Ingeniería de Software

Actividad de aprendizaje 1.2: Tabla Comparativa "Metodologías de Desarrollo de Software"

Integrantes

Ibarra Perez Oscar Fernando	219549216
Murillo Garcia Mario Abraham	218756307
Viveros Gutierrez Gabriel	223386623
Viña Hernández Antonio	223386356

Profesor
Francisco Javier Quintanilla Moreno

Fecha de entrega
26 de enero del 2025

Introducción a la actividad

Las metodologías de desarrollo de software establecen los lineamientos, fases, características, ventajas y desventajas para el diseño de sistemas de información. Por lo que es de suma importancia que se conozcan para establecer cuál es la idónea para un contexto específico, dadas las características de las necesidades de información, de la organización y del equipo de desarrollo.

Objetivo de la actividad

El estudiante distinguirá la metodología de desarrollo de software idónea para el diseño de un sistema de información.

Instrucciones

1. El profesor integrará equipos de trabajo de 5 estudiantes.
2. Leer detenidamente las siguientes fuentes bibliográficas publicadas en la plataforma moodle (puedes hacer uso de la Biblioteca Digital de la Universidad de Guadalajara):
    a. Capítulo 2 “Procesos de software”. Sommerville, Ian(2011). Ingeniería de Software.
    b. Páginas 27 a la 29: “Tabla Comparativa”. Pimienta, Julio (2012). Estrategias de enseñanza-aprendizaje. México: Pearson education.
    c. Páginas 94 y 95: “Tabla Comparativa”. Pimienta, Julio (2007). Metodología constructivista: guía para la planeación docente. México: Pearson education
3. Cada equipo de trabajo discutirá las características, ventajas y desventajas de cada metodología de desarrollo de software. Una vez finalizada la discusión crearán una tabla comparativa para expresar lo siguiente: Metodología, Características, Ventajas, Desventajas.
4. En un documento de word insertará la tabla comparativa realizada.
5. Una vez terminado el tiempo para la entrega de la actividad expondrán sus productos

Cuerpo del documento

Tabla con ayuda de Chat GPT

Metodología	Características	Ventajas	Desventajas
Modelo en cascada	Es un proceso lineal y secuencial, que está dividido en las fases: análisis, diseño, implementación, pruebas, despliegue, y mantenimiento.	- Simple y fácil de entender. - Adecuado para proyectos pequeños o bien definidos. - Documentación clara y exhaustiva.	- Poco flexible ante cambios en los requisitos. - Difícil de aplicar en proyectos complejos o con incertidumbre. - Altos costos si hay errores en etapas tempranas.
Desarrollo incremental	Se desarrollan versiones del sistema de forma iterativa, con cada incremento añade funcionalidad adicional. De esta manera logra combinar elementos de desarrollo y mantenimiento.	- Más flexible y adaptable a cambios. - El cliente puede ver prototipos iniciales. - Reducción de riesgos al detectar problemas tempranamente.	- Puede ser difícil planificar y estimar costos. - Requiere una buena gestión para evitar desvíos. - Complejidad en la integración de los incrementos.
Orientada a la reutilización	Se basa en el uso de componentes ya existentes. Además se enfoca en hacer módulos que sean reutilizables, de esta forma enfatiza la personalización e integración en lugar de desarrollo desde cero.	- Reducción de tiempo y costos de desarrollo. - Aprovechamiento de componentes probados y confiables. - Incrementa la productividad.	- Dependencia de componentes existentes, lo que puede limitar la personalización. - Dificultad para integrar componentes heterogéneos. - Riesgos de problemas de compatibilidad.

Tabla con ayuda de Microsoft Copilot

Metodología	Características	Ventajas	Desventajas
Modelo en cascada	Sigue un enfoque lineal y secuencial. Esto significa que el desarrollo avanza en fases específicas y ordenadas. Entre sus características principales están la secuencialidad y	- Claridad y control - Fácil de entender - Estructura clara	- Rigidez menos flexible para adaptarse a cambios - Retrasos: los errores retrasan significativamente

	la necesidad de una documentación detallada en cada fase antes de avanzar a la siguiente.		- Adaptabilidad limitada: no es ideal para proyectos de software
Desarrollo incremental	- Desarrollo en incrementos: el proyecto se divide en partes pequeñas para entregar de manera incremental. - Iterativo: Cada ciclo incluye todas las fases del desarrollo - Flexibilidad: Permite realizar cambios y mejoras en cada iteración	- Entrega rápida y constante - Reducción de riesgos: se detectan los problemas y se corrigen en etapas tempranas - Mejor adaptabilidad: Facilita la adaptación	- Complejidad en la gestión - Mayor costo para realizar revisiones frecuentes - Dependencia de la retroalimentación
Orientada a la reutilización	- Componentes reutilizables: se enfoca en crear componentes que puedan ser reutilizados - Modularidad: Se construye a partir de módulos independientes que pueden integrarse fácilmente . Bibliotecas y frameworks: Utiliza bibliotecas y frameworks que faciliten dicha integración de componentes.	- Ahorro de tiempo y costos: ya que se reutilizan los componentes. - Mayor confiabilidad: Suelen estar probados, depurados por lo que aumenta la confiabilidad - Mantenimiento simplificado: Facilita el mantenimiento y las actualizaciones	- Dependencia a componentes externos. - Integración compleja: se debe realizar una integración dinámica lo que provoca que sean más complejas y requieren ajustes adicionales - Falta de personalización: Pueden no ajustarse perfectamente.

Repositorio de Github

https://github.com/AntonioTach/IngenieriaDeSoftware-Equipo5

Conclusiones

Fernando Ibarra: Con esta actividad me pude dar cuenta que dependiendo la metodología que sigas puedes llegar más rápido a un resultado, o perfeccionar un producto, lo cual me lleva a concluir que cada modelo tiene lo suyo, y deben ser usados dependiendo de la situación, el producto, el cliente, la complejidad, entre otras características.

Mario Murillo: Con esta actividad se reafirmó lo ya dicho en clase previamente, de que a pesar de ser modelos diferentes, todos tienen la misma base en que son ia´s generativas, así que el resultado que quieras obtener depende 100% de la forma en la que formules tus preguntas.

Gabriel Viveros: En conclusión, en esta actividad podemos hacer la comparación de los diferentes métodos para conocer sus ventajas y desventajas, para así tener en cuenta a la hora de elegir una para utilizarla, dependiendo del contexto y lo que se quiera lograr.

Antonio Viña: Mi conclusión obtenida es que cada metodología tiene grandes ventajas y desventajas, es importante identificar creo yo en donde utilizarla y porque, cada una es adecuada dependiendo del proyecto. En mi perspectiva todas son buenas. Pero cambia según tu metodología de trabajo y del proyecto que deseas realizar. Si requieres mayor estructura clara y ordenada la de cascada es buena opción, si requieres algo rápido y adaptable la incremental, si requieres algo que pueda reutilizar mucho y que se que realizare el proyecto varias veces para distintos clientes, el orientado a la reutilización.

Bibliografías

- *Capítulo 2 “Procesos de software”. Sommerville, Ian(2011). Ingeniería de Software.*
- *Páginas 27 a la 29: “Tabla Comparativa”. Pimienta, Julio (2012). Estrategias de enseñanza-aprendizaje. México: Pearson education.*
- *Páginas 94 y 95: “Tabla Comparativa”. Pimienta, Julio (2007). Metodología constructivista: guía para la planeación docente. México: Pearson education*

IA’s Utilizadas

- ChatGPT (OpenAI)
- Copilot