

# •CICLOS DE VIDA DEL SOFTWARE

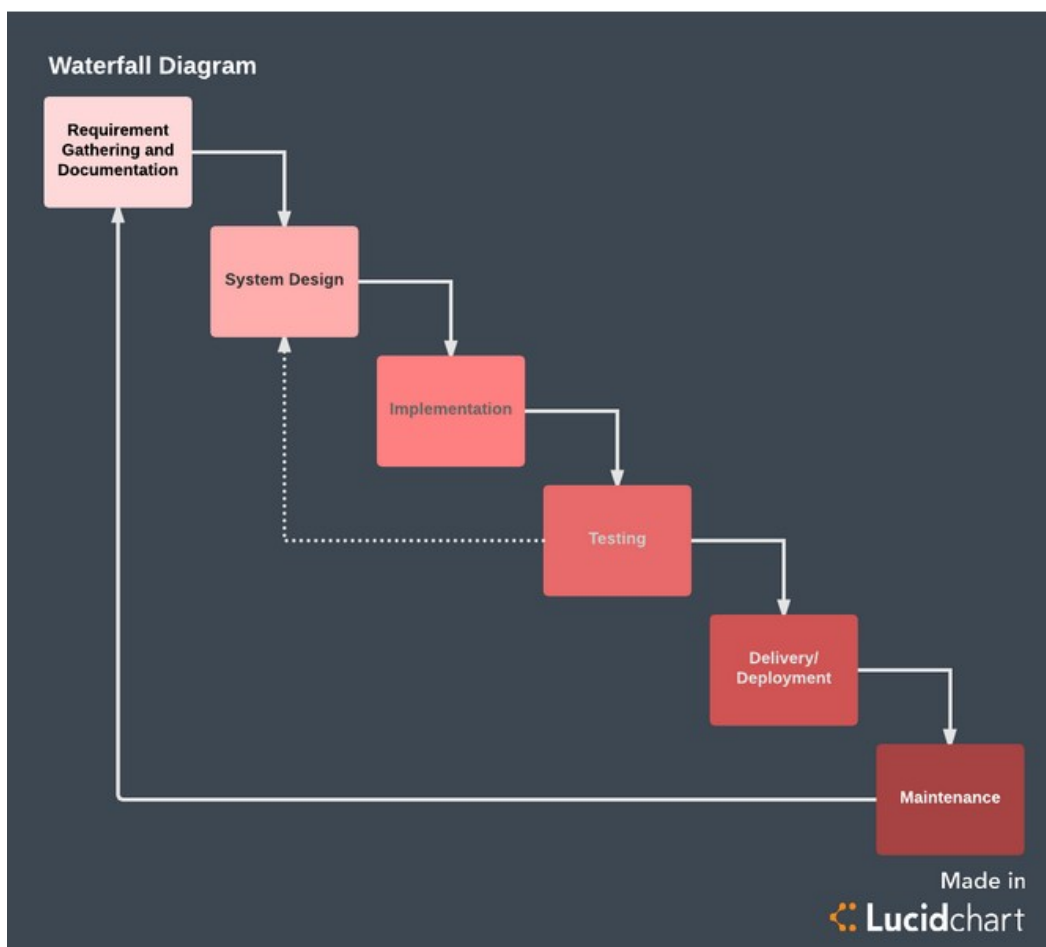
Óscar Úbeda

## •MODELO EN CASCADA

Pequeños desarrollos, ya que las etapas pasan de una a otra sin retorno posible

**-Ventajas:** Utiliza una estructura clara lo que hace que los pasos a seguir estén claros y definidos, determina el objetivo final rápidamente, donde se conoce el objetivo general desde el comienzo, se transmite bien la información ya que el enfoque que se le da es sumamente metódico

**-Desventajas:** Dificulta los cambios o las revisiones ya que este no permite ningún lugar para estos, excluye al cliente o al usuario final ya que este proceso se centra muy poco en el usuario o cliente final de un proyecto, retrasa las pruebas hasta después de la finalización ya que este insiste hasta el paso cuatro de seis para probar sus productos

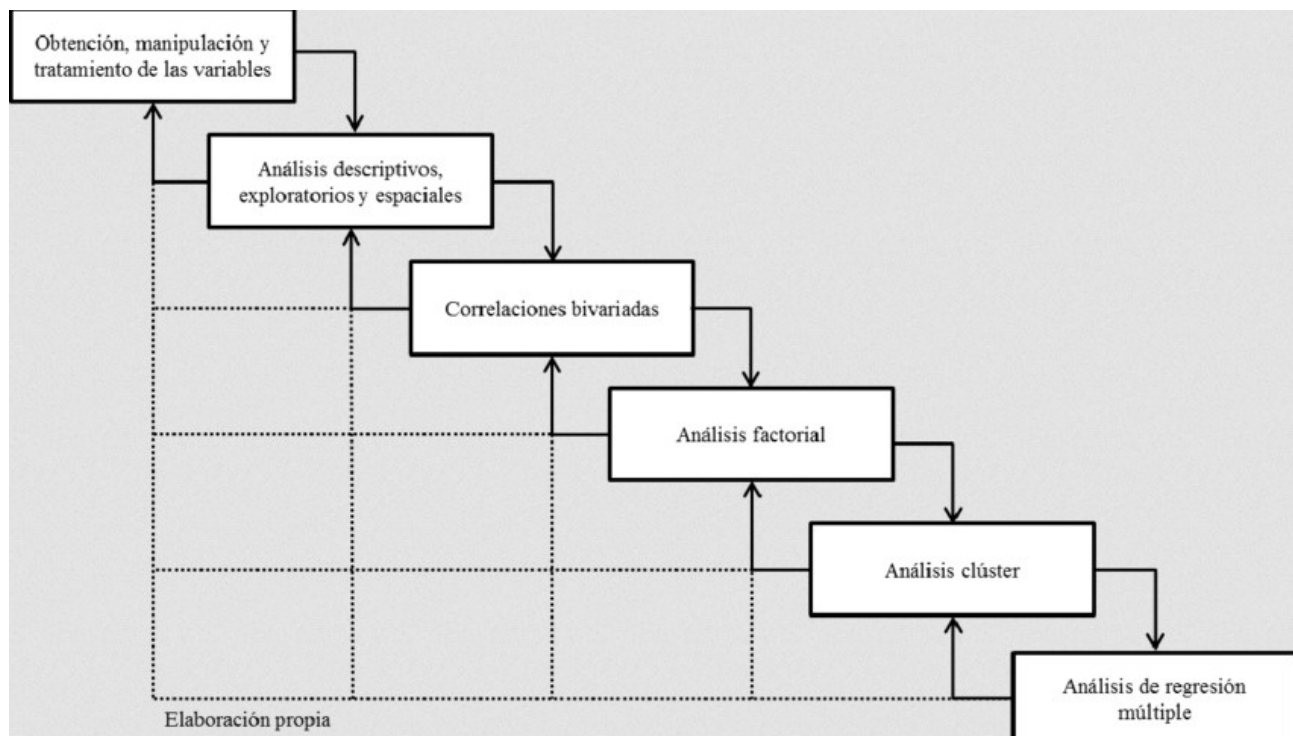


## •MODELO EN CASCADA CON RETROALIMENTACIÓN

Es uno de los modelos más utilizados. Proviene del modelo anterior, pero se introduce una realimentación entre etapas, de forma que podamos volver atrás en cualquier momento para corregir, modificar o depurar algún aspecto.

**-Ventajas:** Fácil de comprender, de planificar y de seguir, y existen herramientas que lo soportan.

**-Desventajas:** Los cambios generan confusiones conforme el proyecto avanza, es difícil para el cliente especificar todos los requisitos al principio, cuanto más tarde se detecte un fallo más costoso serán hacer los cambios que implique (rehacer todas las fases).

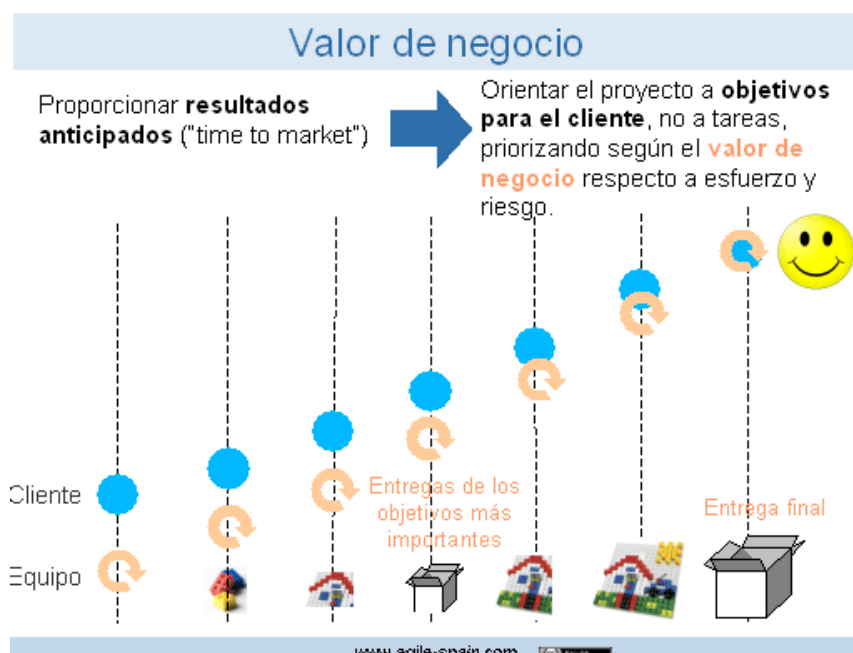


## •MODELO ITERATIVO INCREMENTAL

Para resolver el problema del tiempo del modelo en cascada, podemos ir haciendo entregas con una funcionalidad parcial del software al cliente y aumentarla en entregas posteriores.

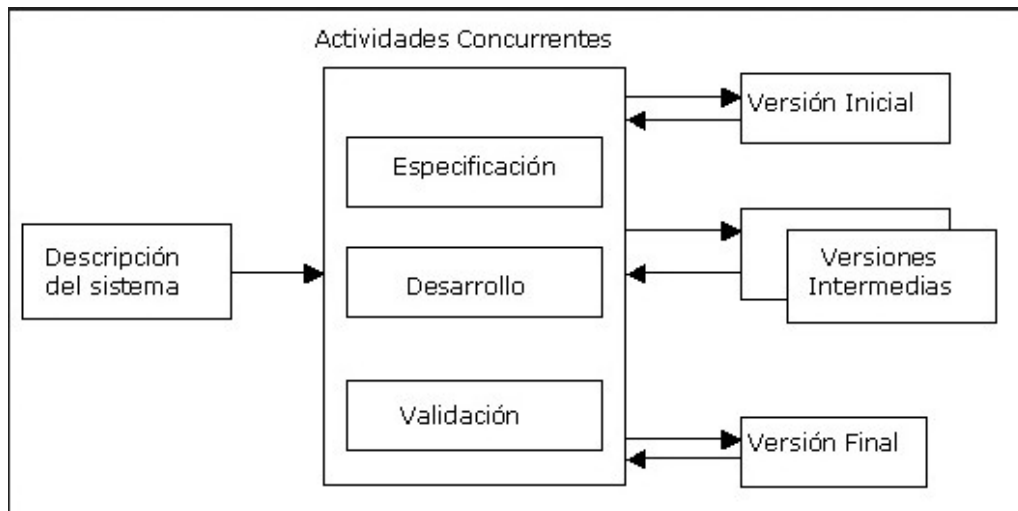
### Modelo iterativo incremental

Es decir, se realiza el modelo anterior (cascada) eliminando la retroalimentación y haciendo los cambios necesarios en cada incremento y entrega.



## •MODELOS EVOLUTIVOS

Son más modernos que los anteriores. Tienen en cuenta la naturaleza cambiante y evolutiva del software.



## •CONSTRUCCIÓN DE PROTOTIPOS

En la práctica, la construcción de prototipos ayuda al ingeniero de sistemas y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

**-Desventajas:** El cliente considera la mayoría de las veces al prototipo como el producto final. Es conveniente seguir este modelo cuando el cliente es capaz d definir un conjunto de objetivos generales para el software, pero no puede identificar los requerimientos en detalle o cuando los desarrolladores tienen dudas considerables.

## •MODELO EN ESPIRAL

Es una combinación del modelo anterior con el modelo en cascada añadiendo un nuevo elemento, el análisis de riesgo.

**-Siempre hay una serie de etapas que debemos seguir para construir software fiable y de calidad.**

- **Análisis de requisitos:** Se especifican los requisitos funcionales y no funcionales del sistema.
- **Diseño:** Se divide el sistema en partes y se determina la función de cada una.
- **Codificación:** Se elige un lenguaje de programación y se codifican los programas.
- **Pruebas:** Se prueban los programas para detectar errores y se depuran.
- **Documentación:** De todas las etapas, se documenta y guarda toda la información.
- **Explotación:** Instalamos, configuramos y probamos la aplicación en los equipos del cliente.

- **Mantenimiento:** Se mantiene el contacto con el cliente para actualizar y modificar la aplicación el futuro.

- **Autoevaluación**