

Indice

- [Modelo de proceso](#)
- [Metodologia](#)
- [Proceso de desarrollo de software](#)
- [Modelos de desarrollo](#)
 - [Modelo de cascada](#)
 - [Puntos a favor](#)
 - [Puntos en contra](#)
 - [Prototipo](#)
 - [Etapas de prototipo](#)
 - [Puntos a favor](#)
 - [En contra](#)
 - [Modelo en espiral](#)
 - [Puntos a favor](#)
 - [En contra](#)
 - [El modelo de procesos](#)
 - [A favor](#)
 - [En contra](#)
 - [Desarrollo incremental](#)
 - [Puntos a favor](#)
 - [Puntos en contra](#)

Modelo de proceso

El modelo de proceso por si solo, no nos sirve para construir software. Esto se debe a que nos va a faltar mucha informacion. Aun asi, el aporte del modelo de proceso es muy importante. Este nos dice como interactuan las etapas entre si.

Es el gran marco de un software. No nos sirve para la construccion de uno, por la falta de detalle que tiene.

Metodologia

Le agrega informacion al modelo de proceso. Que info? Le agrega los roles, las actividades que se hacen, los productos que se construyen y el orden en el que se hacen las actividades. Es un marco mucho mas definido para la construccion de software, ya que contiene mucho mas detalle.

Es el marco de modelo de proceso, pero con muchos mas detalles.

A veces puede ser muy grande, por lo que empiezan a aparecer subconjuntos de las metodologias. Eso es lo que se llama **proceso de desarrollo de software**.

Proceso de desarrollo de software

El proceso de desarrollo de software, no es mas que la metodologia adaptada a un proyecto existente en la realidad.

Tailorization o **parametrizacion**, o **tailoring** es personalizar una metodologia para lograr el proceso de desarrollo de software.

Modelos de desarrollo

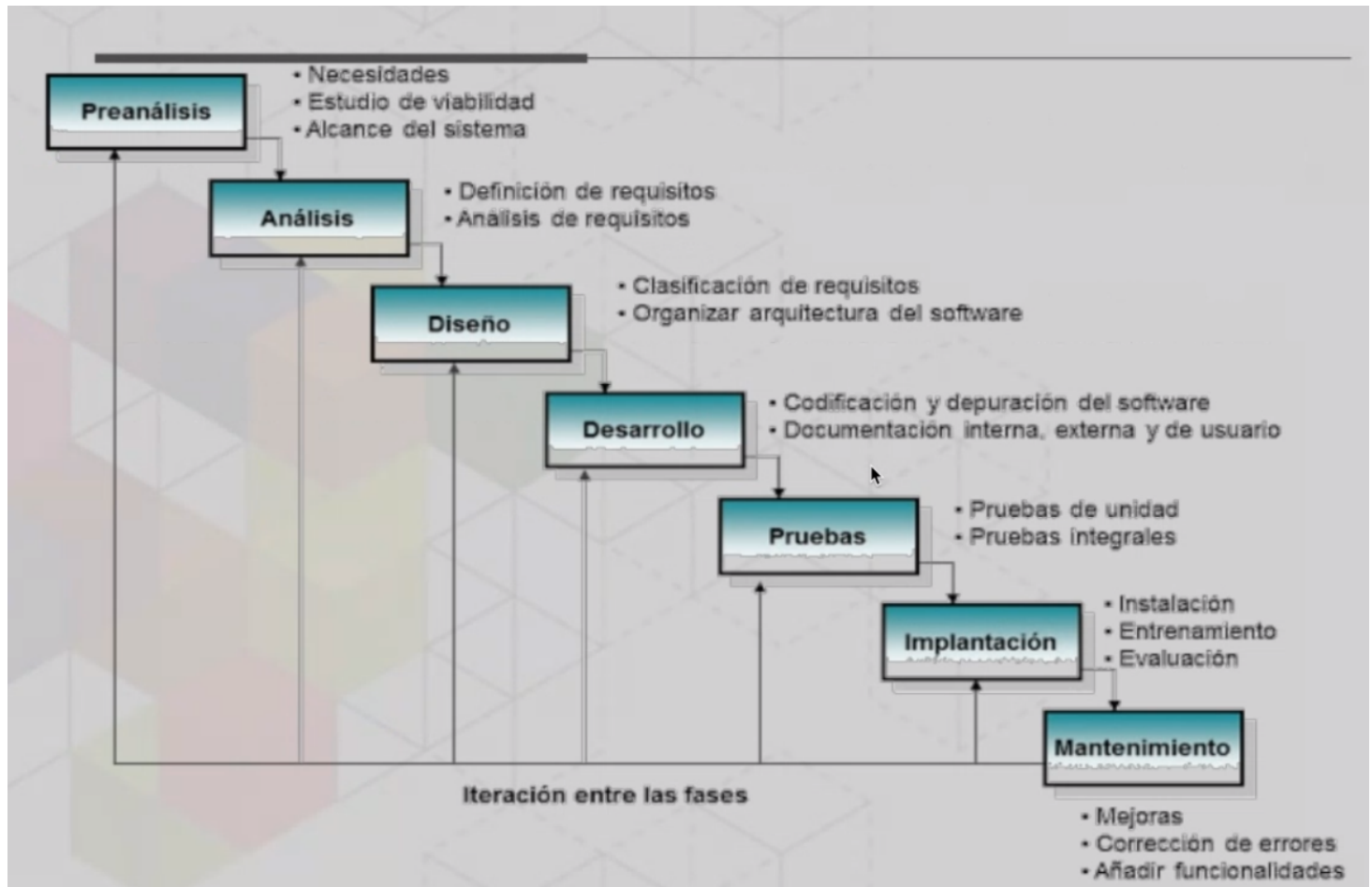
- Modelo de cascada
- Modelo en V
- Prototipos
- Modelo de espiral
- Modelo de procesos
- Desarrollo incremental

Modelo de cascada

La cascada es un modelo de procesos que tiene una secuencia ordenada. Es un conjunto de actividades que está una a continuacion de la otra.

- Secuencia ordenada.
- El trabajo de una etapa previa es la entrada del siguiente proceso.
- Provee de un gran control sobre las fechas de entrega y entregables.
- Establece criterios de entrada y salida en cada fase claramente definidos.
- Dado que provee pocos puntos de visibilidad, da la impresion de que es lento.

No se ve lo que pasa adentro del proyecto, hasta que termine el proyecto. Por eso da la sensación que es lento.



Puntos a favor

- Excelente cuando se tiene un producto estable y se conoce la tecnología.
- Es un método muy estructurado que funciona bien con gente de poca experiencia.
- Provee estabilidad en los requerimientos.
- La planeación se puede hacer anticipadamente.

Puntos en contra

- Tiene poca flexibilidad.
- Los proyectos en la práctica raramente siguen un flujo secuencial.
- Siempre es difícil para el cliente mostrar todos los requerimientos explícitamente, y con mucha anticipación.
- El cliente debe tener paciencia, debido a que no se ven resultados hasta terminar.

Prototipo

Un prototipo es una version preliminar de un sistema de informacion con fines de demostracion o evaluacion. Dicho prototipo evoluciona a medida que habla con el cliente.

La característica principal es que al prototipo se lo va validando con el tiempo.

- Es un metodo menos formal de desarrollo.
- El prototipo es una tecnica para comprender las especificaciones.
- Un prototipo puede ser eliminado. (Puedo volver a construir un sistema mas robusto).
- Puede llegar a ser parte del producto final. (Cuando esta lo suficientemente maduro).

Que llegue a ser parte del producto final, suele ser malo, ya que de inicio se lo habia pensado como una muestra rapida, o vista previa para el cliente. Por lo que utilizarlo como software final no es aconsejable.

Etapas de prototipo

- Identificacion de requerimientos.
- Diseño rapido.
- Utilizar el prototipo.
- Revisar y mejorar.

Estos pasos conforman un bucle, por lo que se repetiran sin fin, hasta que el sistema se termine.

Puntos a favor

- Utiles cuando los requerimientos son cambiantes.
- No se conoce bien la aplicacion.
- Cuando el usuario no se quiere comprometer con los requerimientos.
- Cuando se quiere probar una arquitectura o tecnologia.
- Cuando se requiere rapidez en el desarrollo.

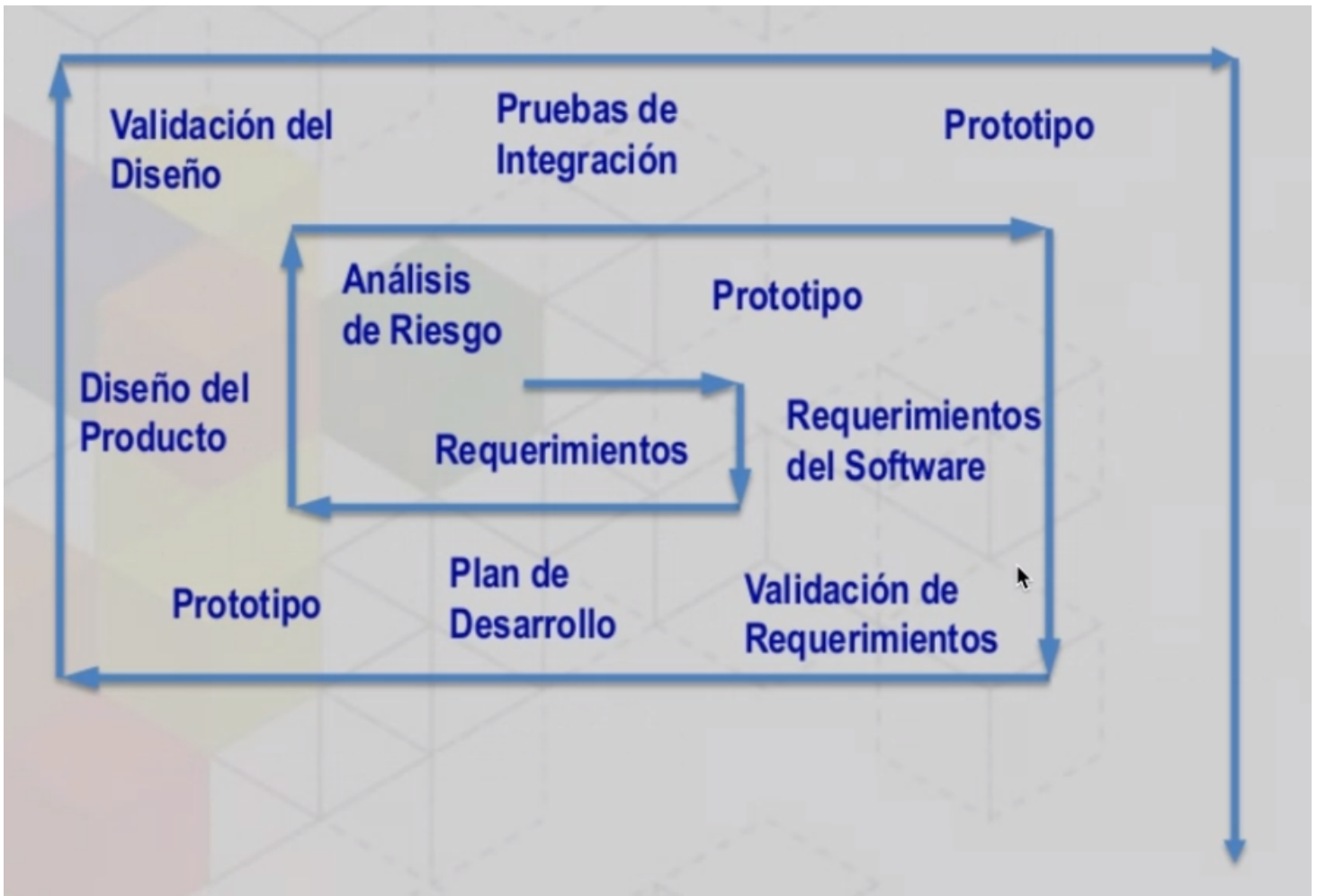
En contra

- No se conoce cuando se tendra un producto aceptable.
- No se sabe cuantas iteraciones seran necesarias.
- Da una falsa ilusion al usuario sobre la velocidad de desarrollo.
- Se puede volver el producto aun y cuando no este con los estandares.

Modelo en espiral

Esta orientado a los riesgos. Son creado a traves de multiples repeticiones del proceso del ciclo de vida. Basicamente se divide el proyecto en mini proyectos.

Hay que pensar que riesgos va a tener este proyecto.



Puntos a favor

- Avanza a pasos firmes solucionando riesgos en cada iteracion.
- Termina con todos los riesgos resueltos.
- Se pueden incluir otros metodos de desarrollo en las iteraciones.
- A medida que el costo aumenta, los riegos se reducen.
- Se tienen puntos de control en cada interaccion.

En contra

- Es complicado.
- Requiere de mucha administracion.
- Dificil de definir los objetivos, o las metas que indiquen que podemos avanzar al siguiente ciclo.

- Se puede caer en un desarrollo de nunca acabar. (O sea podemos avanzar ilimitadamente resolviendo riesgos y nunca acabar).

El modelo de procesos

- Impulsa un proceso iterativo de desarrollo.
- Cada ciclo es una versión del producto.
- Utiliza metas definidas para marcar la transición entre las distintas etapas.
- Ofrece mayor poder de decisión a los usuarios.
- Busca mejorar la calidad y creatividad.





A favor

- Etapas claramente definidas con metas entregables y responsables.
- Bueno cuando tengo proyectos grandes, ya que los voy a dividir en proyectos mas chiquitos.

En contra

- Para proyectos chicos, resulta poco practico.
- Al considerar versiones, hace que se dejen de lado algunas decisiones. (?)

Desarrollo incremental

- Permite construir el proyecto en etapas incrementales en donde cada etapa agrega funcionalidad.
- Cada etapa consiste de requerimientos, diseño, codificacion, pruebas y entrega.
- Permite entregar al cliente un producto mas rapido en comparacion del modelo en cascada.

La diferencia es que el modelo en espiral esta basado en los **riesgos**. Mientras que en este, lo primero que identifico son las **funcionalidades** y no los riesgos.



Puntos a favor

- Se pueden hacer implementaciones parciales si se cuenta con la suficiente funcionalidad.
 - Las pruebas y la integración es constante.
 - El progreso se puede medir en periodos cortos de tiempo.
- Resulta mas sencillo acomodar cambios.

Ademas **reduce los riesgos**, ya que:

- Provee visibilidad sobre el progreso a traves de sus nuevas versiones.
- Provee retroalimentación a traves de la funcionalidad mostrada.
- Permite atacar las mayores funcionalidades desde el inicio, lo que implica tambien atacar los mayores riesgos.

Puntos en contra

- Requiere de mucha planeación, tanto administrativa como técnica.
- Requiere de metas claras.