

UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

Tópicos selectos de ciencias de la ingeniería

Actividad 1. Ingeniería de Software

Matricula	Nombre
1548074	Cardona Pedro Antonio
1592335	García Roque Erika Yareli
1617998	Martínez Ramírez Jorge Antonio
1589361	Monsivais Puente Felipe Augusto
1561188	Padilla Reyes Leonardo Daniel
1592731	Quiroz Vela Abel
1587852	Robles Rincón Rodrigo
1719213	Rodríguez Garza Ana Lucia

Índice.

- ☐ **Introducción**
- ☐ **Alcance de la Ingeniería del software**
- ☐ **Conclusión**

Resumen.

La investigación habla sobre los alcances de la ingeniería de software la cual es una ingeniería joven y aún está en desarrollo. Estos alcances pueden ser de mucha ayuda en muchas áreas, ya sean dentro del desarrollo de software, la investigación y experimentación, líneas de productos, herramientas de desarrollo, etc.

También habla sobre el proceso de desarrollo y su ciclo de vida dentro de una empresa; los modelos de ciclo de vida para el desarrollo de software sirven para tener una manera eficiente de trabajar para obtener una mayor calidad en los proyectos que se crean dentro de la empresa, mejorando el proceso de desarrollo del software.

Introducción

Nos gustara iniciar respondiendo a las siguientes preguntas: ¿Qué es software?, ¿Qué es la ingeniería de software?, ¿En qué se diferencia la ingeniería de software de las ciencias de la computación?

Marco teórico.

Alcance de la ingeniería de software.

- Aspectos en la ingeniería de software:

Los aspectos ayudan a los ingenieros de software a lidiar con los atributos de calidad al proporcionar herramientas para añadir o quitar código repetitivo de muchas áreas en el código fuente. Los aspectos describen cómo todos los objetos o funciones deben comportarse en circunstancias particulares. Por ejemplo, los aspectos pueden agregar control de depuración, registro o bloqueo en todos los objetos de un tipo particular.

- Ágil:

El desarrollo ágil de software guía a los proyectos de desarrollo de software que evolucionan rápidamente con cambiantes expectativas y mercados competitivos. Algunas personas creen que las empresas y agencias exportan muchos de los puestos de trabajo que pueden ser guiados por procesos pesados. Conceptos relacionados incluyen la programación extrema, scrum y lean software development.

- Experimental:

La ingeniería de software experimental es una rama de la ingeniería de software interesada en la elaboración de experimentos sobre el software, en la recolección de datos de los experimentos y en la elaboración de leyes y teorías desde estos datos. Los proponentes de este método defienden que la naturaleza del software es tal que podemos hacer avanzar el conocimiento en software a través de sólo experimentos.

- Líneas de productos de software:

Las líneas de producción de software es una forma sistemática para producir familias de sistemas de software, en lugar de crear una sucesión de productos completamente individuales. Este método destaca una extensiva, sistemática, reutilización de código formal, para intentar industrializar el proceso de desarrollo de software.

- Model-driven:

El diseño manejado por modelos desarrolla modelos textuales y gráficos como artefactos primarios de diseño. Hay disponibles herramientas de desarrollo que usan transformación de modelo y generación de código para generar fragmentos de código bien organizado que sirven como base para producir aplicaciones completas.

Proceso del desarrollo de software.

Un proceso de software define el conjunto de tareas, que tienen que ser realizadas para producir un producto de software de alta calidad. En otras palabras, este es el enfoque que se toma para el desarrollo del software.

Es el proceso que se sigue para construir el producto de software desde la concepción de una idea, hasta la entrega y el retiro final del sistema.

Las características de un proceso de software se resumen a continuación:

- **Comprensión:** Este requiere claridad y declaración de la naturaleza explícita de la definición del proceso.
- **Visibilidad:** Se refiere a la capacidad de observar la salida de varias actividades del proceso, de manera que se mida el proceso del progreso.
- **Confiabilidad:** Se refiere a la capacidad del proceso para evadir errores o detectar errores y manejarlos antes de que estos avancen en el producto.
- **Robustez:** Se refiere a la capacidad del proceso de no detenerse a pesar de problemas inesperados.
- **Facilidad de mantenimiento:** Se refiere a la cantidad de modificaciones que pueden hacerse al sistema de software sin introducir errores.
- **Facilidad de verificación:** Un proceso es verificable si sus propiedades pueden ser fácilmente verificadas.
- **Rapidez:** Se refiere a la agilidad y rapidez del proceso para ser capaz de entregar un producto final a partir de las especificaciones.
- **Facilidad de soporte:** Se refiere a la posibilidad de que las actividades del proceso sean soportadas por un conjunto de herramientas automatizadas.
- **Facilidad de aceptación:** Se refiere a la capacidad del proceso a ser aceptado y usado por el equipo de ingenieros.
- **Facilidad de adaptación:** Se refiere a la capacidad del proceso a ser modificado para satisfacer las necesidades de cambio en el ambiente de desarrollo.

Después de haber discutido las características del proceso de desarrollo de software, se presenta a continuación las diferentes fases del proceso de desarrollo de software.

- **Fase de definición:** esta fase se concentra principalmente en que tiene que ser completado por el proceso de software.
- **Fase de desarrollo:** esta fase enfoca en el cómo los requerimientos de un sistema y el software serán completados.
- **Fase de mantenimiento:** esta fase se enfoca en cambio, el mantenimiento incluye la corrección de errores y la adaptación, conforme evoluciona el entorno del software.

Ciclo de vida del software.

El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final. El propósito de este programa es definir las distintas fases intermedias que se requieren para validar el desarrollo de la aplicación, es decir, para garantizar que el software cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo: se asegura de que los métodos utilizados son apropiados.

Estos programas se originan en el hecho de que es muy costoso rectificar los errores que se detectan tarde dentro de la fase de implementación. El ciclo de vida permite que los errores se detecten lo antes posible y, por lo tanto, permite a los desarrolladores concentrarse en la calidad del software, en los plazos de implementación y en los costos asociados.

Modelos de ciclo de vida del software.

Para facilitar una metodología común entre el cliente y la compañía de software, los modelos de ciclo de vida se han actualizado para reflejar las etapas de desarrollo involucradas y la documentación requerida, de manera que cada etapa se valide antes de continuar con la siguiente.

- **Modelo de cascada.**

El modelo de cascada es el modelo de paradigma más simple en desarrollo de software.

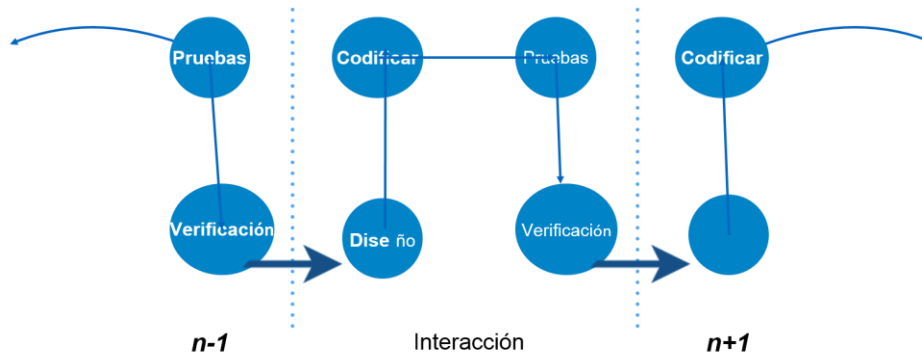


Sigue un modelo en que las fases del SDLC funcionarán una detrás de la otra de forma lineal. Lo que significa que solamente cuando la primera fase se termina se puede empezar con la segunda, y así progresivamente.

Este modelo es recomendable cuando el desarrollador ya ha diseñado y desarrollado softwares similares con anterioridad, y por eso está al tanto de todos sus dominios.

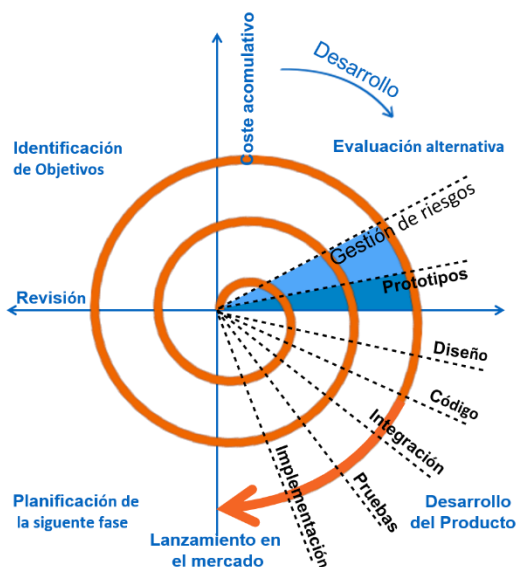
- **Modelo repetitivo.**

Este modelo guía el proceso de desarrollo de software en repeticiones. Proyecta el proceso de desarrollo de forma cíclica repitiendo cada paso después de cada ciclo en el proceso de SDLC.



- **Modelo en espiral.**

El modelo en espiral es una combinación de ambos modelos, el repetitivo y uno del modelo SDLC. Se puede ver como si se combina un modelo de SDLC combinado con un proceso cíclico (modelo repetitivo).

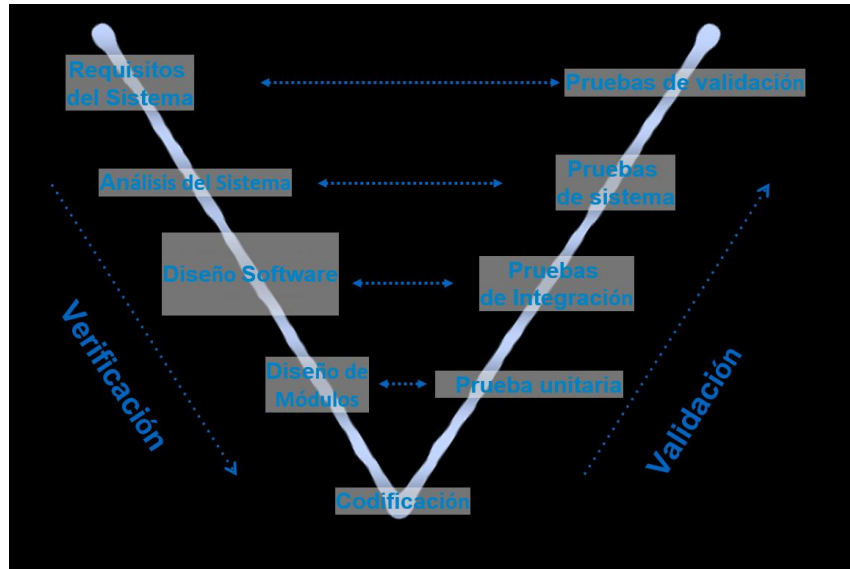


Este modelo considera el riesgo, factor que otros modelos olvidan o no prestan atención en el proceso. El modelo empieza determinando los objetivos y las limitaciones del software al inicio de cada repetición. En la siguiente etapa se crean los modelos de prototipo del software.

Esto incluye el análisis de riesgos. Luego un modelo estándar de SDLC se usa para construir el software. En la cuarta etapa es donde se prepara el plan de la siguiente repetición.

- Modelo V.

El mayor inconveniente del modelo de cascada es que solo se pasa a la siguiente fase cuando se completa la anterior, por tanto, no es posible volver atrás si se encuentra algún error en las etapas posteriores. El Modelo V aporta opciones de evaluación del software en cada etapa de manera inversa.



Conclusiones.

La ingeniería de software es sin lugar a dudas una fuente de soluciones a los problemas que las industrias enfrentan por lo que es, cómo se mencionó en la introducción, indispensable. La ingeniería de software no es sólo desarrollo de código, sino que lleva consigo una lista de fases donde intervienen incluso los usuarios, desde la especificación hasta el mantenimiento de un producto terminado. Habrá situaciones donde el cliente no sabe lo que quiere por lo que la ingeniería de software es abierta a cambios, en caso de que se quiere cambiar alguna definición, siempre y cuando se encuentre dentro de los tiempos establecidos, por lo que podemos definir la ingeniería de software como un proceso abierto.

Referencias bibliográficas.

- https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm
- <http://es.ccm.net/contents/223-ciclo-de-vida-del-software>
- <https://sg.com.mx/content/view/467>
- <http://brfranciscoosunaiuty.blogspot.mx/2012/07/proceso-de-desarrollo-de-software.html>
- <http://www.redalyc.org/html/1815/181517923005/>