

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

1.

- **Complejidad del sistema:** la complejidad del sistema interfiere en la comunicación entre componente, es decir, no tener el software subdivididos en pequeños módulos interrelacionados hace que aumente su complejidad.
- **Invisibilidad del progreso:** no se tiene una transparencia en el progreso del proceso, lo que relaciona también la comunicación en un equipo que lleve a cabo un proyecto de software.
- **Conformidad a estándares y regulaciones:** llevar a cambio una revisión de cumplimiento periódica durante las iteraciones, para tener un mayor control y enfoque de lo que se espera en cada una de ellas.

2.

Problema	Principios	Prácticas	Metodología
Falta de planificación y organización	Iteración y adaptación	Planificación de iteraciones	XP
Dificultad para adaptarse a los cambios	Feedback continuo	Revisión y adaptación por cada iteración	Agil
Baja calidad del software	Mejora e integracion continua y desarrollo iterativo	Pruebas automatizadas - ejemplo: unitarias, de integración, etc. Integración continua Revisiones de código	XP, Agil y TDD
Dificultad para la colaboración y la comunicación	Transparencia, colaboración, equipos auto-organizados	Reuniones diarias Planificación de iteraciones	XP y Lean
Baja productividad	Eficiencia, eliminación de desperdicios, iteración rápida.	Visualización del flujo de trabajo	Lean
Falta de conocimientos en herramientas, técnicas de desarrollo e inteligencia artificial	Aprendizaje continuo, innovación	Capacitación continua, investigación y desarrollo, uso de herramientas modernas	XP y Lean
Complejidad del sistema	Simplicidad, modularidad	Descomposición del sistema en módulos independientes, uso de patrones de diseño.	RUP
Invisibilidad del progreso	Transparencia y comunicación.	Ciclos semanales y trimestrales de reuniones, diseño incremental	XP
Conformidad a estándares y regulaciones	Cumplimiento, responsabilidad.	Revisión de cumplimiento durante las iteraciones	RUP

3.

Pendientes	Tareas en progreso	Revision	Prueba	Espera	Tareas finalizadas
Falta de planificación y organización					
	Baja productividad				
		Baja calidad del software	Baja calidad del software		
				Dificultad para adaptarse a los cambios	
					Invisibilidad del progreso

Pendientes:

Falta de planificación y organización: al tener una lista clara y visible de todas las tareas pendientes, se facilita la planificación y organización del trabajo futuro. Todos los miembros del equipo pueden acceder a las mismas, lo que mejora la comunicación y la colaboración. La empresa, debería de planificar y organizar lo que pretende llevar a cabo

Tareas en progreso:

Baja productividad: al limitar la cantidad de tareas que pueden estar en progreso al mismo tiempo, se reduce la multitarea y se mejora la eficiencia, permitiendo una organización en iteraciones más pequeñas para llevar a cabo.

Revisión:

Baja calidad del software: garantiza que todas las tareas sean revisadas antes de considerarlas completas, lo que ayuda a mantener la calidad. A su vez, se tiene en cuenta el feedback continuo, que facilita la recepción de feedback temprano y continuo, permitiendo ajustes rápidos y mejora constante.

Prueba:

Baja calidad del software: al realizar pruebas exhaustivas, se asegura que el software cumple con los requisitos y no introduce errores, lo que facilita la integración continua y asegura que las mejoras se prueben antes de ser implementadas en el ámbito productivo.

Espera:

Dificultad para adaptarse a los cambios: visualiza claramente los bloqueos y dependencias, lo que implica que el no se logró una adaptación esperada del sistema, esto, permite al equipo reaccionar y buscar una nueva alternativa para adaptarse, previniendo errores en un tiempo ya avanzado del proyecto.

Tarea Finalizada:

Invisibilidad del progreso: se determina que la invisibilidad fue trabajada de alguna forma, ejemplo en la modularización del sistema, esta sección proporciona una visión clara de qué tareas están listas para ser entregadas, mejorando la visibilidad del progreso del proyecto. Facilita la comunicación con los stakeholders al mostrar claramente qué trabajo está listo para ser lanzado.