



EL PROCESO DEL SOFTWARE.

Es un mapa de carreteras que le ayude a obtener el resultado oportuno de calidad.

INGENIERIA DEL SOFTWARE: UNA ESTRATEGÍA ESTRATIFICADA.

La ingeniería del software es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.

La aplicación de, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software.

Proceso, métodos y herramientas.

El proceso de la ingeniería del software es la unión que mantiene las capas de tecnologías y que permite un desarrollo racional y oportuno de la ingeniería del software.

Los métodos indican como construir técnicas del software. Abarcan una gama de tareas que incluyen análisis de requisitos, diseño, construcción de programas, pruebas y mantenimiento.

Las herramientas establecen un sistema de soporte para el desarrollo de software llamado *ingeniería del software asistido por computadora (CASE)*.

Una visión general de la ingeniería del software.

La ingeniería es el análisis, diseño, construcción, verificación y gestión de entidades técnicas o sociales.

El trabajo que se asocia a la ingeniería del software se puede dividir en tres fases genéricas:

La *fase de definición*, identifica que información ha de ser procesada, que función y rendimiento se desea, que compartimiento del sistema, que interfases van a ser establecidas, que interfases de diseño existen, y que criterios de validación se necesitan para definir un sistema correcto.

La *fase de desarrollo*, como han de diseñarse las estructuras de datos, como ha de implementarse la función dentro de una arquitectura del software, como ha de implementarse los detalles procedimentales, como han de caracterizarse interfaces, como ha de traducirse el diseño en un lenguaje de programación y como ha de realizarse la prueba.

La *fase de mantenimiento*, cambios asociados a la corrección de errores a las adaptaciones requeridas a medida que evoluciona el entorno del software y a



cambios debidos a las mejoras producidas por los requisitos cambiantes del cliente. En esta fase existen cuatro tipos de cambios:

- **Corrección.** Cambia el software para corregir los defectos.
- **Adaptación.** Produce modificación en el software para acomodarlo a los cambios de su entorno externo.
- **Mejora.** Lleva al software más allá de sus requisitos funcionales originales.
- **Prevención.** Llamado reingeniería del software, se debe conducir a permitir que el software sirva para las necesidades de los usuarios finales. Hace cambios en programas de computadora fin de que se pueda corregir, adaptar y mejorar más fácilmente.

El enfoque del Software Engineering Institute (SEI) proporciona una medida de la efectividad global de las prácticas de ingeniería del software de una compañía y establece cinco niveles de madurez del proceso:

- **Nivel 1: Inicial.** Se definen pocos procesos.
- **Nivel 2: repetible.** Se establecen los procesos de gestión del proyecto para hacer seguimiento del coste, planificación y funcionalidad.
- **Nivel 3: Definido.** Las actividades de gestión y de ingeniería se documenta, se estandariza y se integra dentro de un proceso de software de toda una organización.
- **Nivel 4: Gestionado.** Se recopilan medidas detalladas para comprender y controlar cuantitativamente tanto productos como el del proceso del software.
- **Nivel 5: Optimización.** Mediante una retroalimentación cuantitativa del proceso, ideas y tecnologías innovadoras se posibilita una mejora del proceso.

La SEI ha asociado áreas claves del proceso (ACPs) a cada uno de los niveles de madurez.

Cada ACP se describe identificando las características siguientes:

- *Objetivos* que debe alcanzar la ACP.
- *Compromisos* que debe cumplir para lograr objetivos.
- *Capacidades*, elementos que deben encontrarse para permitir a la organización cumplir los objetivos.
- *Actividades*, tareas específicas que se requieren PATRA lograr la función ACP.
- *métodos para supervisar la implementación*, la manera en que las actividades son supervisadas conforme se aplican.
- *métodos para verificar la implementación*, la forma en que se puede verificar la practica adecuada para la ACP.



Las ACP, se definen como practicas claves que son normas, procedimientos y actividades que deben ocurrir antes de que haya instituido completamente un área clave de proceso.

MODELOS DE PROCESO DEL SOFTWARE.

Se selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren.

En el desarrollo del software se puede caracterizar como bucle de resolución de problemas en el que se encuentran cuatro etapas distintas: status quo definición de problemas, desarrollo técnico e integración de soluciones. Status quo representa el estado actual de sucesos; la definición de problemas identifica el problema específico a resolverse; el desarrollo técnico resuelve el problema a través de la aplicación de alguna tecnología y la integración de soluciones ofrece los resultados a los que solicitan la solución.

MODELO LINEAL SECUENCIAL.

También llamado “ciclo de vida clásico” o “modelo en cascada”, sugiere un enfoque sistemático, secuencial, para el desarrollo del software que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

Este modelo comprende las siguientes actividades:

Ingeniería y modelado de sistemas/Información. Estableciendo requisitos de todos los elementos del sistema y asignado al software algún subgrupo de estos requisitos.

Diseño. Se centra en cuatro atributos distintos de programa: estructura de datos, arquitectura de software, comportamiento, rendimiento e interconexión.

Generación de código. El diseño se debe traducir en una forma legible por la máquina.

Pruebas. Se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales; realización de pruebas para detección de errores y asegurar que la entrada definida produce resultados reales de acuerdo con los requisitos requeridos.

Mantenimiento. El software tendrá cambios después de ser entregado al cliente. Se producirán cambios porque se han detectado errores, porque el software debe adaptarse a para acoplarse a los cambios de su entorno externo.

Problemas del modelo lineal secuencial:



1. Los proyectos reales rara vez siguen el modelo secuencial que propone el modelo.
2. Es difícil que el cliente exponga explícitamente todos los requisitos
3. El cliente debe tener paciencia.

MODELO DE CONSTRUCCION DE PROTOTIPOS

El paradigma de construcción de prototipos comienza con la recolección de requisitos. El desarrollador y el cliente se encuentran y definen los objetivos globales para el software, identifican los requisitos conocidos y las áreas del esquema en donde es obligatoria más definición. Aparece un diseño rápido, que se centra en una representación de esos aspectos del software que serán visibles para el usuario/cliente. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el cliente/usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

La construcción de prototipos puede ser problemática por las siguientes razones:

1. El cliente ve lo que parece ser una versión de trabajo del software.
2. El desarrollador, hace compromisos de implementación para hacer que el prototipo funcione rápidamente.

MODELO DRA.

Desarrollo rápido de Aplicaciones (DRA) es un modelo del proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una aplicación a alta velocidad del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Cuando utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

Modelo de Gestión. Se modela el flujo de información entre las funciones de gestión.

Modelado de datos. Conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características de cada uno de los objetos y las relaciones entre estos objetos.

Modelado del proceso. Los objetos de datos quedan transformados para lograr el flujo de información. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos.

Gestión de aplicaciones. Utilizar componentes de programas ya existentes o a crear componentes reutilizables. En todos los casos se utilizan herramientas para facilitar la construcción del software.



Pruebas y entrega. Se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

MODELOS EVOLUTIVOS DE PROCESO DEL SOFTWARE.

Se caracterizan por la forma en que permiten a los ingenieros del software desarrollar versiones cada vez más complejas del software.

Modelo incremental.

Combina elementos del modelo secuencial con la filosofía interactiva de construcción de prototipos, aplica secuencias de forma escalonada, cada secuencia lineal produce un incremento del software.

Se centra en la entrega de un producto operacional con cada incremento.

Modelo espiral.

Es un modelo de proceso de software evolutivo que conjuga la naturaleza interactiva de construcción de prototipos con los aspectos controlados y sistemáticos del modelo línea secuencial.

En este modelo, el software se desarrolla en una serie de versiones incrementales.

El modelo espiral se divide en una serie de actividades de marco de trabajo, llamadas regiones de trabajo:

- Planificación.
- análisis de riesgos.
- Ingeniería.
- construcción y acción.
- Evaluación del cliente.

El modelo espiral es un enfoque realista del desarrollo de sistemas y de software a gran escala. Utiliza la creación de prototipos como mecanismo de reducción de riesgos, permite a quien lo desarrolla aplicar el enfoque de construcción de prototipos en cualquier etapa de evolución del producto.

Modelo espiral WINWIN.

Define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral.

Modelo de desarrollo concurrente.



Se puede representar en forma de esquema como una serie de actividades técnicas importantes, tareas y estados asociados a ellas. Define una serie de acontecimientos que dispararán transacciones de estado para cada una de las actividades de la ingeniería del software. Por ejemplo, durante las primeras etapas del diseño, no se contempla una inconsistencia del modelo de análisis. Esto genera la corrección del modelo de anales de sucesos que disparará la actividad de análisis del estado hecho al estado cambios en espera.

Este modelo se utiliza a menudo como paradigma de desarrollo de aplicaciones cliente/servidor.

DESARROLLO BASADO EN COMPONENTES.

Conduce a la reutilización del software, y la reutilización proporciona beneficios a los ingenieros del software.

El proceso unificado de desarrollo de software representa un numero de modelos de desarrollo basados en componentes que has sido propuestos en al industria.

MODELO DE METODOS FORMALES.

Comprende un conjunto de actividades que conduce a las especificaciones matemáticas del software de computadora.

Proporciona un mecanismo para eliminar muchos de los problemas que son difíciles de superar con paradigmas de la ingeniería del software.

Cuando se utilizan métodos formales durante el diseño, sirven como base para la verificación de programas y permiten que el ingeniero del software descubra y corrija errores.

TECNICAS DE CUARTA GENERACIÓN.

(T4G) facilita al ingeniero de software la especificación de algunas características del software a alto nivel. Cuanto mayor sea el nivel en el que se especifique el software, mas rápido se podrá construir el programa. El paradigma T4G para la ingeniería del software se orienta hacia la posibilidad de especificar el software usando formas de lenguaje especializado o notaciones graficas que describa el problema que hay que resolver en términos que lo entienda el cliente.

Un entorno para el desarrollo de software que soporte el paradigma T4G puede incluir todas o algunas de las siguientes herramientas: lenguaje no procedimentales de consulta a base de datos, interacción y definición de pantallas, generación de códigos, capacidades graficas de alto nivel y capacidades de hoja de calculo, y generación automatizada de HTML y lenguajes similares utilizados para la creación de sitios web.

TECNOLOGIAS DE PROCESO.



Permiten que una organización de software construya un modelo automatizado del marco de trabajo común de proceso, conjuntos de tareas y actividades de protección.

PRODUCTO Y PROCESO.

La ingeniería del software es una aplicación que integra procesos, métodos y herramientas para el desarrollo del software de computadora. Se han propuesto varios modelos de procesos para la ingeniería del software diferentes, cada uno exhibiendo ventajas e inconvenientes, pero todos tienen una serie de fases genéricas en común.