

Metodologías de desarrollo de Software – Practico N° 1

1- Describa brevemente el significado de los siguientes términos:

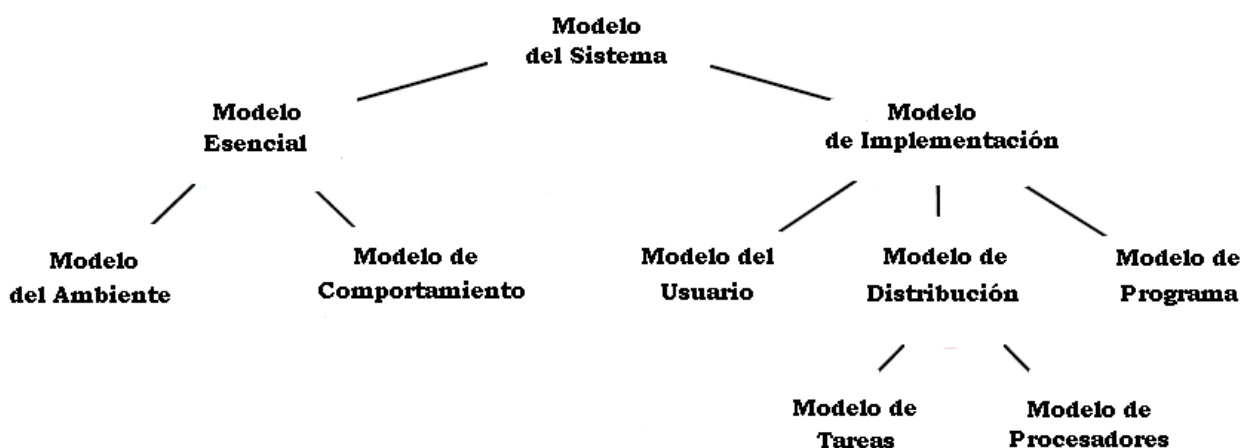
a- Metodología de desarrollo de software: es un proceso organizado para la producción de software. Especifica el ciclo de vida a utilizar, indicando además qué personas deben desempeñar cada rol en el desarrollo de las actividades.

b- Modelo de ciclo de vida para el desarrollo del software: Indica el orden en que voy a llevar a cabo las etapas y lo que hay que obtener en cada una de ellas a lo largo del desarrollo del proyecto.

c- Etapa dentro de un ciclo de vida: Son los diferentes pasos a seguir durante el desarrollo del software. Determinado orden de estos pasos se los conoce como ciclo de vida. Ejemplo captura de requerimientos, análisis del problema, diseño, implementación, testing, mantenimiento.

d- Rol que debe cumplir una persona en el desarrollo de software: Es el papel que juega una persona en el desarrollo del software, como por ejemplo analista, diseñador, programador, tester.

e- Modelo/Diagrama de las características de un sistema de software y sus partes componentes:



Modelo Esencial: Responde a la pregunta ¿qué tiene que satisfacer el sistema?

Modelo del Ambiente: Declaración de los Objetivos, captura de requerimientos.

Modelo de Comportamiento: Análisis.

Modelo de Implementación: Responde a la pregunta ¿Cómo realizar el sistema? Se deben considerar, las imperfecciones de la tecnología y determinar: la cantidad de procesadores necesarios, las cualidades de estos, el tamaño de disco necesario de acuerdo al volumen de la información a ser almacenada, etc. Luego se diseña la solución sobre la base de esas restricciones tecnológicas.

Modelo del Usuario: Interfaz.

Modelo de Distribución: Decide como distribuyo mi sistema.

Modelo de Tareas: Describe todas las decisiones relativas a la arquitectura de software.

Modelo de Procesadores: Describe todas las decisiones relativas a la arquitectura de hardware.

Modelo de Programa: Se realiza un diagrama estructurado del programa a

partir de las técnicas y estrategias elegidas.

2- Describa brevemente en que situación es imprescindible seguir una metodología para el desarrollo de software, y en cuales situaciones no lo sería tanto.

Cuando el sistema es grande y participan varias personas en su desarrollo cumpliendo un rol específico, es necesaria una metodología, mientras que en un sistema pequeño donde trabajan una o dos personas las modificaciones que pueden surgir a lo largo del desarrollo no son tan significativas como para seguir una metodología.

3- Enumere las categorías mas comunes de metodologías existentes para el desarrollo de software junto con sus características principales.

- Orientada a dato o función: en este método las funciones y/o los datos son tratados como entidades independientes. Estos sistemas resultan difíciles de mantener. El mayor problema es que las funciones generalmente dependen de la estructura de los datos. A menudo diferentes tipos de datos tienen distintos formatos y se necesita verificar el tipo del dato (con sentencias If-Then o CASE), produciendo programas difíciles de leer y modificar. Si se desea hacer alguna modificación en la estructura de los datos se debe modificar en todos los lugares donde es utilizado.
- Orientada a Objetos: Son aquellos métodos en los cuales datos y funciones están altamente relacionados. El énfasis está centrado en la abstracción de datos. Se piensa en forma natural, los objetos representan entidades del mundo real. Los programas son fácilmente mantenibles y extensibles por medio de la construcción de subclases.
- Formales: Surgieron como enfoques analíticos en donde el desarrollo de software puede ser verificado mediante demostraciones matemáticas, garantizando el correcto funcionamiento bajo situaciones que involucran ambientes sensibles e información crítica en donde es necesario que cada uno de los componentes involucrados (hardware, software y componentes humanos) actúe de manera correcta ante situaciones específicas. Por ejemplo: planeación de tráfico, aplicaciones militares y sistemas médicos, entre otras.
- Métodos Ágiles: dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Surge con la necesidad de reducir los tiempos de desarrollo pero manteniendo una alta calidad en sistemas muy cambiantes.

4- Enumere los modelos de ciclos de vida mas comunes para el desarrollo de software, junto con sus características principales.

- El ciclo de vida en Cascada, su principal característica es que para pasar a la siguiente etapa debe estar terminada la primera. Esto traía varios problemas a la hora de hacer modificaciones si eran requeridas.
- El ciclo de vida en Espiral, soluciona el problema del ciclo de vida en cascada ya que su desarrollo es incremental y permite realizar tareas de diferentes etapas.

5- Enumere las etapas mas comunes que comprenden los diferentes modelos de ciclo de vida existentes para el desarrollo del software. Describa brevemente el propósito de cada una.

- Captura de requerimientos: es donde el cliente expone sus necesidades y requerimientos.
- Análisis: se modelan los requerimientos del usuario.
- Diseño: se planea una solución al problema, tiene en cuenta como va a ser implementado.
- Implementación: se implementa el sistema.
- Testeo: se prueba que el sistema funcione correctamente.
- Mantenimiento: es donde se actualiza o modifica el sistema si surgen nuevos requerimientos. Es la etapa más difícil del desarrollo.

6- Enumere los roles mas comunes que puede cumplir una persona en el desarrollo del software. Describa brevemente las responsabilidades principales de cada uno.

- Ingeniero de requerimientos (Documentador): trabaja con el cliente para realizar el análisis y la especificación del sistema a construir. Esta capacitado para obtener claramente todos los requisitos necesarios para el desarrollo del software.
- Analista: Estudia el problema (de una complejidad determinada) y lo descompone en subproblemas de menor complejidad. Transforma los requisitos de usuario en requisitos de software
- Diseñador: Es el encargado de generar el diseño arquitectónico y diseño detallado del sistema, basándose en los requisitos.
- Programador: Los programadores deben convertir la especificación del sistema en código fuente ejecutable utilizando uno o más lenguajes de programación, así como herramientas de software de apoyo a la programación. No necesita conocer el funcionamiento del sistema, solo se encarga de codificar los módulos a partir de los distintos datos de entrada y salida que se le especifican.
- Tester: El tester realiza las tareas de detección y eliminación de los errores y defectos del sistema en construcción.
- Ingeniero de Manutención: adapta los sistemas existentes de acuerdo a cambios en su ambiente externo, realiza las mejoras pedidas por los usuarios, y la adaptación del sistema para usos futuros.

7- Enumere los factores que influyen a la hora de elegir el modelo de ciclo de vida para el desarrollo de un sistema.

- Características del Sistema: Complejidad de los requerimientos, dominio de aplicación (estático o cambiante durante el desarrollo del sistema)
- El cliente: muchas veces el usuario decide realizar cambios en los requerimientos del sistema cuando el proyecto se encuentra muy avanzado o terminado lo que ocasiona un atraso en el proyecto, por eso es preferible no seguir un ciclo de vida estático sino adaptarse a las características del proyecto.

8- Considere el desarrollo de un sistema cuyo dominio de aplicación es conocido, sus objetivos y requerimientos funcionales son estables y simples de comprender desde un principio, la tecnología a utilizar ya esta predeterminada y es bien conocida por el equipo de desarrollo. ¿Qué tipo de modelo de ciclo de vida elegirías para el desarrollo de dicho sistema?

Si el sistema es pequeño, pero tengo que seguir un modelo con esas características elegiría un ciclo de vida en cascada.

9- Una vez elegido el modelo de ciclo de vida para el desarrollo del sistema planteado en el ejercicio anterior ¿Qué etapas escogerías para dicho modelo de ciclo de vida, teniendo en cuenta que el desarrollo lo realizan una o pocas personas?

No sería necesaria la captura de requerimientos pero las demás etapas serían necesarias. Para evitar problemas a la hora de la utilización del sistema.

10- Considere ahora el desarrollo de un sistema cuyo dominio de aplicación no es muy conocido por el equipo de desarrollo. En este caso, el cliente tampoco tiene muy claro que es lo que quiere, de manera que los objetivos y requerimientos funcionales del sistema son inestables y difíciles de comprender. Además, el equipo de desarrollo va a utilizar una tecnología que le resulta completamente nueva. Discuta que modelo de ciclo de vida es más apropiado y qué etapas se deberían utilizar para desarrollar este sistema.

En este caso elegiría un ciclo de vida en espiral, y serían necesarias todas las etapas del ciclo de vida, y con mucha mas prioridad la captura de requerimientos, ya que no se quiere que al finalizar el sistema, se tenga una solución que difiere de lo que buscaba el cliente.

11- Considere ahora que el dominio del sistema a desarrollar es el de Control de Tráfico Ferroviario de una gran ciudad. ¿En cual de los tipos de sistemas que conoce ubicaría a este sistema? ¿Que tipo de metodología de desarrollo de software utilizaría en este caso? ¿Por qué? ¿Cuales son los factores a tener en cuenta para elegir este tipo de metodología?

Para el desarrollo de este sistema lo más adecuado sería elegir una Metodología Formal. Se debe estar completamente seguro del correcto funcionamiento de este ya que corre riesgo la vida de las personas.