  

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN**

**SISTEMA UNIVERSIDAD ABIERTA Y EDUCACIÓN A DISTANCIA**

**Licenciatura en Informática**

**Procesos de ingeniería de software**

**Unidad 1 Actividad 2**

**Asignatura:**

Informática VII (ingeniería de software)

**Grupo:**

9790

**Alumno:**

Antonio López Sántiz

**Matricula:**

418018284

**e-mail:**

lopezsantizantonio84@gmail.com

**Asesor:**

Eva Luz Zapata Nava

\*\* \*\*

**Procesos de ingeniería 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.

Algunas de las tareas de esta fase incluyen actividades como la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados, la estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las diferentes etapas del proyecto.

* Análisis

Hay que averiguar qué es exactamente lo que tiene que hacer el software. Por eso, 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.

* Diseño

En esta fase se estudian posibles opciones de implementación para el software que hay que construir, así como decidir la estructura general del mismo. El diseño es una etapa compleja y su proceso debe realizarse de manera iterativa.

* Implementación

En esta fase hay que elegir las herramientas adecuadas, un entorno de desarrollo que facilite el trabajo y un lenguaje de programación apropiado para el tipo de software a construir. Esta elección dependerá tanto de las decisiones de diseño tomadas como del entorno en el que el software deba funcionar.

* Pruebas

La fase de pruebas del ciclo de vida del software busca detectar los fallos cometidos en las etapas anteriores para corregirlos. Por supuesto, lo ideal es hacerlo antes de que el usuario final se los encuentre. Se dice que una prueba es un éxito si se detecta algún error.

* Instalación o despliegue

La siguiente fase es poner el software en funcionamiento, por lo que hay que planificar el entorno teniendo en cuenta las dependencias existentes entre los diferentes componentes de este.

Es posible que haya componentes que funcionen correctamente por separado, pero que al combinarlos provoquen problemas. Por ello, hay que usar combinaciones conocidas que no causen problemas de compatibilidad.

* Uso y mantenimiento

Esta es una de las fases más importantes del ciclo de vida de desarrollo del software. Puesto que el software ni se rompe ni se desgasta con el uso, su mantenimiento incluye tres puntos diferenciados:

* Eliminar los defectos detectados durante su vida útil (mantenimiento correctivo).
* Adaptarlo a nuevas necesidades (mantenimiento adaptativo).
* Añadirle nuevas funcionalidades (mantenimiento perfectivo).

**Tipos de metodología en el desarrollo de software**

* Desarrollo ágil

El desarrollo ágil de software se refiere a un grupo de metodologías de desarrollo de software basadas en el desarrollo iterativo, donde los requisitos y las soluciones evolucionan a través de la colaboración entre equipos multifuncionales autoorganizados.

En concreto, las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo pequeñas piezas de software en funcionamiento para aumentar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un enfoque sencillo de la documentación de software y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos.

* Desarrollo en cascada

El modelo en cascada es un enfoque de desarrollo secuencial, en el que se considera que el desarrollo fluye constantemente hacia abajo (como una cascada), a través de varias fases, las cuales típicamente son las siguientes:

* Análisis de requisitos que resulta en una especificación de requisitos de software
* Diseño del software
* Implementación
* Testeo
* Integración, si hay múltiples subsistemas
* Despliegue (o Instalación)
* Mantenimiento

Con tantos puntos a favor y un rendimiento de máxima calidad, es de esperar que cientos de empresas y emprendimientos hayan utilizado al menos una vez en la vida el modelo en cascada. En la siguiente lista se enumera algunos de los tipos de proyectos más importantes que han utilizado las fases del modelo en cascada:

* Desarrollo de softwares para la industria automotriz.
* Creación de sistemas de gestión de recursos humanos.
* Diseño de sistemas de gestión de la cadena de suministros.
* Avances en los sistemas de control de las instalaciones nucleares.
* Estudios y proyectos sobre transbordadores espaciales.
* Desarrollo de espiral

Es uno de los modelos de ciclo de vida de desarrollo de software más importantes, que proporciona soporte para la gestión de riesgos. Su nombre es debido a que en su representación esquemática parece un espiral con muchos bucles.

Cada proyecto tiene su propio número de bucles, y por otro lado cada ciclo de la espiral se llama Fase del proceso de desarrollo de software.

Los principios básicos son:

* Se centra la atención en la evaluación de riesgos y en minimizar el riesgo del proyecto al dividir un proyecto en segmentos más pequeños y brindar mayor facilidad de cambio durante el proceso de desarrollo, así como brindar la oportunidad de evaluar los riesgos y evaluar la continuación del proyecto a lo largo del ciclo de vida.
* Cada ciclo implica una progresión a través de la misma secuencia de pasos, para cada parte del producto y para cada uno de sus niveles de elaboración, desde un documento de concepto de operación general hasta la codificación de cada programa individual.​
* Cada viaje alrededor de la espiral pasa por cuatro cuadrantes básicos:

1. Determinar objetivos, alternativas, y limitaciones de la iteración;

2. Evaluar alternativas; Identificar y resolver riesgos;

3. Desarrollar y verificar entregables de la iteración;

4. Planear la iteración próxima.

* Empieza cada ciclo con una identificación de las partes interesadas y sus condiciones de victoria, y finaliza cada ciclo con una revisión y compromiso.

Fases

Sus cuatro fases principales son:

* Determinar objetivos
* Alternativas
* Restricciones
* Fase de planificación

# Bibliografía

*¿Qué es la metodología ágil?* (19 de julio de 2022). Obtenido de https://www.redhat.com/es/devops/what-is-agile-methodology#:~:text=En%20concreto%2C%20las%20metodolog%C3%ADas%20%C3%A1giles,equipo%20para%20ofrecer%20mejoras%20constantes.

*El método espiral de desarrollo de software*. (s.f.). Obtenido de https://www.tecnologias-informacion.com/metodo-espiral.html

*El modelo en cascada: desarrollo secuencial de software*. (21 de 03 de 2019). Obtenido de https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/