



Nombre de alumno:

Teresa Méndez Pérez

Nombre del profesor:

Icel Bernabé Lepe Arriaga

Nombre del trabajo:

Ensayo

Materia:

Ingeniería en software

Grado: 8 cuatrimestre

Comitán de Domínguez Chiapas a 03 de marzo de 2022.

Introducción

Como tema principal “fundamentos de la ingeniería del software” del cual nos surgen otros subtemas los cuales son los siguientes: 1.1.- Definición y objetivos de la Ingeniería del Software. 1.2.- Características y aplicaciones del software. 1.3.- Evolución histórica del software. 1.4.- Perspectiva general de la Ingeniería del Software 1.5.- Procesos, métodos y herramientas. 1.6.- Modelo clásico o lineal, modelo en cascada. 1.7.- Construcción de prototipos. 1.8.- Modelos evolutivos.

Los siguientes temas antes mencionados se trataran de explicar lo más claro y concretamente posible, por su contenido que es muy extenso se tratara la manera de ser preciso, quiero mencionar principalmente que en esta ocasión hay algunos subtemas que conozco y otros no, estoy muy vaga en ese sentido porque si necesito investigar un poco más, me apoyare en la antología y en otras fuentes las cuales anexare al final de la investigación para que se pueda corroborar la información plasmada.

Sin más por agregar por el momento pasare a la investigación de los puntos antes mencionados, esperando tener una buena información, continuaremos con el ensayo.

1.1. DEFINICIÓN Y OBJETIVOS DE LA INGENIERÍA DEL SOFTWARE

Surge en 1968 en Alemania tras la crisis del software

Se puede analizar, diseñar, programar y aplicar un software de manera correcta y organizada, de esta manera se cumplen todas las especificaciones del cliente y el usuario final.

- Mejorar la calidad de los productos de software
- Aumentar la productividad y el trabajo de los ingenieros de software.
- Utilidad
- Facilitar el control en el proceso de desarrollo de software
- Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.

1.2. CARACTERISITICAS Y APLICACIONES DEL SOFTWARE

Las características deseables en un producto de software son:

- Corrección. Que cumpla con su objetivo.
- Usabilidad. Que sea fácil de aprender.
- Seguridad. Que sea resistente a ataques externo.
- Flexibilidad. Que pueda ser modificado por los desarrolladores.
- Portabilidad. Que pueda ser utilizado en diversos equipos.

Características del software:

1. Software de sistema
2. Software de programación
3. Software de aplicación
4. Desarrollo
5. Lenguajes de programación

Aplicaciones de software

> Software de sistema

> Software de tiempo real

>software de negocios

>software de ingeniería/científico

>software incrustado

>software de PC

>software de IA

>aplicaciones web

1.3. EVOLUCIÓN HISTÓRICA DEL SOFTWARE

La primera teoría sobre el software fue propuesta por Alan Turing en su ensayo de 1935 sobre números computables, con una aplicación destinada a la toma de decisiones. El término "software" fue utilizado por primera vez de forma escrita por John W. Tukey en 1958.

1era era (1950-1965) se trabajaba con la idea de codificar y corregir, no existía un planteamiento previo, no existía documentación de ningún tipo, existencia de pocos métodos formales y pocos creyentes de ellos, desarrollo a base de ensayos y errores.

2da era (1965-1972) se busca simplificar los códigos, aplicación de multiprogramación y sistemas multiusuario, sistema de tiempo real ayuda a la toma de decisiones, aparición del software como producto, inicio de la crisis del software, se busca procedimientos del software.

3era era(1972-1989) nuevo concepto: sistemas distribuidos, complejidad en los sistemas de información, aparición de las redes de área local y global, además de los comunicadores digitales, amplio uso de microprocesadores.

4ta era(1989-hoy) impacto colectivo del software, aparición de las redes de información y de las tecnologías orientadas a objetivos, aparecen las redes neuronales, sistemas expertos y SW de inteligencia artificial, la información como valor preponderante dentro de las organizaciones.

1.4 PERSPECTIVA GENERAL DE LA INGENIERIA DEL SOFTWARE

1. Especificación del software: Tienen que definirse tanto la funcionalidad del software como las restricciones de su operación.

2. Diseño e implementación del software: Debe desarrollarse el software para cumplir con las especificaciones.

3. Validación del software: Hay que validar el software para asegurarse de que cumple lo que el cliente quiere.

4. Evolución del software: El software tiene que evolucionar para satisfacer las necesidades cambiantes del cliente.

Las principales metodologías tradicionales o clásicas son:

- Waterfall (cascada)
- Prototipado
- Espiral
- Incremental
- Diseño rápido de aplicaciones (RAD)

Las principales metodologías ágiles son:

- Kanban
- Scrum
- Lean
- Programación extrema (XP)

1.5 PROCESOS, MÉTODOS Y HERRAMIENTAS

Un método de ingeniería del software es un enfoque estructurado para el desarrollo de software cuyo propósito es facilitar la producción de software de alta calidad de una forma costeable. Métodos como Análisis Estructurado (De Marco, 1978) y JSD (Jackson, 1983) fueron los primeros desarrolladores en los años 70.

Descripción de los modelos del Sistema que se desarrollará y la notación utilizada para definir los modelos.

- Reglas
- Restricciones que siempre aplican a los modelos de sistemas.
- Recomendaciones

Técnicas de caja Negra de AQ maneja aplicaciones, o partes de ellas, que ya están construidas. Estas técnicas verifican si el software cumple o no con sus requerimientos.

Técnicas de caja blanca (o caja de vidrio) de QA se aplica a los componentes que forman la unidad que se está probando. Encender el televisor, cambiar los canales y luego observar su calidad equivaldría a la técnica de la caja negra.

1.6 MODELO CLÁSICO O LINEAL, MODELO EN CASCADA

Representación del modelo en cascada

Se desarrolla el software en etapas y que después del término de una etapa no es posible regresar a ella, este modelo tiene cuatro etapas que son:

- Planificación: se determinan los objetivos, metas, requerimientos y restricciones en el proyecto

- .– Análisis de riesgos: identificación de situaciones inconvenientes para evitarlas y solucionarlas.

- Ingeniería: desarrollo del producto con respecto al diseño y otras consideraciones planteadas.

- Evaluación del cliente: valorización de los resultados del proyecto (producto obtenido).

El modelo en lineal secuencial es también conocido como el ciclo de vida del software, está conformado por las etapas de Análisis de requerimientos, Diseño, Codificación y pruebas. Cada etapa tiene dependencias de finalización y orden la una de la otra. En ese orden de ideas no se podrá iniciar la etapa de codificación sino no se ha pasado ya por la etapa de análisis y diseño previamente y respectivamente.

Debido a la forma en la que este proceso aborda la solución de proyectos software tiene algunas ventajas y falencias.

- Análisis de requerimientos: en esta etapa se procede a recopilar todos los requisitos que debe cumplir el software a desarrollar, el cliente aquí tiene un papel fundamental, ya que analiza junto con los desarrolladores del software los requisitos que debe cumplir el software a desarrollar.

– Diseño: esta etapa está enfocada hacia el desarrollo de un esbozo de la arquitectura, la interfaz y el manejo de datos del software.

– Generación de código: es cuando se implementa el diseño del software en algún lenguaje de programación definido en el mismo diseño.

- Pruebas: se hace una revisión de los procesos que realiza el software, para determinar que esté haciendo lo planteado para cumplir con los requerimientos.

– Mantenimiento: se verifica el correcto funcionamiento del software en su entorno de uso, y si existen errores o defectos, proceden a corregirlos

- DESVENTAJAS:
- Los proyectos reales raramente siguen el flujo secuencial que propone el modelo, siempre hay iteraciones y se crean problemas en la aplicación del paradigma.
- Normalmente, es difícil para el cliente establecer explícitamente al principio todos los requisitos. El ciclo de vida clásico lo requiere y tiene dificultades en acomodar posibles incertidumbres que pueden existir al comienzo de muchos productos.
- VENTAJA:
- La ventaja de este método radica en su sencillez ya que sigue los pasos intuitivos necesarios a la hora de desarrollar el software.

1.7 CONSTRUCCIÓN DE PROTOTIPOS

El paradigma de construcción de prototipos tiene tres pasos:

- Escuchar al cliente. Recolección de requisitos. Se encuentran y definen los objetivos globales, se identifican los requisitos conocidos y las áreas donde es obligatorio más definición.
- Construir y revisar la maqueta (prototipo).
- El cliente prueba la maqueta (prototipo) y lo utiliza para refinar los requisitos del software.

Este modelo es útil cuando:

- El cliente no identifica los requisitos detallados

- El responsable del desarrollo no está seguro de la eficiencia de un algoritmo, sistema operativo o de la interface hombre-máquina.

Ventajas del Modelo de Prototipo.

Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida. También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

Desventajas del Modelo de Prototipo.

Su principal desventaja es que una vez que el cliente ha dado su aprobación final al prototipo y cree que está a punto de recibir el proyecto final, se encuentra con que es necesario reescribir buena parte del prototipo para hacerlo funcional, porque lo más seguro es que el desarrollador haya hecho compromisos de implementación para hacer que el prototipo funcione rápidamente.

1.8 MODELOS EVOLUTIVOS

MODELO ESPIRAL

Es un ciclo de vida de software definido por Barry Boehm en 1988, utilizado mayormente en la ingeniería de software. Fue descrito por Boehm de la siguiente manera. Se tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software.

- Planificación: determinación de objetivos, alternativas y restricciones.
- Análisis de riesgo: análisis de alternativas e identificación/resolución de riesgos.
- Ingeniería: desarrollo del producto del "siguiente nivel",
- Evaluación del cliente: Valorización de los resultados de la ingeniería.

Características: Tiene y está conformado en un enfoque cíclico para el crecimiento del grado de definición e implementación de un sistema, mientras que disminuye su grado de riesgo. Utiliza un conjunto de puntos de fijación para asegurar el compromiso que asume el usuario con las soluciones de sistema que sean factibles y totalmente satisfactorias.

Ventajas:

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.

1. Reduce riesgos del proyecto
2. Incorpora objetivos de calidad
3. Integra el desarrollo con el mantenimiento, etc. Además, es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

Desventajas:

1. Genera mucho tiempo en el desarrollo del sistema
2. Modelo costoso
3. Requiere experiencia en la identificación de riesgos

CONCURRENTE

Características:

- Se expresa de manera esquematizada y organizada.
- Cada actividad lleva procesos concurrentes.
- Se aplica a la mayoría de tipos de desarrollo de software
- Es un módulo aplicable para el cliente soñador.
- Está dirigido básicamente y esencialmente a las necesidades del usuario.
- Es aplicable al cliente servidor.

INCREMENTAL

Características:

- Se evitan proyectos largos y se entrega “algo de valor” a los usuarios con cierta frecuencia.

- El usuario se involucre más.
- Difícil de evaluar el costo total.
- Difícil de aplicar a los sistemas transaccionales que tienden a ser integrados y a operar como un todo.
- Requiere gestores experimentados.
- Los errores en los requisitos se detectan tarde.
- El resultado puede ser muy positivo.

Ventajas:

- Con un paradigma incremental se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial.
- También provee un impacto ventajoso frente al cliente, que es la entrega temprana de partes operativas del Software.
- El modelo proporciona todas las ventajas del modelo en cascada realimentado, reduciendo sus desventajas sólo al ámbito de cada incremento.
- Permite entregar al cliente un producto más rápido en comparación del modelo de cascada.
- Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos.
- Por su versatilidad requiere de una planeación cuidadosa tanto a nivel administrativo como técnico.

Desventajas:

- El modelo Incremental no es recomendable para casos de sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido, y/o de alto índice de riesgos. Requiere de mucha planeación, tanto administrativa como técnica.
- Requiere de metas claras para conocer el estado de los proyectos.

Conclusión

Como en un principio mencione no sabía mucho con respecto a los subtemas pero en base a las investigaciones comprendí que si sabía pero no la definición concreta.

Básicamente es lo que he visto a lo largo de la carrera pero en diferentes términos y los he abarcado en otros temas y materias, espero que los temas hayan quedado bien explicados y que esta información le pueda servir a otra persona que al igual que yo no conozca sobre ello.

Puedo concluir llevándome un gran conocimiento adquirido ya que si sabía pero no sabía el término exacto pero ahora que ya lo se me siento con una gran satisfacción sobre este ensayo.

Fuente: <https://www.caracteristicas.co/software/#ixzz7MUkCcHOk>

<http://www.tiposdesoftware.com/historia-del-software.htm#:~:text=La%20primera%20teor%C3%ADa%20sobre%20el,Tukey%20en%201958.>

<http://ingenieriadesoftwareerigo.blogspot.com/2012/08/tecnicas-metodos-y-herramientas-para-el.html>

<http://www.monografias.com/trabajos108/modelos-del-proceso-del-software/modelos-del-proceso-del-software.>

<https://www.monografias.com/docs114/modelos-ingenieria-del-software/modelos-ingenieria-del-software#:~:text=Un%20m%C3%A9todo%20de%20ingenier%C3%ADa%20del,desarrolladores%20en%20los%20a%C3%B1os%2070.>