

Tarea 2

Ciclo de vida del software (Relación 1)

1.- Define "Ciclo de vida del software".

El ciclo de vida de un software es una secuencia estructurada y bien definida de las etapas en ingeniería de software para desarrollar el software deseado.

Lo describe desde el inicio hasta el final, con el objetivo de definir las distintas fases intermedias para validar el desarrollo de la aplicación y confirmar que cumpla con los requisitos de verificación.

2.- Nombra las fases principales del desarrollo de software y explica brevemente qué se hace en cada una de ellas.

-**Análisis**: establecer los requisitos del programa(que tiene que hacer el programa).Autenticar los requisitos(100%).No tiene que dudar.Utilizar lenguaje formal.Tiene que ser entendible por el cliente.Requisitos no funcionales:son las restricciones o los requisitos impuestos al sistema y requisitos funcionales:es una declaración de cómo debe comportarse un sistema.Dividir y jerarquizar(organizar por niveles) el modelo.

-**Diseño**:Se descompone y organiza el sistema en elementos componentes que pueden ser desarrollados por separado.Se especifica la interrelación y funcionalidad de los elementos componentes.

-**Codificación**:escribir el código fuente.Pueden utilizarse distintos lenguajes informáticos:

o Lenguajes de programación: C, C + +, Java, Javascript, ...

o Lenguajes de otro tipo:HTML,XML,JSON, ...

-**Pruebas**:El principal objetivo de las pruebas debe ser conseguir que el programa funcione incorrectamente y que se descubran defectos. Deberemos someter al programa al máximo número de situaciones diferentes

-**Mantenimiento**:Durante la explotación del sistema so ware es necesario realizar cambios ocasionales. Para ello hay que rehacer parte del trabajo realizado en las fases previas.



3.- Explica brevemente en qué consiste el modelo en cascada cuando hablamos de desarrollo de software.

Es el modelo más antiguo. Identifica las fases del software. Tiene que pasar por todas las fases. Es el modelo más adecuado para proyectos con requisitos bien definidos, donde no se esperan cambios apresurados en la gestión de proyectos.

Si bien el uso del modelo en cascada ha disminuido en los últimos años a favor de las metodologías ágiles, este sigue siendo un proceso de diseño de software común en la industria.

En este artículo expondremos lo que es el desarrollo en cascada y qué proyectos de software son los más adecuados para su implementación.



4.- Ventajas e inconvenientes del modelo en cascada.

Ventajas:

- Usa una estructura clara: En comparación con otras metodologías, la cascada se concentra mayormente en una serie de pasos claros y definidos. El avance en la cascada es intuitivo.
- Determina el objetivo final rápidamente: Uno de los pasos definitorios del método de cascada es comprometerse con un producto final, un objetivo o un entregable desde el principio, y los equipos deberían evitar desviarse de ese compromiso.
- Transmite bien la información: El enfoque de la cascada es sumamente metódico, así que no debería resultar una sorpresa que la metodología enfatice una transferencia clara de información en cada paso.

Desventajas:

- Dificulta los cambios: La metodología de la cascada se basa completamente en seguir una serie de pasos que hacen que los equipos siempre avancen. Esta metodología, en su forma tradicional, no deja prácticamente ningún lugar para cambios o revisiones imprevistos.
- Retrasa las pruebas hasta después de la finalización: Dejar la fase de pruebas para la última mitad de un proyecto es riesgoso, pero la metodología de cascada insiste en que los equipos esperen hasta el paso cuatro de seis para probar sus productos.

5.- ¿Qué se entiende por verificación? ¿Y por validación?

Verificación:

Depende de la validación, es decir, que para verificar si una cosa está bien o mal, primero hay que validar. Suele ser el proceso que se realiza para revisar si una determinada cosa está cumpliendo con los requisitos y normas previstos.

Validación:

Es un requisito que deben completar las empresas del sector para cumplir con la normativa aplicable y conseguir las autorizaciones y certificaciones necesarias. Es un procedimiento que garantiza que una aplicación o sistema informático cumple con las buenas prácticas de fabricación, por lo que se garantiza la calidad de sus procesos informáticos.

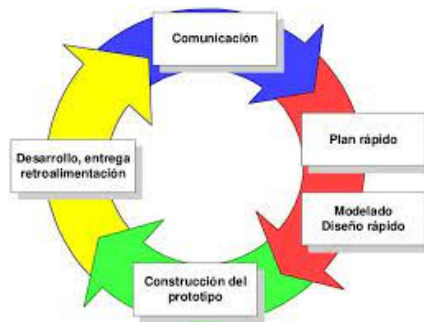
6.- Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.

Se crea un prototipo durante la fase de análisis y es probado por el usuario/cliente para refinar los requisitos del software a desarrollar. Se repite el paso anterior las veces necesarias.

Tipos de prototipos:

- rápido: se describe como un método basado en ordenador que pretende reducir el ciclo iterativo de desarrollo. En este modelo se da prioridad al desarrollo rápido pues el objetivo es conseguir una rápida evaluación del modelo

- evolutivo: La especificación se puede mostrar de forma creciente. Sin embargo es muy posible que el cliente no quiera que empecemos a construir el sistema desde 0. El lo que querrá será que construyamos el sistema partiendo del prototipo, esto lo convertiría en un prototipo evolutivo



7.- Explica cómo funciona el modelo espiral cuando se aplica al desarrollo orientado a objetos.

Es una combinación entre el modelo lineal o de cascada y el modelo iterativo. Se utiliza con éxito en proyectos donde el coste de un fallo es un gran riesgo, de ahí que su principal aportación sea considerar la gestión de esos riesgos, algo que en los modelos anteriores ni siquiera se menciona. Los factores de riesgo son reducidos. La duración de la ejecución no es concreta. Es una forma innovadora y efectiva de desarrollar y diseñar software. Cada ciclo se compone de cuatro etapas clave: planificación, análisis, diseño y evaluación.

En la etapa de planificación, se define el alcance del proyecto y se establecen los objetivos y las metas.

En la etapa de análisis, se recopilan y analizan los requisitos del usuario.

En la etapa de diseño, se crea una arquitectura de software en base a los requisitos del usuario, y se definen los objetos y las clases necesarias.

En la etapa de evaluación, se evalúa el software en función de los requisitos establecidos y se identifican los problemas que deben ser corregidos.

