

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.
*La falta de planificación y organización también puede causar que el cliente cuando logra ver algo tangible, el producto final, al finalizar el desarrollo y no durante, encuentre requerimientos mal implementados o surjan nuevos. Por lo que es recomendable hacer pequeñas entregas de paquetes de funcionalidades

donde el cliente lo puede apreciar, entender y analizar si está correcto, si es lo que quiere.

*Otro problema es que al no aplicar ninguna metodología, está debe estar especificando tiempo de entrega y precio exacto, y no acordando tiempos y costos por paquete de funcionalidades. Esta problemática puede causar que el proyecto no llegue a terminarse en el tiempo acordado, por lo que el cliente no puede pagar por lo acordado, y ambos pierden, ya que la empresa no cobra, pierde dinero y el cliente después de trabajar por un largo tiempo con la empresa al final no obtiene nada. Ambos pierden tiempo y dinero.

*Otra de las problemáticas es que cuenta con 30 clientes que utilizan diversos sistemas y soluciones consolidadas, y dudo que exista un control de actualización o control de versiones de cada sistema.

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

Problema	Principios	Práctica	Metodología
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.	<ul style="list-style-type: none"> * Mejora continua *Automatización *Crear conocimiento 	<ul style="list-style-type: none"> *Incorporando herramientas de integración continua. *Automatización de los procesos de construcción y despliegue del software. *Incorporando representaciones gráficas de la evolución del desarrollo, para medirlo. También organización mediante gestión visual (ej Kanban). 	Learn Xp
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.	<ul style="list-style-type: none"> *Beneficio mutuo *Calidad del código *Pequeños pasos *Aprender rápido y responder rápido *Colaboración con el cliente 	<ul style="list-style-type: none"> *Pair Programming: Programación en pareja. *Revisión de código *Uso de herramientas de análisis estático del código 	Xp TDD Learn
Baja calidad del software: La ausencia de pruebas y control de calidad puede generar un software con errores y problemas de	<ul style="list-style-type: none"> *Mejora *Calidad *Fallo 	<ul style="list-style-type: none"> *Pruebas continuas, los errores se detecten y corrijan de inmediato, promoviendo prácticas 	TDD Xp

rendimiento.		estrictas de calidad *Programación en pareja e integración continua. *Eliminación de defectos	
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.	*Responsabilidad aceptada *Beneficio mutuo *Individuos e interacción *Respetar a la gente y potenciar el equipo *Reflexión	*Busca prácticas que beneficien a todos los involucrados en el proyecto. *Fomentar la comunicación. *Determinar roles y tareas pro rol. *Reuniones semanales de avance.	Xp Manifiesto Ágil
Baja productividad: La falta de eficiencia en los procesos de desarrollo puede afectar negativamente a la productividad del equipo.	*Responsabilidad aceptada *Individuos e interacción *Humanidad *Reflexión	*Automotivarse. *Programación en pareja.	Xp
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.	*Pequeños pasos *Eliminar desperdicios *Calidad de código *Auto-similitud *Menor Tasa de Errores *Simplicidad	*Quitarse o viejo, lo que no funciona correctamente. *Implementar separación por módulos o funcionalidades. *Incorporar test. *Aplica patrones	Learn TDD *Xp
La falta de planificación y organización también puede causar que el cliente cuando logra ver algo tangible, el producto final, al finalizar el desarrollo y no durante, encuentre requerimientos mal implementados o surjan nuevos. Por lo que es recomendable hacer pequeñas entregas de paquetes de funcionalidades donde el cliente lo puede apreciar, entender y analizar si está correcto, si es lo que quiere	*Software que funciona *Calidad de código	*Entregas del software funcionando, en intervalos de unas pocas semanas a unos pocos meses, con preferencia en intervalos cortos. *Revisión de código	Manifiesto Ágil Xp
Otro problema es que al no aplicar ninguna metodología, está debe estar especificando tiempo de entrega y precio exacto, y no acordando tiempos y costos por paquete de funcionalidades. Esta	*Software que funciona *Entregar rápido *Flujo *Colaboración con el cliente	*Entregas del software funcionando, en intervalos de unas pocas semanas a unos pocos meses, con preferencia en intervalos cortos.	Manifiesto Ágil TDD Learn

problemática puede causar que el proyecto no llegue a terminarse en el tiempo acordado, por lo que el cliente no puede pagar por lo acordado, y ambos pierden, ya que la empresa no cobra, pierde dinero y el cliente después de trabajar por un largo tiempo con la empresa al final no obtiene nada. Ambos pierden tiempo y dinero.	*Entrega frecuente de software funcional	*Revisión de código	
Otra de las problemáticas es que cuenta con 30 clientes que utilizan diversos sistemas y soluciones consolidadas, y dudo que exista un control de actualización o control de versiones de cada sistema	*Redundancia *Desarrollo sostenible	*Implementar un sistema de gestión de versiones como github	Xp Desarrollo ágil

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.

	Tareas pendientes	Tareas por hacer	Tareas en proceso	Pruebas y testing	Terminadas	Deploy
Descripción	Es la lista de tareas pendientes importantes, están separadas por funcionalidad y módulo y tienen orden de prioridad estos módulos y las tareas funcionalidades también. Se deben priorizar. Pasarán a la siguiente sección cuando se esté terminando el módulo o funcionalidad anterior.	Tareas que están listas para ser trabajadas en el próximo ciclo. Es decir pasan todas las tareas pendientes de la sección anterior, que pertenezcan al mismo módulo, pasarán todas las tareas por funcionalidad.	Tareas que están actualmente siendo trabajadas. Tareas de la sección anterior que están siendo acatadas, desarrolladas.	Tareas que han sido completadas y están en fase de revisión o pruebas. Cuando pasen todas las pruebas podrán pasar a la siguiente sección,	Tareas que han sido completadas y aprobadas. Tareas que pasaron las pruebas y los test. Cuando se terminen todas las tareas de la funcionalidad pasarán a la siguiente sección para subirse a productivo.	Tareas que se subieron a productivo, por funcionalidad.
Problemática que	*Falta de planificación y organización *Dificultad para	*Falta de planificación y organización, mejora en la	*Falta de planificación y organización. *Mejor control y	*La falta de planificación y organización	*	*Inexistencia de control de actualización

aborda	adaptarse a los cambios.	visibilidad del trabajo pendiente. *Organización en equipo. *Dificultad para adaptarse a los cambios	seguimiento del desarrollo en curso. *Dificultad para adaptarse a los cambios. *Dificultad para la colaboración y la comunicación.	también puede causar que el cliente cuando logra ver algo tangible, el producto final, al finalizar el desarrollo y no durante, encuentre requerimientos mal implementados o surjan nuevos. *Baja calidad del software.		n o control de versiones de cada sistema. *Al no aplicar ninguna metodología, está debe estar especificando tiempo de entrega y precio exacto, y no acordando tiempos y costos por paquete de funcionalidades. *La falta de planificación y organización también puede causar que el cliente cuando logra ver algo tangible, el producto final, al finalizar el desarrollo y no durante, encuentre requerimientos mal implementados o surjan nuevos. *Dificultad para adaptarse a los cambios.
--------	--------------------------	--	---	--	--	---

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