TALLER 1

PARCIAL No. 1

METODOLOGIAS DE DESARROLLO EN SOFTWARE

METODOLOGIAS DE DESARROLLO EN SOFTWARE
ÁREA DE COMPUTACION
DEPARTAMENTO DE ITIN

TALLER.

Nombres de los estudiante:	Cristian Jesus Becerra Loaiza
	Jhon Kevin Castillo Quishpe
	Ismael Alejandro Silva Flores

Nivel:	Tercer Nivel
NRC:	29022
Asignatura:	Met. de Desarrollo en Software
Nombre del profesor:	Ing. Jenny Alexandra Ruiz Robalino

METODOLOGIAS DE DESARROLLO EN SOFTWARE
ÁREA DE COMPUTACION
DEPARTAMENTO DE ITIN

Contenido

TALLETR 1

Primera parte

Elaborar una tabla comparativa entre metodologías tradicionales y ágiles considerando al menos cinco criterios: estructura, documentación, flexibilidad, rol del cliente, entregas.

lentificar en qué tipo de proyectos consideran más adecuado cada enfoque. Justifiquen.

Tabla de comparación entre metodologías tradicionales y agiles.

Criterio	Metodologías Tradicionales	Metodologías Ágiles
Estructura del proceso	-El modelo en cascada mantiene actividades secuenciales que parten desde la definición de requisitos hasta la operación y el mantenimiento. (p.153, Ruiz Robalino & Lucio Moreno) -Ocupa las actividades importantes del proceso… luego los representa como fases separadas. (p.45, Ruiz Robalino & Lucio Moreno) -El proceso unificado racional RUP… presenta las fases de este modelo a través de una línea de tiempo. (p.117, Ruiz Robalino & Lucio Moreno) -Cada iteración consta de cuatro fases: concepción, elaboración, construcción y transición. (p.120, Ruiz Robalino & Lucio Moreno) -El ciclo de vida en cascada… entrega un producto software funcional. (p.153, Ruiz Robalino & Lucio Moreno)	-SCRUM es un método ágil general, se basa en la administración iterativa del desarrollo de un producto software. (p.131, Ruiz Robalino & Lucio Moreno) -El producto software se desarrolla y se entrega en incrementos. (p.129, Ruiz Robalino & Lucio Moreno) -La programación extrema XP… ocurre en el marco de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. (p.131, Ruiz Robalino & Lucio Moreno) -Cada ciclo desarrolla un prototipo funcional para el usuario. (p.156, Ruiz Robalino & Lucio Moreno) -El desarrollo basado en un plan… se realiza como una serie de ciclos cortos llamados Sprint. (p.134, Ruiz Robalino & Lucio Moreno)

Documentación	-Cada fase genera documentos específicos como manuales de instalación y usuario. (p.119, Ruiz Robalino & Lucio Moreno) -El documento de especificación de requisitos (ERS) tiene que definir con exactitud lo que se implementará. (p.59, Ruiz Robalino & Lucio Moreno) -La validación del software examina el software para asegurar el cumplimiento de requerimientos. (p.45, Ruiz Robalino & Lucio Moreno)	Se minimizan así la documentación del proceso y la burocracia. El foco de desarrollo está en el código. (p.111, Ruiz Robalino & Lucio Moreno) -Los requisitos se expresan como escenarios o historias de usuario. (p.131, Ruiz Robalino & Lucio Moreno) -La validación se realiza mediante retroalimentación continua. (p.164, Ruiz Robalino & Lucio Moreno) -La documentación se adapta según el feedback del cliente. (p.135, Ruiz Robalino & Lucio Moreno)
Flexibilidad	-El modelo en cascada… entrega un producto software funcional al final del proceso. (p.153, Ruiz Robalino & Lucio Moreno) -Cambiar requisitos implica rediseñar fases completas. (p.45, Ruiz Robalino & Lucio Moreno) -La especificación del software define la funcionalidad y las restricciones. (p.45, Ruiz Robalino & Lucio Moreno) -La evolución del software es una característica de flexibilidad… pero limitada. (p.45, Ruiz Robalino & Lucio Moreno) -El proceso es más rígido y menos tolerante a cambios. (p.117, Ruiz Robalino & Lucio Moreno)	-La adaptación a los cambios los aborda desde la perspectiva de la mejora continua. (p.153, Ruiz Robalino & Lucio Moreno) -SCRUM… permite incorporar cambios en cada Sprint. (p.131, Ruiz Robalino & Lucio Moreno) -XP prioriza los cambios que se generan a nivel de requisitos funcionales. (p.129, Ruiz Robalino & Lucio Moreno) -El desarrollo ágil permite ajustar el trabajo en cada iteración. (p.134, Ruiz Robalino & Lucio Moreno) -La alta volatilidad de los requisitos… es enfrentada con eficiencia. (p.153, Ruiz Robalino & Lucio Moreno)
Rol del cliente	La validación del software	-Los clientes deben

	examina el software para asegurar el cumplimiento de requerimientos establecidos por el cliente. (p.45, Ruiz Robalino & Lucio Moreno) -El cliente participa al inicio y al final del proceso. (p.117, Ruiz Robalino & Lucio Moreno) -Los requisitos se aprueban antes de iniciar el diseño. (p.45, Ruiz Robalino & Lucio Moreno) -La intervención del cliente es limitada durante el desarrollo. (p.153, Ruiz Robalino & Lucio Moreno) -El cliente valida el producto al final del ciclo. (p.119, Ruiz Robalino & Lucio Moreno)	intervenir conjuntamente con el equipo de desarrollo durante el proceso. (p.135, Ruiz Robalino & Lucio Moreno) -La colaboración con el cliente debe prevalecer sobre la negociación de contratos. (p.114, Ruiz Robalino & Lucio Moreno) -El cliente participa en la valoración de cada Sprint. (p.156, Ruiz Robalino & Lucio Moreno) -La retroalimentación del cliente se realiza en cada iteración. (p.164, Ruiz Robalino & Lucio Moreno) -El cliente ayuda a priorizar los requisitos funcionales. (p.135, Ruiz Robalino & Lucio Moreno)
Entregas	-Al terminarla, se entrega un producto software funcional, así como la documentación para el usuario. (p.119, Ruiz Robalino & Lucio Moreno) -La entrega se realiza al final del ciclo de vida. (p.153, Ruiz Robalino & Lucio Moreno) -No se valida hasta el cierre del proyecto. (p.45, Ruiz Robalino & Lucio Moreno) -Cada fase culmina en un entregable único. (p.117, Ruiz Robalino & Lucio Moreno) -El producto se entrega completo y probado. (p.119, Ruiz Robalino &	-El producto software se desarrolla y se entrega en incrementos… cada uno de los cuales termina construyendo una nueva funcionalidad. (p.129, Ruiz Robalino & Lucio Moreno) -Cada Sprint entrega un prototipo funcional. (p.156, Ruiz Robalino & Lucio Moreno) -La validación se realiza en cada ciclo. (p.164, Ruiz Robalino & Lucio Moreno) -Permite correcciones tempranas y mejora continua. (p.135, Ruiz Robalino & Lucio Moreno) -El cliente aprueba cada entrega según sus

	Lucio Moreno)	necesidades. (p.156, Ruiz Robalino & Lucio Moreno)

Resumen de tabla:

Criterio	**Metodologías Tradicionales**	**Metodologías Ágiles**
Estructura del proceso	Secuencial, por fases fijas (análisis → diseño → desarrollo → pruebas).	Iterativa e incremental, con ciclos cortos (Sprints o iteraciones).
Documentación	Extensa y detallada en cada fase (ERS, manuales, validaciones formales).	Mínima, centrada en el código y ajustada según feedback del cliente.
Flexibilidad	Rígida, difícil de adaptar a cambios una vez definidos los requisitos.	Alta, permite cambios frecuentes y mejora continua en cada iteración.
Rol del cliente	Participa al inicio y al final; intervención limitada durante el desarrollo.	Participación constante; colabora en cada ciclo y prioriza requisitos.
Entregas	Una sola entrega final del producto completo y probado.	Entregas frecuentes de funcionalidades parciales y validadas por el cliente.

Segunda parte.

Una empresa desarrolla un sistema para gestionar reservas de vuelos. El cliente tiene ideas generales, pero cambiará varios requisitos durante el proceso. Se necesita entregar funcionalidades cada pocas semanas.

¿Qué metodología elegirían para este caso?

En este caso nosotros recomendamos usar una metodología ágil, como Scrum o Kanban (Solis, 2024).

¿Por qué?

- Cambios Frecuentes de Requisitos: Las metodologías ágiles aceptan de manera muy fácil los cambios y en lugar de ver nuevos requisitos como un problema, los incluyen en las próximas iteraciones, llamadas "Sprints" en Scrum ya que en un método tradicional trataría de mantener los requisitos iguales desde el principio, pero esto no funciona aquí (Pohl, 2015).
- Entregas Rápidas y por etapas: Es fundamental entregar funciones cada pocas semanas. Cada ciclo corto (de 2 a 4 semanas) termina con una versión lista del software que aporta valor al cliente (Acosta, 2016).
- Ideas del Cliente: Agil no necesita una lista de requisitos que sea perfecta desde el comienzo en cambio trabaja con una lista priorizada de funciones deseadas (llamada "Backlog") que se mejora y ajusta según el feedback del cliente (Pressman, 2005).
- Feedback constante y menor riesgo: Con entregas frecuentes el cliente puede ver cuál es el progreso y dar su opinión. Así, el equipo evita desviarse y asegura que el producto final sea lo que el cliente esperaba recibir, reduciendo mucho el riesgo de hacer algo equivocado (Piattini, 2016).

¿Qué ventajas y desventajas tendrían si eligieran un enfoque tradicional en lugar de ágil?

Si se usara un método tradicional, como el Modelo en Cascada, el proyecto tendría estos problemas:

Desventajas:

- No se puede cambiar fácilmente: El método Cascada sigue pasos en orden y es rígido. Si el cliente quiere cambiar un requisito después de la fase de diseño, es muy caro y lleva mucho tiempo hacer ese cambio. No está hecho para adaptarse rápidamente (Pressman, 2005).
- Entregas tardías: El cliente no vería un producto completo hasta las etapas finales (pruebas e implementación). Pasarían meses antes de que el cliente pudiera ver y dar su opinión, y quizás sea demasiado tarde para hacer cambios importantes (Somerville, 2011).
- Alta probabilidad de error: Los requisitos iniciales suelen ser solo ideas generales, y muchas veces están incompletos o mal. Construir todo sobre esas ideas puede dar un producto que no satisface lo que necesita el usuario final (Pressman, 2005).

Ventajas posibles:

- Muchos documentos: Los métodos tradicionales suelen generar muchos papeles y detalles, útiles en sistemas muy importantes, como en medicina o aviones. Pero para un sistema de reservas, esto sería muy lento y pesado (IEEE Computer, 2014).
- Presupuesto y tiempos previsibles (en teoría): Si los requisitos no cambian, el método Cascada permite estimar costos y fechas al principio. Pero en nuestro caso, eso no se cumple (Pressman, 2005).

Referencias

Acosta, M. (2016). *Software Enginnering - Ian Sommerville*. Obtenido de Software Enginnering - Ian Sommerville: https://es.slideshare.net/MatiasGonzaloAcosta/procesos-de-software-unidad-2-software-enginnering-ian-sommerville

IEEE Computer, S. (2014). *Guide to the Software Engineering Body of Knowle dge Version 3.0 (SWEBOK Guide V3.0.* . Obtenido de Guide to the Software Engineering Body of Knowle dge Version 3.0 (SWEBOK Guide V3.0. : https://www.academia.edu/8583868/Guide_to_the_Software_Engineering_Body_of_Knowled ge_Version_3_0_SWEBOK_Guide_V3_0

Jenny Alexandra Ruiz Robalino y Xavier Iván Lucio Moreno. (s.f.). Fundamentos de ingeniería de software. Sangolqui, Rumiñahui, Ecuador.

Piattini, M. (2016). *Evolución de la ingeniería del software y la formación de profesionales. Revista Institucional de la Facultad de Informática UNLP.* . Obtenido de Evolución de la ingeniería del software y la formación de profesionales. Revista Institucional de la Facultad de Informática UNLP. : http://sedici.unlp.edu.ar/bitstream/handle/10915/57358/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Pohl, K. &. (2015). *Santa Barbara CA: Rocky Nook Inc. Requeriments Engineering Fundamentals: a study guide for the certified professional for re quirements engineering exam, foundation level--IREB compliant (second edi tion):*. Obtenido de Santa Barbara CA: Rocky Nook Inc. Requeriments Engineering Fundamentals: a study guide for the certified professional for re quirements engineering exam, foundation level--IREB compliant (second edi tion)::

https://rockynook.com/shop/computing/requirements-engineering-fundamentals-2nd-edition/

Pressman, R. (2005). *Un enfoque práctico (Séptima ed.)*. Obtenido de Un enfoque práctico (Séptima ed.): https://docs.google.com/document/d/1x1uFkX13aWHfVkssPTqSZgRDN-b0GiYVXuSScnQ3YCs/edit?usp=sharing

Solis, C. (2024). *In Learning. Scrum esencial:*. Obtenido de In Learning. Scrum esencial:: https://www.linkedin.com/learning/aprende-scrum/presentacion-del-curso-aprende-scrum

Somerville, I. (2011). *Software Engineering 9.* Obtenido de Software Engineering 9.: https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/index.html