

# Parcial 1 – Metodologías ágiles

Santiago Vittori

21/05/2024

## Introducción al ejercicio

Una empresa de software funciona como una cooperativa con 30 años de antigüedad. Cuenta con una cartera de 30 clientes que utilizan diversos sistemas y soluciones consolidadas. La empresa no aplica ninguna de las metodologías de trabajo mencionadas en este curso sobre:

- Ciclo de vida del desarrollo de software en cascada,
- Cascada con testing en cada etapa,
- Xtreme Programming,
- Metodología Lean,
- TDD,
- Proceso Unificado,
- RUP, Etc.

La empresa desea modernizar su stack tecnológico para incorporar nuevas funcionalidades y mejoras a sus productos. Además, busca adquirir conocimientos en herramientas, técnicas de desarrollo e inteligencia artificial para mejorar la eficiencia y la calidad del software.

### **Estas son las principales problemáticas que se requiere solucionar:**

- Falta de planificación y organización: La empresa no utiliza metodologías de trabajo, lo que puede generar problemas de planificación, organización y seguimiento del desarrollo de software.
- Dificultad para adaptarse a los cambios: La falta de flexibilidad en los procesos de desarrollo puede dificultar la adaptación a los cambios en las necesidades del cliente o del mercado.
- Baja calidad del software: La ausencia de pruebas y control de calidad puede generar un software con errores y problemas de rendimiento.
- Dificultad para la colaboración y la comunicación: La falta de metodologías de trabajo puede dificultar la colaboración y la comunicación entre los miembros del equipo de desarrollo.
- Baja productividad: La falta de eficiencia en los procesos de desarrollo puede afectar negativamente a la productividad del equipo.
- Falta de conocimientos en herramientas, técnicas de desarrollo e inteligencia artificial: La empresa no está familiarizada con las últimas tecnologías y herramientas, lo que puede limitar su capacidad para desarrollar software moderno y eficiente.

## Ejercicios:

1. Identifique tres nuevas problemáticas de las enunciadas anteriormente
2. Para cada problemática debe indicar con cuales principios y prácticas de las metodologías se podría abordar.

Se sugiere hacer una tabla con las siguientes columnas:

- a. Problema
- b. Principios
- c. Prácticas
- d. Metodología

3. Diseñe un tablero Kanban para que la empresa lo pueda comenzar a usar. Por cada sección del tablero (o grupos de secciones) del tablero describa qué aspecto y/o problemáticas está abordando.

1.

Las nuevas problemáticas pueden ser:

1. Falta de estandarización en los procesos:

La ausencia de metodologías de trabajo estandarizadas puede resultar en una falta de consistencia en el desarrollo de software. Esto puede llevar a variaciones significativas en la calidad del producto y dificultades para mantener un nivel de calidad uniforme.

Esto se desarrolla en Lean y XP, donde se enfatiza la consistencia y la mejora continua. Sin un proceso estándar, es difícil implementar mejoras o realizar ajustes eficaces.

2. Ineficiencia en la gestión de recursos y tiempo:

Sin una metodología de gestión de proyectos, es probable que la empresa enfrente problemas para asignar recursos y gestionar el tiempo de manera eficiente. Esto puede llevar a sobrecargas de trabajo, plazos incumplidos y un uso ineficiente de los recursos disponibles.

Las metodologías vistas promueven la gestión eficiente del tiempo y los recursos a través de prácticas como la entrega incremental y la planificación iterativa. La falta de una metodología clara puede hacer que estas eficiencias no se logren.

3. Escalabilidad limitada de las soluciones desarrolladas:

Se necesita un estructurado para el desarrollo y la integración de nuevas tecnologías, las soluciones pueden no ser fácilmente escalables. Esto limita la capacidad de la empresa para expandir sus productos y servicios de manera efectiva.

La escalabilidad es un aspecto clave en metodologías como RUP y el Proceso Unificado, que enfatizan la arquitectura y el diseño robustos desde el inicio del proyecto.

2.

Tabla con los conceptos requeridos:

Problema	Principios	Prácticas	Metodología
----------	------------	-----------	-------------

Falta de estandarización en los procesos	Mejora continua (XP) Optimizar el conjunto (Lean) Calidad constante (TDD)	Reflexión (XP) Crear conocimiento (Lean) Desarrollo guiado por pruebas (TDD)	XP, Lean, TDD
Ineficiencia en la gestión de recursos y tiempo	Eliminar los desperdicios (Lean) Decidir lo más tarde posible (Lean)	Weekly cycle (XP) Slack (XP)	XP, Lean
Escalabilidad limitada de las soluciones desarrolladas	Pequeños pasos (XP) Optimizar el conjunto (Lean) Iteración e incremento (RUP)	Incremental design (XP) Build Quality In (Lean) Iterative Development (RUP) Component-Based Development (RUP)	XP, Lean, RUP
Falta de planificación y organización	Flujo continuo (Lean) Reflexión (XP)	Informative Workspace (XP) Weekly cycle (XP)	XP, Lean
Dificultad para adaptarse a los cambios	Responder al cambio sobre seguir un plan (Manifiesto Ágil) Oportunidad (XP) Ciclo rápido de retroalimentación (TDD)	Test-first programming (XP, TDD) Continuous integration (XP, TDD)	XP, TDD
Baja calidad del software	Fomentar la calidad (Lean) Calidad (XP)	Test-first programming (XP) Ten-minute build (XP)	XP, Lean
Dificultad para la colaboración y la comunicación	Comunicación (XP) Respeto a las personas (Lean)	Sit together (XP) Whole team (XP)	XP, Lean
Baja productividad	Eliminar los desperdicios (Lean) Energía en el trabajo (XP)	Energized work (XP) Slack (XP)	XP, Lean
Falta de conocimientos en herramientas, técnicas de desarrollo e inteligencia artificial	Crear conocimiento (Lean) Reflexión (XP)	Informative Workspace (XP) Stories (XP)	XP, Lean

3.

Tablero Kanban.

## Política para mover tarjetas y descripción de cada sección

### Política para mover tarjetas

#### 1. Backlog a To-do:

- Criterio: La tarjeta debe ser priorizada por el Product Owner o el equipo de gestión de proyectos.
- Acción: Mover la tarjeta a "To-do" para que esté lista para iniciar su desarrollo.

#### 2. To-do a In Progress:

- Criterio: El equipo de desarrollo debe tener capacidad disponible (respetando los límites de WIP - Work In Progress).
- Acción: Asignar la tarjeta a un miembro del equipo y moverla a "In Progress" para iniciar el trabajo.

#### 3. In Progress a Code Review:

- Criterio: El desarrollo de la tarjeta debe estar completado y cumplir con los criterios de aceptación iniciales.
- Acción: Enviar la tarjeta para revisión de código, donde otro desarrollador revisará el trabajo.

#### 4. Code Review a Testing:

- Criterio: La tarjeta debe pasar la revisión de código sin errores críticos.
- Acción: Mover la tarjeta a "Testing" para realizar pruebas funcionales y de integración.

#### 5. Testing a Ready for Deployment:

- Criterio: La tarjeta debe pasar todas las pruebas (unitarias, de integración y funcionales) y estar aprobada por el equipo de QA.
- Acción: Marcar la tarjeta como lista para despliegue.

#### 6. Ready for Deployment a Done:

- Criterio: El código debe estar desplegado en el entorno de producción y funcionando correctamente.
- Acción: Mover la tarjeta a "Done" para indicar que la tarea está completada.

### Descripción de cada sección utilizada

#### 1. Backlog:

Esta sección contiene todas las tareas, historias de usuario y mejoras propuestas que aún no han sido priorizadas para el trabajo inmediato.

2. To-do:

Aquí se colocan las tareas que han sido priorizadas y están listas para ser trabajadas. Estas tareas son las siguientes en la fila de trabajo.

3. In Progress:

Esta sección incluye las tareas que están actualmente en desarrollo. Los desarrolladores están trabajando activamente en estas tareas.

4. Code Review:

Las tareas en esta sección han completado el desarrollo y están pendientes de revisión por otro desarrollador.

5. Testing:

En esta sección se encuentran las tareas que han pasado la revisión de código y están siendo sometidas a pruebas.

6. Ready for Deployment:

Las tareas en esta sección han pasado todas las pruebas y están listas para ser desplegadas en el entorno de producción.

7. Done:

Esta sección contiene las tareas que han sido completadas y desplegadas en el entorno de producción.

### **Sistema push y pull**

- Sistema Push:

Implementado al mover tareas del Backlog a To-do. Aquí, el equipo de gestión decide qué tareas se priorizan y se empujan hacia adelante para su desarrollo.

- Sistema Pull:

Implementado desde Ready for Deployment a Done. Aquí, el equipo de desarrollo y QA tiran del trabajo según su capacidad, asegurando que no se sobrecarguen y que el flujo de trabajo se mantenga eficiente y controlado.

Link de trello:

<https://trello.com/invite/b/LJhiXIkY/ATTlab2e6953f136aacaadc5a9759feccae01C7A52B4/kanban-software>

Imagen del trello:

Backlog

Integración de Pruebas Automatizadas

Implementar Sistema de Gestión de Recursos

+ Add a card

To-Do

Configurar el entorno de pruebas de seguridad

Identificación de las necesidades de gestión de personal

+ Add a card

In Progress

Diseño del modelo de datos para el sistema.

+ Add a card

Code Review

Verificación del correcto funcionamiento del backend.

Revisión del frontend para asegurar la usabilidad.

Validar la corrección de las vulnerabilidades.

+ Add a card

Testing

Validar la integración con otros módulos de gestión de recursos humanos.

Validar la integración con otros módulos de gestión de recursos humanos.

+ Add a card

Ready for Deployment

Generar el paquete de despliegue.

Documentar el proceso de implementación y uso del módulo.

+ Add a card

Done

Módulo de seguimiento de asistencias implementado y desplegado.

Sistema de pruebas de seguridad integrado y funcionando.

+ Add a card