

{Método de proceso de software}

Cascada o lineal

- Castro Cruces Jorge Eduardo
- Mendoza Cuellar José Oscar
- Pérez Aguilar Ariadna Jaqueline

Méndez Segundo Laura
3CM13
Ingeniería de software
Lunes, 14 de Febrero de 2022



Contenido

- Definición.
- Características.
- Etapas.
- Ventajas y desventajas.
- Ejemplos.
- Conclusión.
- Referencias.



Definición

El primer modelo publicado sobre el proceso de desarrollo de software se derivó a partir de procesos más generales de ingeniería de sistemas (**Royce, 1970**). Este modelo se ilustra en la figura 1. Debido al paso de una fase en cascada a otra, este modelo se conoce como “modelo en cascada”, ciclo de vida del software o waterfall. El modelo en cascada es un ejemplo de un proceso dirigido por un plan; en principio, usted debe planear y programar todas las actividades del proceso, antes de comenzar a trabajar con ellas.[1]

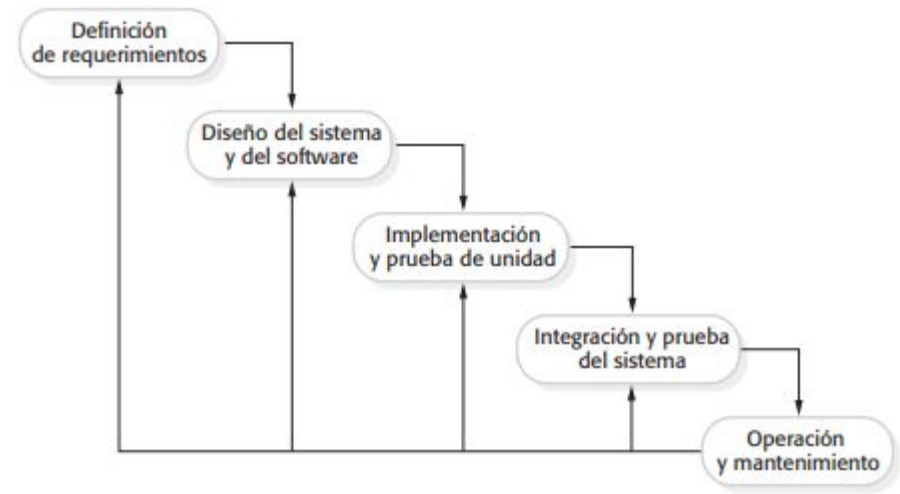


Figura 1. Modelo de cascada[1]



Características

- La siguiente fase no debe comenzar sino hasta que termine la fase previa.[1]
- En la práctica, dichas etapas se traslapan y se nutren mutuamente de información.[1]
- Durante el diseño se identifican los problemas con los requerimientos.[1]
- En la codificación se descubren problemas de diseño.[1]
- El proceso de software implica retroalimentación de una fase a otra.[1]

- Los problemas se dejan para una resolución posterior, se ignoran o se programan.[1]
- El freno prematuro de los requerimientos quizá signifique que el sistema no hará lo que el usuario desea.[1]
- El modelo en cascada es consecuente con otros modelos del proceso de ingeniería y en cada fase se produce documentación.[1]



Etapas

Fase de ingeniería y análisis del sistema.

Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.[2]

Fase de análisis de los requisitos.

Se analizan las necesidades de los usuarios finales del software a desarrollar para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (Documento de Especificación de Requisitos), que contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos.[2]

Fase de diseño.

Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Surge el SDD (Documento de Diseño del Software), este contiene la descripción de la estructura global del sistema y la especificación de lo que debe hacer cada una de sus partes.[2]

Fase de codificación.

Es la fase de programación propiamente dicha.

Aquí se desarrolla el código fuente, haciendo uso de prototipos así como pruebas y ensayos para corregir errores. Dependiendo del lenguaje de programación y su versión, se crean las librerías y componentes reutilizables dentro del mismo proyecto para hacer que la programación sea un proceso mucho más rápido.[2]

Fase de Pruebas.

Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente antes de ser puesto en explotación. Una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.[2]

Fase de Mantenimiento.

Es una de las fases finales del proyecto. En el desarrollo surgen cambios, para corregir errores o bien para introducir mejoras. El software sufre cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento. [2]



Ventajas y desventajas



Ventajas.

- La cantidad de recursos necesarios para implementar este modelo es mínimo.^[3]
- La documentación se produce en cada etapa del desarrollo del modelo de cascada.
- Planificación sencilla.^[3]
- Después de cada etapa importante de la codificación de software, las pruebas se realizan para comprobar el correcto funcionamiento del código.^[3]
- La calidad del producto resulta alta.^[3]



Desventajas.

- No se puede volver atrás, si algunas de las fases han resultado mal, las cosas pueden ser muy complicado en la fase de ejecución.[3]
- Los pequeños cambios o errores que surgen en el software completo puede causar mucho problema.[3]

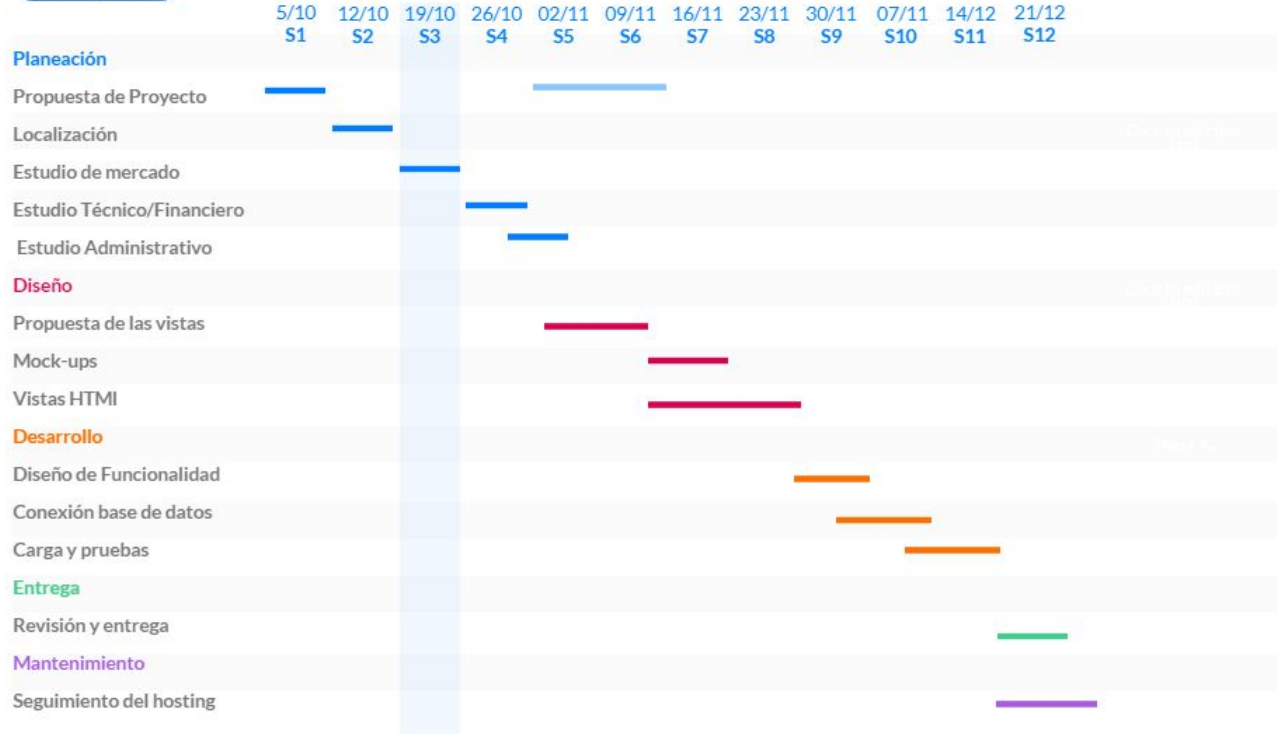


Ejemplos

Los Bolcheviques

App Para el cuidado de la salud

Dic 21 | 2021





Conclusiones



El método en cascada:

- Es de los métodos más sencillos para el desarrollo de software.
- Ordena rigurosamente las etapas del ciclo de software.
- Es exitoso cuando se tienen bien especificados los requerimientos del software.
- Permite una organización más fácil de comprender siempre y cuando no se mezclen las diferentes fases.

Referencias

- [1] Sommerville, I. (2012). *Ingeniería de software* (Spanish Edition) by Ian Sommerville (9.a ed.). Pearson (México). https://sistemamid.com/panel/uploads/biblioteca/2018-06-11_03-37-12144643.pdf
- [2] *Procesos para la ingeniería de software*, Facultad de Estadística e Informática, Recuperado el 11 de febrero de 2022, de https://www.uv.mx/personal/ermeneses/files/2018/02/Clase7-Modelos_de_procesos_de_desarrollo_de_software.pdf
- [3] *Modelo en cascada*, Recuperado el 11 de febrero de 2022, de <https://sites.google.com/site/proyectoadpmodelosdedesarrollo/home/modelo-en-cascada/modelo-en-cascada-ventajas-y-desventajas>