

MODELOS DE DESARROLLO DE SOFTWARE

Un modelo para el desarrollo de software es una representación abstracta de un proceso. Cada modelo representa un proceso desde una perspectiva particular y así proporcione información parcial sobre el proceso. Éstos modelos generales no son descripciones definitivas de los procesos del software más bien son abstracciones de los procesos que se pueden utilizar para el desarrollo del software.

MODELO DE PROTOTIPO

Este diseño se centra en una representación de la parte visual del software que serán vistos por el usuario final. Este será evaluado por el cliente para retroalimentación por este medio se refinan los requisitos del sistema que se desarrollará. La interacción ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo. Pertenece a los modelos de desarrollo evolutivo. Se debe de construir en poco tiempo usando programas adecuados y optimizando recursos.

CARACTERÍSTICAS

- El prototipo es una aplicación que funciona
- Los prototipos se crean con rapidez
- Los prototipos evolucionan a través de un proceso iterativo.
- Los prototipos tienen un costo bajo de desarrollo.

CARACTERÍSTICAS

- Reduce considerablemente que fallen los proyectos grandes de software, ya que evalúa repetidamente los riesgos y verifica cada vez el producto en desarrollo
- Contiene componentes de casi cualquier otro modelo del ciclo de vida del software, como el modelo de cascada, el modelo de creación de prototipos, etc.
- Es capaz de manejar casi cualquier tipo de riesgo que por lo general no manejan los otros modelos. Sin embargo, debido a tener tantos componentes, este modelo es mucho más complejo que los otros modelos de desarrollo del software.

MODELO ESPIRAL

Consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo Cascada, pero no necesariamente debe ser así. El Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del modelo MCP con los aspectos controlados y sistemáticos del Modelo Cascada, con el agregado de gestión de riesgo.

Al ser un modelo de Ciclo de Vida orientado a la gestión de riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique tenga la necesaria experiencia y habilidad para detectar y catalogar correctamente los riesgos.

MODELO DE CASCADA

Es el enfoque metodológico que ordena rigurosamente las etapas del proceso para el desarrollo de software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la etapa anterior.¹ Al final de cada etapa, el modelo está diseñado para llevar a cabo una revisión final, que se encarga de determinar si el proyecto está listo para avanzar a la siguiente fase. Este modelo fue el primero en originarse y es la base de todos los demás modelos de ciclo de vida.

Es un modelo conocido y utilizado con frecuencia. Promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar.

CARACTERISTICAS

- Cada fase genera documentación para la siguiente. Esta documentación debe ser aprobada.
- Una fase no comienza hasta que la anterior ha terminado. • Requiere disponer de unos requisitos completos y precisos al principio del desarrollo.
- Se disponga de unos requisitos completos y consistentes al principio del desarrollo.
- Sea un proyecto pequeño, en el que el período de congelación de los requisitos es corto, o un proyecto con unos requisitos bastante estables.

CARACTERISTICAS

- Los casos de uso son los artefactos primarios para establecer el comportamiento deseado del sistema.
- La arquitectura es utilizada para conceptualizar, construir, administrar y evolucionar el sistema en desarrollo.
- Maneja una serie de entregas ejecutables.
- Integra continuamente la arquitectura para producir nuevas versiones mejoradas.

MODELO RUP

Es un proceso de ingeniería de software, que hace una propuesta orientada por disciplinas para lograr las tareas y responsabilidades de una organización que desarrolla software.

Su meta principal es asegurar la producción de software de alta calidad que cumpla con las necesidades de los usuarios, con una planeación y presupuesto predecible.