EL CICLO DE VIDA DEL DESARROLLO DEL SOFTWARE

El ciclo de vida del desarrollo del software (también conocido como SDLC o Systems Development Life Cycle) contempla las fases necesarias para validar el desarrollo del software y así garantizar que este cumpla los requisitos para la aplicación y verificación de los procedimientos de desarrollo, asegurándose de que los métodos usados son apropiados.

FASES DE DESARROLLO DE SOFTWARE

Planificación

Antes de empezar un proyecto de desarrollo de un sistema de información, es necesario hacer ciertas tareas que influirán decisivamente en el éxito del mismo. Dichas tareas son conocidas como el fuzzy front-end del proyecto, puesto que no están sujetas a plazos.

Análisis del sistema

la etapa de análisis en el ciclo de vida del software corresponde al proceso a través del cual se intenta descubrir qué es lo que realmente se necesita y se llega a una comprensión adecuada de los requerimientos del sistema (las características que el sistema debe poseer).

Diseño de Software

Los inputs (aportacines) de los usuarios y los resultados de la recogida de información hecha en la fase anterior seran las aportaciones base de la fase actual. El output (o resultado) de esta etapa toma la forma de 2 diseños; El diseño lógico y el diseño físico.

Codificación

Esta fase también se puede denominar 'fase de programación'. La implementación del diseño de software empieza con el lenguaje de programación más conveniente, y desarrollando programas ejecutables y sin errores de manera eficiente.

Pruebas

Las pruebas de Software se hacen mientras se codifica y suelen hacerlo los desarrolladores y otros expertos evaluadores a varios niveles. Esto incluye evaluación de módulos, evaluación del programa, evaluación del producto, evaluación interna y finalmente evaluación con el consumidor final.

Integración

El Software puede necesitar estar integrado con las bibliotecas, Bases de datos o con otro u otros programas. Esta fase del SDLC se focaliza en la integración del software con las entidades del mundo exterior.

Implementación

El software necesita instalar configuraciones para el consumidor final con posterioridad. El Software se evalúa por su adaptabilidad y su portabilidad, en cuanto a las cuestiones relacionadas con la integración y conceptos asociados, se resuelven durante la implementación.

Disposición

Esta fase incluye archivar datos y componentes software requeridos, cierre del sistema, planificación de la actividad de disposición y terminación de sistema en el momento final del sistema.

Mantenimiento y Funcionamiento

El software se mantiene de forma temprana actualizando el código en acorde a los cambios que tienen lugar en entornos del usuario o tecnológicos. Esta fase puede que tenga que encarar retos originados por virus ocultos o problemas no identificados del mundo real.

Paradigma de desarrollo de Software

El paradigma de desarrollo software tiene su propio set de herramientas, métodos y procedimientos, los cuales son expresados de forma clara, y define el ciclo de vida del desarrollo del software.

MODELOS DEL DESARROLLO DE SOFTWARE

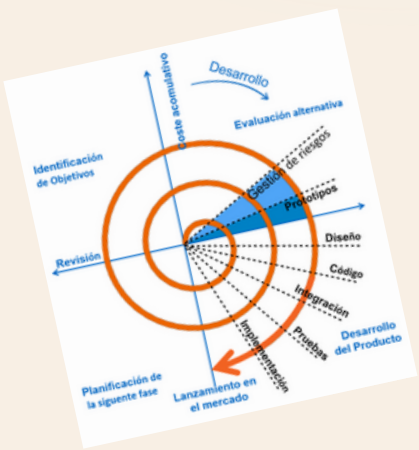
Con el fin de facilitar una metodología común entre el cliente y la compañía de software, los modelos de ciclo de vida (o paradigmas de desarrollo de software como la programación orientada a objetos) se han actualizado para plasmar las etapas de desarrollo involucradas y la documentación necesaria, de forma que cada fase se valide antes de continuar con la siguiente.

Modelo de cascada

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 asume que todo se lleva a cabo y tiene lugar tal y como se había planeado en la fase anterior, y no es necesario pensar en asuntos pasados que podrían surgir en la siguiente fase

Modelo repetitivo

El software primero se desarrolla en menor escala y se siguen y tienen en consideración todos los pasos. Entonces, por cada repetición, más módulos y características son diseñados, codificados, evaluados y añadidos al software. Cada ciclo produce un sotware completo, con más características y capacidad que los previos



Modelo en espiral

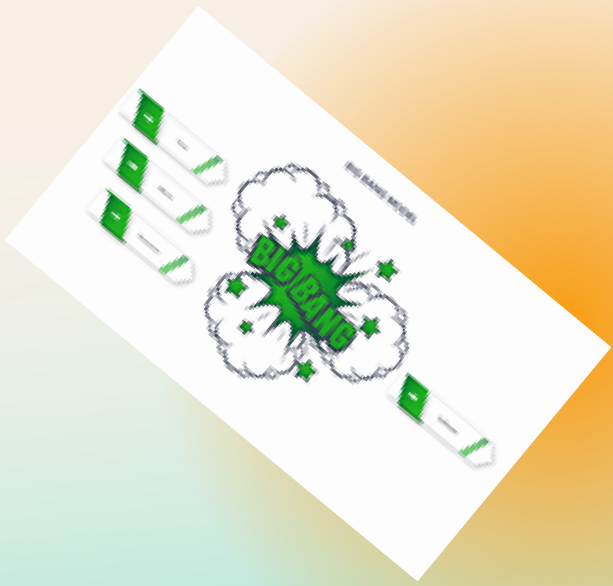
El modelo en espiral es una combinación de ambos modelos, el repetitivo y uno del modelo SDLC. Se puede ver como si se combina un modelo de SDLC combinado con un proceso cíclico (modelo repetitivo). Este modelo considera el riesgo, factor que otros modelos olvidan o no prestan atención en el proceso. 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.

Modelo V

El Modelo V aporta opciones de evaluación del software en cada etapa de manera inversa. En cada etapa, se crea la planificación de las pruebas y los casos de pruebas para verificar y validar el producto según los requisitos de la etapa. Esto hace que tanto la verificación como la validación vayan en paralelo. Este modelo también se conoce como modelo de validación y verificación.

Modelo Big Bang

Para este modelo, se requiere poca planificación. No sigue ningún proceso concreto, y a veces el cliente no está seguro de las futuras necesidades y requisitos. Por tanto la entrada o input respecto a los requisitos es arbitraria. Este modelo no es recomendable para grandes proyectos de software, pero es bueno para aprender y experimentar.



BIBLIOGRAFIAS

https://intelequia.com/blog/post/2083/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber
https://www.tutorialspoint.com/es/software_engineering/software_development_life_cycle.htm

Maria Guadalupe Felipe Hernandez