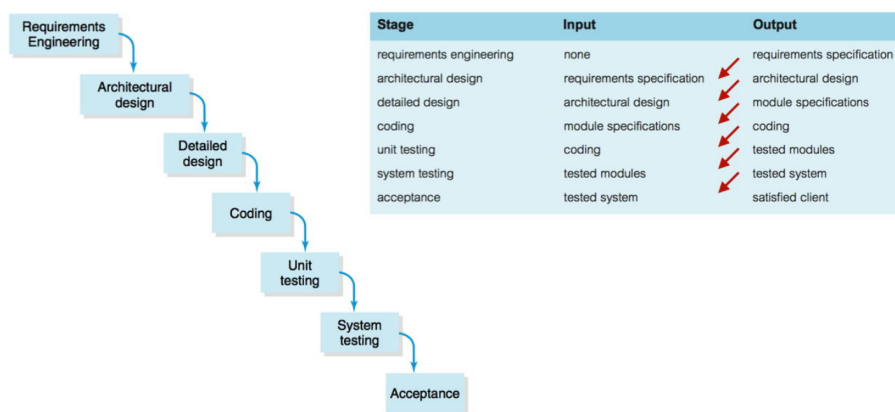


# CASCADA

## Definición

Modelo de proceso que fue utilizado por más de 30 años, es conceptualmente fácil de entender, pero trae consigo algunos problemas.

Su nombre viene del hecho de que cada una de sus etapas tiene un output bien definido el cual “cae” a la siguiente a modo de input (Ver imagen) por lo que se genera un desarrollo **secuencial que avanza en una sola dirección**.



## Ventajas

Como es un modelo secuencial permite ver claramente cuál es el estado actual de desarrollo de software, donde sabemos de manera visual cuál es el input que entra a una etapa, y cuál es el output que recibimos.

## Desventajas

No funciona en la práctica (a menos que sea algo muy corto), ya que en el área del desarrollo de Software nos enfrentamos inevitablemente a cambios en los requisitos o necesidades y con este modelo tendríamos que cambiar todo el modelo para cumplir una nueva necesidad, por lo que en la realidad se usan modelos de carácter iterativo.

# SCRUM

## Definición

En este modelo el desarrollo se separa en distintas etapas o iteraciones denominadas Sprints. Estos están bien estructurados, teniendo roles y una duración determinada (por ejemplo, 3 semanas).

Existen 3 roles fundamentales (además de únicos):

- **Product Owner**, está a cargo del producto, indicando que cosas se le deben incorporar y se responsabiliza del éxito o fracaso del mismo.
- **ScrumMaster**, cuya labor consiste en resolver problemas y ser el coach en cuanto a cómo trabajar.
- **Team**, quienes desarrollan el producto propiamente tal.

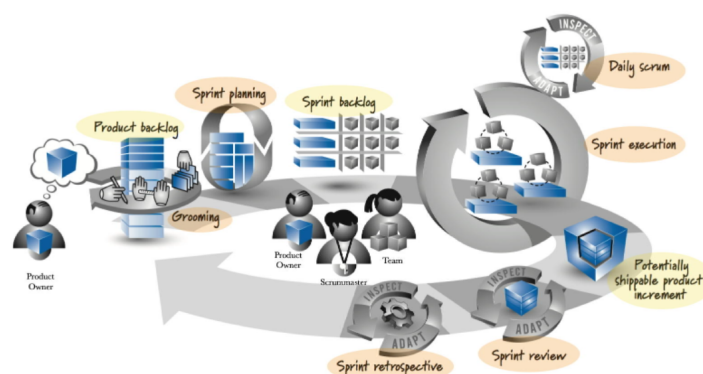
## Actividades

El **Sprint Planning** consiste en una reunión entre el equipo, Product Owner, interesados, sponsors, e incluso clientes, esto con el objetivo de planificar y priorizar las tareas a realizar durante el sprint y resolver las primeras dudas asociadas. Por otro lado, el **Sprint Review** es la misma reunión pero con el objetivo de obtener feedback de buena calidad relacionada al negocio, esta reunión debe realizarse al término de cada Sprint.

La **Daily Meeting o reunión de pie** consiste en una reunión diaria donde en no más de 15 minutos todos indican en qué situación están, qué hicieron ayer, qué harán hoy y si hay algún problema, para de esta manera buscar apoyo o generar cambios en el desarrollo.

## Ventajas

Dentro de las grandes ventajas que posee, destacan que cada Sprint genera un producto que es potencialmente entregable dado a su carácter incremental, osea, siempre estar sumando cosas valiosas, y que permite solucionar de manera efectiva cambios en requisitos iniciales.



# EXTREME PROGRAMMING

## Definición

Este modelo de proceso consiste en una serie de buenas prácticas que permiten un desarrollo ameno y más eficaz.

## Prácticas

### Pair-Programming

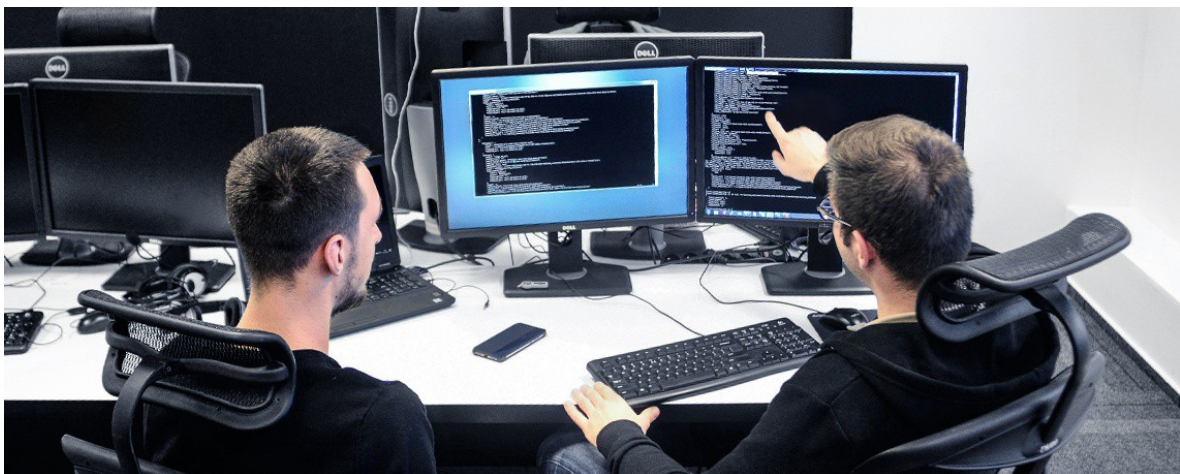
La más destacable es **Pair-Programming**, que consiste en que nadie programa solo, siempre de a pares con únicamente 1 teclado y 1 mouse, ambos utilizados por solo 1 persona que actúa de conductor, mientras que el par, hace de revisor.

**Ventaja:** Esta simple acción ha logrado demostrar ser de las mejores prácticas en cuanto a su efectividad ya que permite un código más legible, de calidad y una rápida detección de errores.

### Cliente en el equipo

Otras características que posee es tener al **cliente como parte del equipo**, quien además especifica casos de prueba, y un constante testeo del producto, asegurando su correcto funcionamiento.

**Ventaja:** Permite detectar si el desarrollo que se está generando va por buen camino y si no, permite realizar cambios durante el desarrollo, no cuando el producto esté finalizado.



# KANBAN

## Definición

El nombre tiene un origen japonés Kan - visual y Ban - tarjeta. Lo que distingue a este modelo de otros es que se utiliza un gran tablero donde se ubican tarjetas con las tareas en distintas etapas de avance, adicionalmente se debe definir el límite de tarjetas a trabajar en paralelo para cada etapa.

## Ventajas

Algunas de las ventajas de este modelo es que permiten visualizar qué se hará hoy, limitar el trabajo en progreso, y optimizar el flujo de tareas desde el “por hacer” hacia el “terminadas”.

Además, te permiten medir el tiempo que demora hacer cada tarea, ya que se puede medir cuánto tiempo le toma a un desarrollador llevar una tarea desde “por hacer” hasta “terminada”.

## Desventajas

Como está definido Kanban, no hay iteraciones de un largo fijo, ni hay tiempo definido.

