Existen tres artefactos en Scrum

1. **Backlog de producto:** Una lista ordenada de las diferentes características del producto, los requerimientos, las respectivas mejoras y correcciones.
2. **Backlog del sprint:** Subconjunto del backlog del producto. Aquí se encuentran todos los literales donde el quipo de desarrollo planea completar en un sprint futuro.
3. **Incrementos:** Resultado acumulativo del trabajo realizado en el Sprint actual, añadiendo los diferentes avances de Sprints previos. Hace referencia a la evolución del producto.

3.1.2. XP (Extreme Programming)

Según Raeburn (2025) menciona en su publicación que: XP Es una metodología ágil caracterizada por la evolución continua de la calidad de software, adaptándose a los respectivos cambios del cliente.

3.1.2.1.Principios ordinarios

- Comunicación constante
- Existe una retroalimentación precisa
- Respeto mutuo
- Valentía para enfrentarse a los diferentes desafíos en el transcurso del desarrollo del proyecto.

3.1.2.2.Prácticas principales

- Programación en pareja
- Desarrollo dirigido por pruebas
- Integración en continuidad
- Refactorización

- Diseño simple
- Propiedad colectica del colegio

3.1.3. Kanban

Para Kanban Tool (2025) Kanban es una metodología visual que se caracteriza porque comprende al flujo de trabajo, limitando el trabajo en progreso y optimizando la eficiencia del equipo.

A continuación, se presenta una tabla donde se describe cómo se puede visualizar el flujo de trabajo:

Tabla 1. Visualización del flujo de trabajo en la Metodología Kanban

| Pasos | Descripción |
|---|---|
| 1. Identificar el flujo de valor | Corresponde a identificar y representar las etapas que constituyen al valor del producto o servicio desde su inicio hasta su fin. |
| 2. Diseñar el tablero Kanban | Una vez identificadas las etapas del flujo de trabajo, se representan en un tablero Kanban mediante columnas que reflejan cada fase del proceso. |
| 3. Utilizar tarjetas Kanban | Cada tarea o elemento de trabajo se representa con una tarjeta que se mueve a través de las columnas del tablero según su estado actual. |
| 4. Limitar el trabajo en progreso (WIP) | Establecer límites al número de tareas que pueden estar en progreso simultáneamente ayuda a evitar la sobrecarga de trabajo y garantiza que el equipo se enfoque en completar las tareas antes de iniciar nuevas. |
| 5. Gestionar y medir el flujo | La observación continua del flujo de trabajo permite identificar cuellos de botella y áreas de mejora. |
| 6. Establecer políticas explícitas | Definir claramente las reglas y criterios para cada etapa del proceso asegura que todos los miembros del equipo comprendan cómo se debe realizar el trabajo y facilita la toma de decisiones coherentes. |

| | |
|---|---|
| 7. Implementar ciclos de retroalimentación | Realizar reuniones periódicas para revisar el desempeño del equipo y del proceso permite identificar oportunidades de mejora y fomentar la colaboración y la comunicación efectiva entre los miembros del equipo. |
|---|---|

3.2. Comparativa entre las metodologías

Tabla 2. Comparativa entre las metodologías Scrum, XP (Extreme Programming) y Kanban

| Aspecto | Scrum | XP (Extreme Programming) | Kanban |
|---------------------------|--|---|--|
| Estructura | Iteraciones fijas (Sprints) | Iteraciones cortas | Flujo continuo |
| Flexibilidad | Moderada | Alta | Muy alta |
| Enfoque en calidad | Inspección y adaptación | Alto. Pruebas automatizadas, refactorización y desarrollo basado en pruebas (TDD) | Medio. La calidad depende de la gestión del flujo y de la mejora continua. |
| Adaptabilidad | Adaptable en revisiones de sprint, pero existe una limitación cuando se pretende hacer un cambio dentro de los sprint. | Muy adaptable, por el hecho de que brinda cambios y mejoras constantes. | Tiene una muy alta adaptabilidad, porque se puede hacer modificaciones en cualquier punto del proceso. |
| Ejemplos de uso | Desarrollo de software en equipos pequeños medianos o grandes con entregas iterativas. | Usada en proyectos donde la calidad del código y la flexibilidad son críticas. | Procesos de mantenimiento, soporte técnico, gestión de producción y entornos con flujo de trabajo constante. |

3.3. Aplicabilidad en escenarios reales

3.3.1. Scrum: Proyectos de desarrollo de Aplicación Bancaria

Contexto: Sistema complejo con cambios requisitos y múltiples equipos.

Razones:

- Necesidad de entregas incrementales
- Gestión de riesgos en sprints cortos
- Clara definición de roles y responsabilidades
- Transparencia en el progreso del proyecto

3.3.2. XP: Startup de Desarrollo de Software

Contexto: Equipo pequeño, producto en etapa temprana con alta innovación.

Razones:

- Énfasis en calidad técnica
- Adaptar un cambio rápido
- Mejora continua del código
- Comunicación intensa
- Programación en parejas para la transferencia de conocimiento

3.3.3. Kanban: Equipos de Soporte y Mantenimiento de TI

Contexto: Gestión de incidencias y solicitudes de servicio con variable flujo.

Razones:

- Visualización del trabajo pendiente
- Adaptación dinámica a la carga de trabajo
- Optimización del flujo de tareas
- Minimización de tiempos de espera
- No requiere reestructuración organizacional

4. Conclusiones

En resumen, las metodologías ágiles Scrum, XP y Kanban, brindan diferentes enfoques para la gestión de proyectos de software. Donde Scrum ofrece una estructura organizada basada en sprints, usada para equipos donde existe la planificación iterativa y roles bien definidos. Por otro lado, XP se basa en la calidad del código y la mejora continua por medios de las respectivas prácticas técnicas avanzadas como TDD y refactorización, donde se enfoca en productos de

software donde la excelencia técnica es fundamental para su desarrollo. Y Kanban, destaca por su flexibilidad y flujo de trabajo continuo, lo que ayuda en la gestión eficiente de actividades sin restricción de iteraciones, siendo una metodología seleccionada para proyectos de soporte técnico y mantenimiento.

Por otro lado, se puede inferir a través de la comparativa realizada entre las metodologías ágiles que cada una de ellas tiene sus ventajas y su aplicabilidad dependiendo del tipo de proyecto y del equipo de trabajo. La elección adecuada depende de los aspectos o factores, por ejemplo, la naturaleza del proyecto, el nivel de experiencia del equipo de trabajo y la necesidad de adaptarse a cambios constantes.

5. Referencias bibliográficas

DoneTunic. (2022). *¿Qué es scrum?* Obtenido de <https://donetonic.com/es/que-es-scrum/>

Kanban Tool. (2025). *Visualizar el Flujo de Trabajo*. Obtenido de <https://kanbantool.com/es/guia-kanban/fundamentos-de-kanban/visualizando-el-flujo-de-trabajo>

Miro. (2025). *Artefactos Scrum*. Obtenido de <https://miro.com/es/agile/que-son-artefactos-scrum/>

Raeburn, A. (13 de Febrero de 2025). *Asana*. Obtenido de La programación extrema (XP) produce resultados, pero ¿es la metodología adecuada para ti?: <https://asana.com/es/resources/extreme-programming-xp>