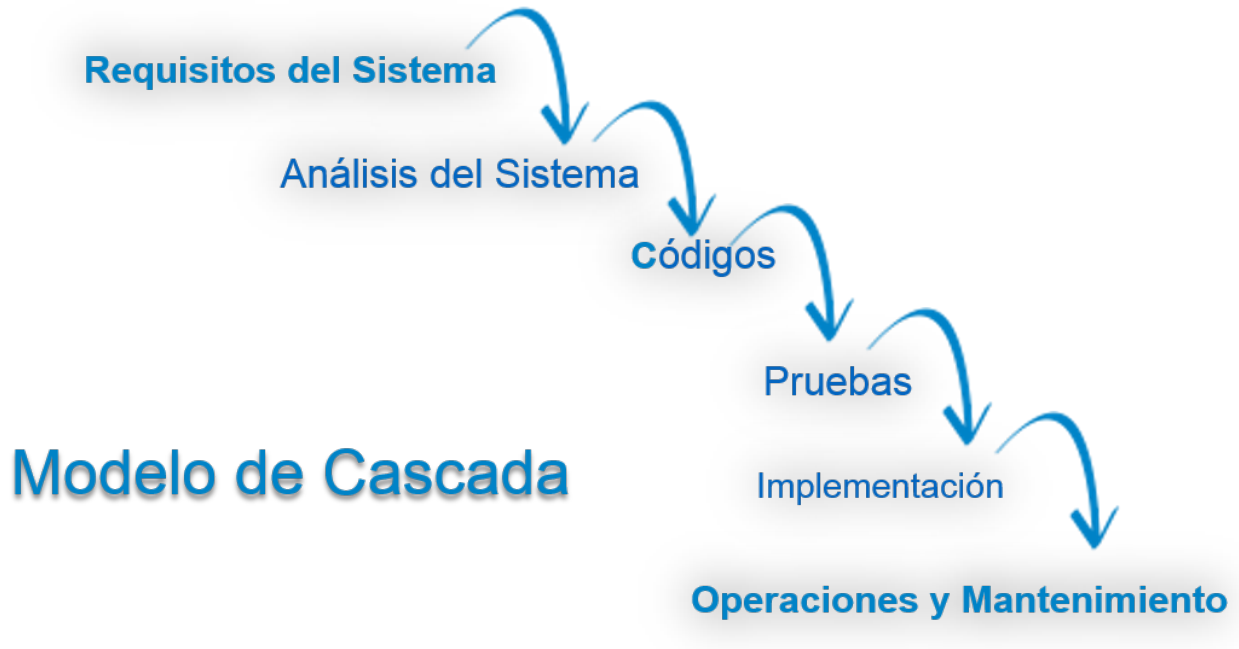
Tema 2-Ciclos de vida

Codificación directa

Se llama también Code and Fix. Es una práctica muy mala y se considera antónimo de la ingeniería del software. Solo sirve para proyectos enanos. Consiste en hacer un software, probarlo y corregir los bugs. Esto se hace en bucle hasta considerar que todo está listo.

Cascada

Apareció como alternativa al Code and Fix. Es uno de los más populares. Pone énfasis especialmente en la realización de actividades de definición de requisitos y documentación de análisis y diseño antes de pasar a la codificación. Es totalmente lineal. Cada fase de la línea genera productos (output) que sirven de base (input) a la siguiente fase de desarrollo.



Ventajas:

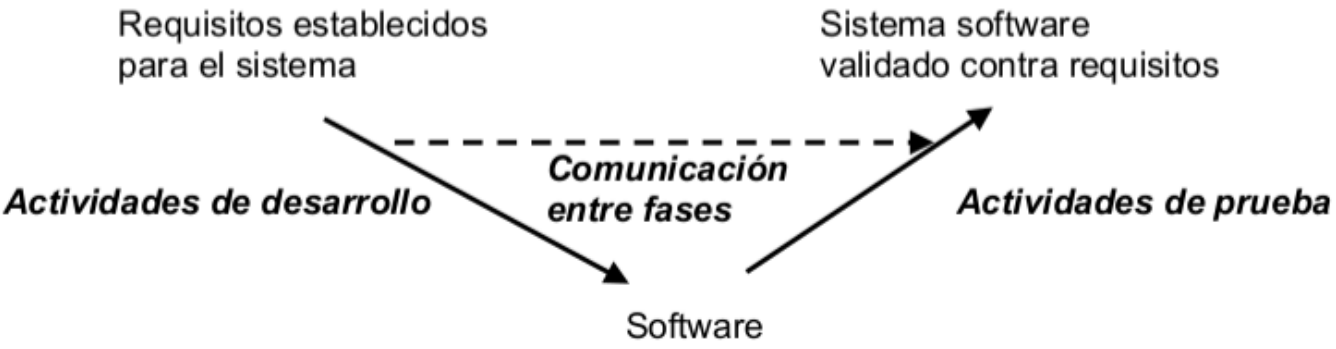
- Sirve como buen marco para colocar todas las actividades de desarrollo software.
- Es muy estructurado e impone pautas claras de trabajo.
- Facilita la coordinación de los recursos utilizados.
- Facilita el seguimiento del proyecto.
- Facilita la detección de desviaciones y las correcciones.
- Proporciona entregables intermedios.

Inconvenientes:

- Establece todos los requisitos de un sistema ya al principio.
- Es extremadamente rígido.
- Muchos de los problemas se detectan tarde.
- Los productos intermedios son simplemente documentos.

Ciclo de vida en V

Es una mejora del modelo cascada. Se separa en 2 ramas (descendente y ascendente) y cada fase de una rama tiene su correspondencia en la otra siguiendo este esquema:



Un ejemplo sería este:

Oferta----------Mantenimiento
Requisitos----------Pruebas de aceptación
Diseño de arquitectura----------Pruebas de integración
Software

Ventajas:

- Hereda las ventajas del cascada
- Favorece la consideración de las pruebas lo antes posible mediaante esa "horizontalidad".

Inconvenientes:

- Hereda los inconvenientes del casscada

Modelo de prototipos

Se desenvuelve en 3 fases cíclicas: el cliente habla, el desarrollador construye el prototipo y el cliente lo prueba. Es adecuado cuando no se sabe exactamente lo que desea el cliente, cuando el desarrollador no está seguro de la eficiencia de una aproximación de su software, etc.

Facilita bastante al desarollador hacer un buen modelo del software a desarrollar. Los prototipos pueden ser desechables o evolutivos.

Ventajas:

- Es muy buena forma de extraer requisitos cuando no están del todo claros.

Inconvenientes:

- El cliente puede valorar los prototipos como un software ya funcional.
- El desarrollador puede tomar decisiones de implementación que no son las mejores para el sistema final o incluyo llegar a dejarlas en el sistema.

Orientados a objetos

Son tanto iterativos como incrementales. Las fases no están delimitadas por fronteras. Se incorporan bibliotecas de clases y otros componentes que son reutilizables.