

## 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.

## Tiempo estimado de resolución: 2hs

## Formato de entrega del ejercicio

El alumno lo puede entregar el examen en los siguientes formatos:

- a. en formato digital a través del repositorio en la carpeta examen/parcial
- b. en formato papel

## Respuestas:

### 1. Otras problemáticas:

1. **Poca retroalimentación con el cliente:** considerando que no se aplican las metodologías dadas en el curso y que la empresa tiene dificultades para adaptarse a los cambios, puedo suponer que la empresa no acostumbre ir haciendo entregas testeables del sistema al cliente para obtener una retroalimentación y así saber si el proyecto está yendo en la dirección correcta.
2. **Baja calidad del software:** Si bien este punto ya se mencionó antes, no se aclaró que la calidad del software se justifica por su apartado técnico, como el rendimiento y buena construcción, y por su apartado funcional. Con la falta de retroalimentación o feedback del cliente y las demás problemáticas de la empresa, probablemente las necesidades que el cliente tenga, por las cuales necesita el software, no sean del todo satisfechas. Es decir, que el software no hace lo que debería de la forma en que debería, no es funcional.
3. **Mala composición de equipos de trabajo:** Al no tener capacidad de análisis, planificación ni adaptabilidad, probablemente para la mayoría de proyectos el personal designado sea siempre el mismo, aunque lo mejor quizá sería trabajar con personal especializado en las tecnologías que se necesitan implementar y/o que tengan noción del contexto del cliente.
4. **Roles poco claros:** no aplicar metodologías de trabajo deja la puerta abierta a que no se tengan claros o definidos los roles de cada integrante. Que cada participante del proyecto no comprenda del todo cuál es su papel puede acarrear consecuencias que afecten negativamente al proyecto y a la moral del grupo.

## 2.

	Problema	Principios	Prácticas	Metodología
1	Falta de planificación y organización	Pequeños pasos	Definir qué es lo menos que se puede hacer y que vaya en la dirección correcta, Implementación de estructuras y métodos de organización como Kanban, Armar y planificar el proyecto con las historias de usuario ordenadas por prioridad	XP, Lean
2	Dificultad para adaptarse a los cambios	Entregar rápido Documentación liviana	Trabajar en iteraciones con productos entregables al final de cada una que el cliente pueda probar y hacer ajustes en base al feedback, No documentar con detalles exhaustivos	XP, Lean
3	Baja calidad del software (técnica)	Fomentar la calidad	Pruebas unitarias, Refactorización, Retroalimentación, TDD	XP, Lean
4	Dificultad para la colaboración y la comunicación	Respetar a la gente y potenciar el equipo	Reuniones regulares, Herramientas de colaboración	XP, Lean
5	Baja productividad	Eliminar desperdicios	Identificación y eliminación de actividades no productivas	Lean
6	Falta de conocimientos en herramientas, técnicas de desarrollo e inteligencia artificial	Crear conocimiento	Formación continua, Pruebas de concepto (probar herramientas en una iteración) Aprender de lo que funciona y de lo que no	XP, Lean
7	Poca retroalimentación con el cliente	Entregar rápido	Trabajar en ciclos o iteraciones donde al final de cada una se obtenga una versión entregable del producto que el cliente pueda probar	Lean
8	Baja calidad del software (funcional)	Entregar rápido Fomentar la calidad	Trabajar en ciclos o iteraciones donde al final de cada una se obtenga una versión entregable del producto que el cliente pueda probar, Pruebas de integridad conceptual	XP, Lean
9	Mala composición de equipos de trabajo	Respetar a la gente y potenciar el equipo	Establecer objetivos comunes, Trabajar en equipo, Definir equipos y roles dentro de los equipos	Lean
10	Roles poco claros			

3. El flujo del tablero es de arriba hacia abajo de izquierda a derecha:
1. **Descripción del proyecto:** Información general del proyecto. Entender en lo que se está trabajando, con quién y para qué une al grupo bajo un objetivo común y ya va perfilando en la cabeza de cada integrante la dirección del proyecto. Sería útil colocar aquí información de reuniones de equipo para hacer retrospectiva sobre lo trabajado y con eso mejorar el desempeño.
  2. **Análisis:** Se deben obtener del cliente las funcionalidades del sistema, teniendo en cuenta el contexto. Estas funcionalidades pueden escribirse como historias de usuarios y deben ser categorizadas y ordenadas por su relevancia.
    - i. **Secundario:** Requisitos de baja prioridad que pueden llegar a estar o no en el producto final dependiendo del tiempo
    - ii. **Prioridad:** Requisitos que deberán llegar a estar en el producto final. Deben estar a su vez ordenados por alguna cualidad definida como Dificultad, Relevancia o Dependencia.
  3. **Desarrollo:** Requisitos en los que se está trabajando en la iteración actual
    - i. **Pendiente:** Requisitos pendientes
    - ii. **En desarrollo:** Requisitos en progreso junto con los colaboradores asignados
  4. **Testing:** Comprobación de calidad. En caso de desaprobarse se debe devolver la funcionalidad al área de análisis y seguir desde allí. En caso de aprobarse se continúa.
    - i. **Técnica:** El prototipo debe pasar test unitarios y de integración. Luego el funcional.
    - ii. **Funcional:** El prototipo debe ser usado por el cliente, indicándole qué funcionalidades y/o aspectos evaluar.
  5. **Producto obtenido:** Conjunto de funcionalidades terminadas

Este tablero:

- **Organiza el trabajo:** deja clara las distintas fases del desarrollo
- **Mejora la calidad técnica y funcional:** el software está bien construido y además cumple su propósito funcional. Esto es gracias a las pruebas unitarias, pruebas de integración y feedback del cliente.
- **Mejora la comunicación y transparencia:** deja ver rápidamente el estado del proyecto, su rumbo y los papeles que juegan cada uno de los colaboradores.
- **Aumenta la percepción de calidad del cliente:** Al involucrar al cliente, el mismo valora más cada avance y siente que el producto se hizo a su medida.
- **Espacio para aprender:** Cuando los requisitos que no aprobaron los test vuelven a la etapa de análisis es el momento de entender por qué no funcionó ¿error en el relevamiento? ¿Error técnico? ¿De comunicación?. De esos errores se aprende y el conocimiento se comparte en las reuniones que se deben establecer en el tablero de Información general. Además si se trabaja en iteraciones se puede decidir experimentar con alguna tecnología o metodología de trabajo en alguna/s iteración.
- **Mejora de productividad:** Esto es gracias a lo mencionado en el punto anterior. Evaluar y compartir el conocimiento acerca de qué cosas no funcionan hace que se eliminen prácticas negativas, dejando espacio a nuevas o ya conocidas prácticas que mejoren el proceso de desarrollo.

Proyecto: Nombre del proyecto

Resumen del proyecto

Cliente: Nombre de cliente

Equipo designado: Integrantes del proyecto y rol

Fecha de inicio: x

Fecha tentativa de finalización: x

Análisis: Requisitos pendientes

Secundario

Historia de usuario

Historia de usuario

Historia de usuario

Prioridad

Historia de usuario

Historia de usuario

Historia de usuario

Desarrollo

Pendiente

Historia de usuario

Historia de usuario

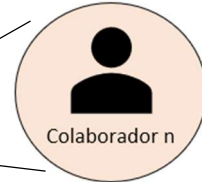
Historia de usuario

En desarrollo: Requerimiento y colaboradores asignados

Historia de usuario

Historia de usuario

Historia de usuario



Testing

Técnico: Por desarrolladores

Pendientes de revisión

Desaprobado

Aprobado

Funcional: Por el cliente

Pendientes de revisión

Desaprobado

Aprobado

Producto obtenido:

Historia de usuario

Historia de usuario

Historia de usuario

Historia de usuario

Historia de usuario

Historia de usuario