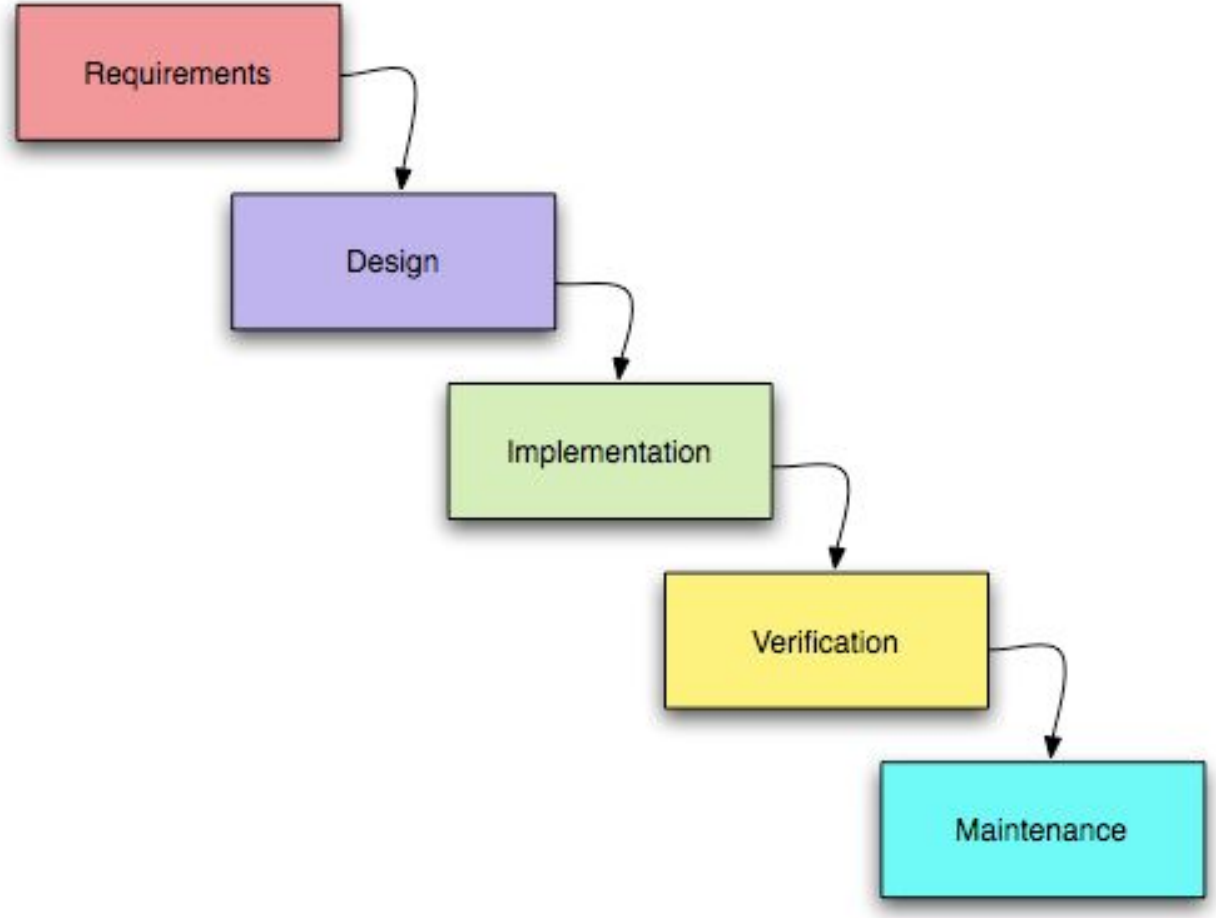
Análisis y Metodología de Sistemas

Unidad 1 - Temario

- Ciclos de Vida de un Sistema
- Modelo Secuencial
- Modelo en Cascada
- Modelo Iterativo
- Modelo Incremental
- Modelo Espiral
- Modelo de Prototipos

Ciclo de Vida de un Sistema



Ciclo de Vida
Desarrollo de Software (SDLC)

- El ciclo de vida del desarrollo Software (Systems Development Life Cycle) , es una secuencia estructurada y bien definida de las etapas en Ingeniería de software para desarrollar el producto software deseado.
- Es un estilo consensuado, reúne, preserva y re-estructura elementos (aún vigentes) de estilos más antiguos.
- Hay varios modelos a seguir para el establecimiento de un proceso para el desarrollo de software, cada uno de los cuales describe un enfoque diferente para diferentes actividades que tienen lugar durante el proceso.
- Suele ser, y es cuestionado por eso, algo burocrático pero presenta buena trazabilidad y buen seguimiento para un proyectos de software.

Metodología de Desarrollo de Software

Son los procesos que hay que seguir sistemáticamente para idear, implementar y mantener un producto de software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cuál fue creado.

La **Norma 12207 de ISO** lo define como: " Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto sw, abarcando desde la definición hasta la finalización de su uso "

La IEEE 1074-2006 lo define como: " Una aproximación lógica a la adquisición, el suministro, el desarrollo, la explotación y el mantenimiento del software "

(Institute of Electrical and Electronics Engineers)

La Metodología

- La metodología es un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas probabilidades de éxito.
- Define estados, etapas o fases de un desarrollo, junto con los criterios de transición entre ellos.
- Tareas, actividades, etc.
- Roles, con sus skills necesarios y las interacciones entre ellos.
- Artefactos o entregables.
- Herramientas de control, seguimiento, medición y perfeccionamiento.
- Principios, criterios para tomar decisiones, estrategias para manejar distintos tipos de situaciones, herramientas de manejo de riesgos, etc.

Ventajas de los procesos de desarrollo

- Todos los integrantes del equipo del proyecto trabajan bajo un marco común.
- Estandarización de conceptos, actividades y nomenclatura.
- Actividades de desarrollo apoyadas por procedimientos y guías.
- Resultados de desarrollo predecibles.
- Uso de herramientas de ingeniería de software.
- Planificación de las actividades en base a un conjunto de tareas definidas y a la experiencia en otros proyectos.
- Recopilación de mejores prácticas para proyectos futuros.

Mejoras de los productos

- Se asegura que los productos cumplen con los objetivos de calidad propuestos.
- Detención temprana de errores.
- Se garantiza la trazabilidad de los productos a lo largo del proceso de desarrollo.

Mejoras en las relaciones cliente/usuario

- El cliente percibe el orden en los procesos.
- Facilita al cliente el seguimiento de evolución del proyecto.
- Se establecen mecanismos para asegurar que los productos desarrollados cumplan con las expectativas del cliente

Ciclo de Vida

Metodologías:
Prescriptivas o Tradicionales

- Mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado.

- Imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el fin de conseguir un software más eficiente.

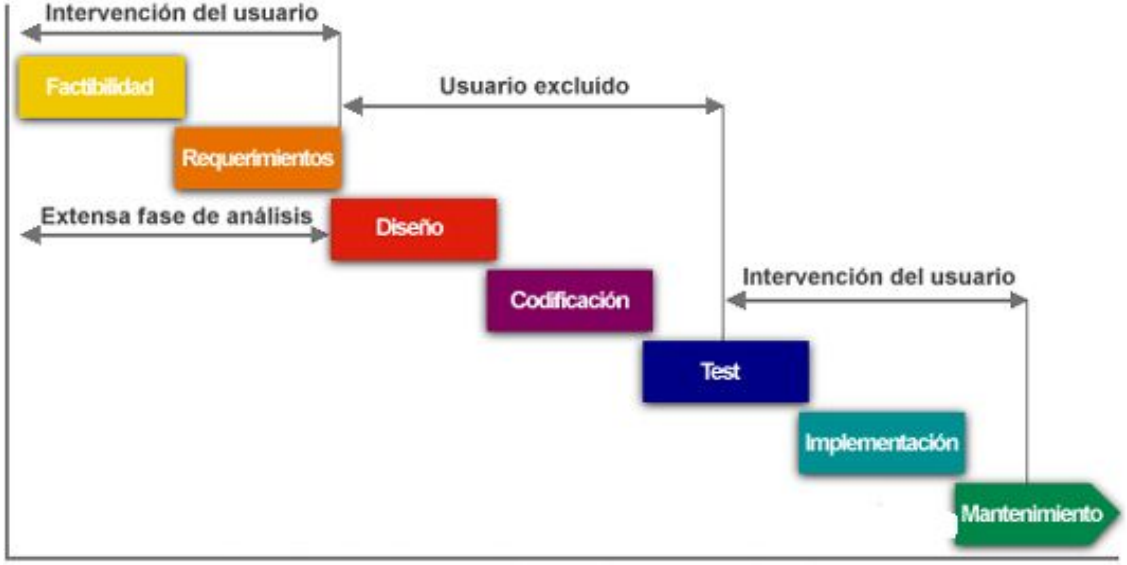
- Se centran especialmente en el control del proceso, mediante una rigurosa definición de roles, actividades, artefactos, herramientas y notaciones para el modelado y documentación detallada.

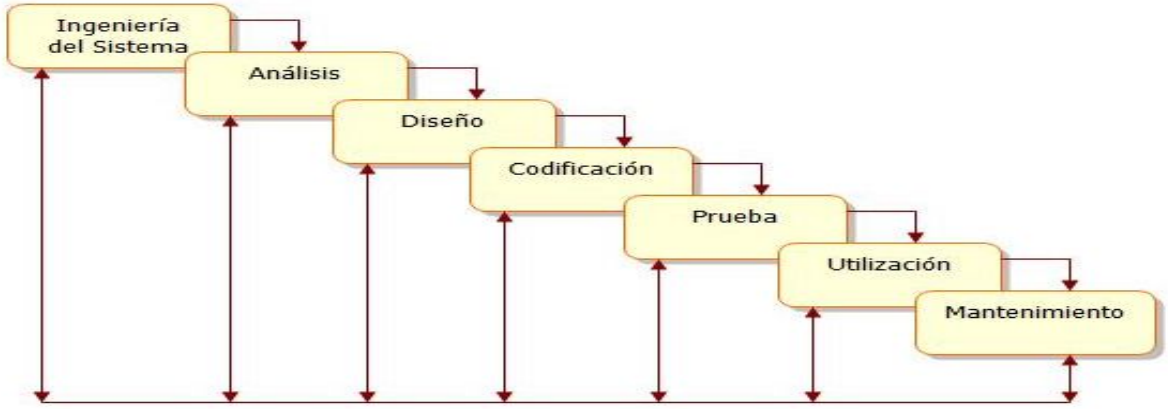
Modelo en Cascada

- Secuencia Ordenada
- Cada fase empieza cuando ha terminado la anterior
- Para pasar de una fase a otra es necesario conseguir todos los objetivos de la fase anterior
- Ayuda a prevenir que se sobrepasen la fecha de entrega y los costos esperados
- Al final de cada fase técnicos y usuarios tienen la oportunidad de revisar el proceso del proyecto.

Modelo de Cascada

- Se define como una secuencia de fases en la que al final de cada una de ellas se reúne la documentación para garantizar que cumple las especificaciones y los requisitos antes de pasar a la fase siguiente
- Admite iteraciones
- Después de cada etapa se realiza una o varias revisiones para comprobar que se puede pasar a la siguiente etapa





Modelo de Cascada: Ventajas ~ Desventajas

Ventajas

- Admite iteraciones (Se permite volver a una etapa anterior del proyecto).
- Planificación sencilla.
- Provee un producto con un elevado grado de calidad sin disponer de un personal altamente calificado.
- Adecuado si se disponen de todos los requerimientos desde el principio.

Usos

- Cuando se disponen de todos los requerimientos desde el principio (x ej: en reingeniería).
- Producto no novedoso o con funcionalidades conocidas.
- Proyectos complejos fácilmente entendibles.

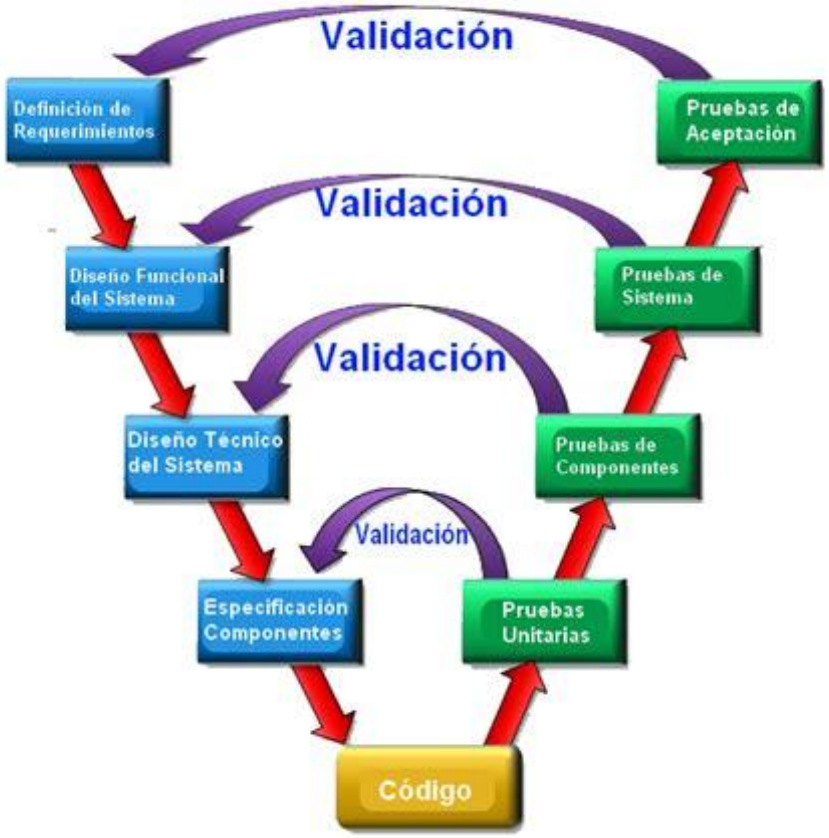
Desventajas

- Es rígido, poco flexible y con muchas restricciones.
- La necesidad de conocer todos los requerimientos al comienzo del proyecto.
- Si se han cometido errores y no se detectan en la etapa inmediatamente siguiente, es costoso y difícil volver atrás para realizar la corrección.
- Los resultados no se ven hasta en las etapas finales del ciclo.
- Cualquier error detectado nos trae un retraso y aumenta el costo del desarrollo.
- Retardo en entregar partes del producto.

Modelo V

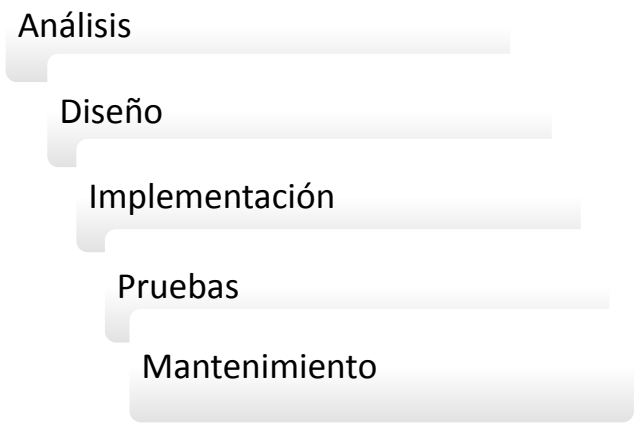
- El mayor inconveniente del modelo de cascada es que solo se pasa a la siguiente fase cuando se completa la anterior, por tanto no es posible volver atrás si se encuentra algún error en las etapas posteriores.

- El Modelo V aporta opciones de evaluación del software en cada etapa de manera inversa, de esta manera se puede monitorear el avance y acompañarlo con validaciones permanentes.



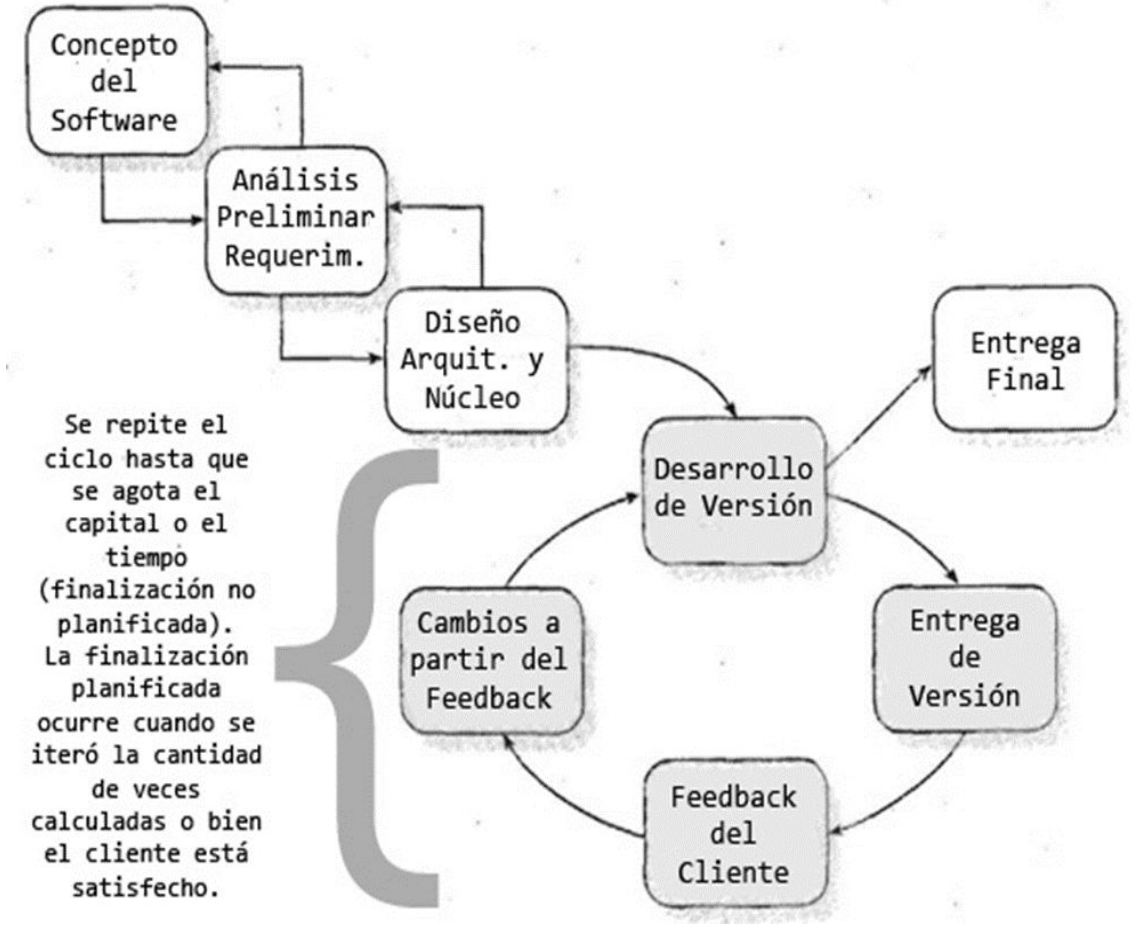
Modelo Sashimi

- Se permite un solapamiento entre fases.
  Ejemplo: sin tener terminado el diseño se comienza a implementar.
- Ventajas:
  - No necesita generar tanta documentación cómo el ciclo de cascada pura debido a la continuidad del mismo personal entre fases.
- Desventajas:
  - Difícil de controlar el progreso del proyecto, dado que los finales de fase ya no son un punto de referencia.
  - Si hay problemas de comunicación pueden surgir inconsistencias.

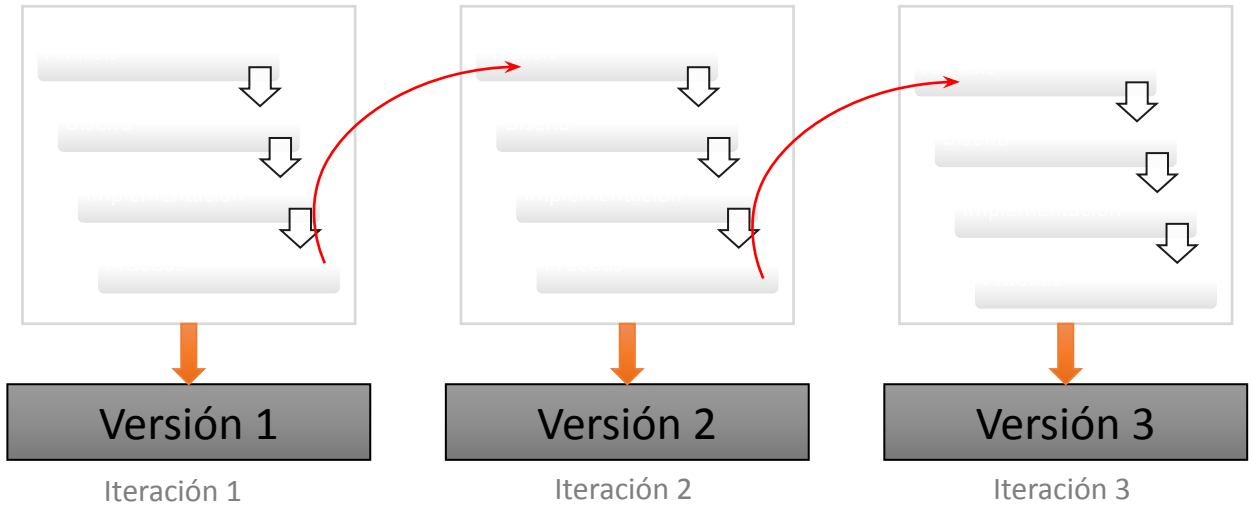


Modelo Evolutivo

- Este modelo acepta que los requerimientos del usuario puedan cambiar en cualquier momento.
- El problemas de los nuevos requerimientos se afronta mediante una iteración de ciclos de requerimientos - desarrollo - evaluación.
- Puede ser muy útil cuando se desconocen la mayoría de los requerimientos iniciales, o estos requerimientos no están completos.



Modelo Iterativo



- Busca reducir el riesgo que surge entre las necesidades del usuario y el producto final por malos entendidos durante la etapa de solicitud de requerimientos.
- Es la iteración de varios ciclos en cascada.
- Al final de cada iteración, se le entrega al cliente una versión mejorada o con mayores funcionalidades del producto.
- El cliente es quien, después de cada iteración, evalúa el producto y lo corrige o propone mejoras.
- Las iteraciones se repetirán hasta que el cliente quede satisfecho.

Modelo Iterativo - Usos

- Se suele utilizar en proyectos en los que los requerimientos no están claros de parte del usuario. Por lo que se hace necesaria la creación de distintos prototipos para presentarlos.

- En aplicaciones medianas a grandes, en las que el cliente no necesita todas las funcionalidades desde el principio del proyecto.

- Por ejemplo, una empresa que quiera migrar sus aplicaciones a otra arquitectura, y desea hacerlo paulatinamente.

Modelo por Prototipos

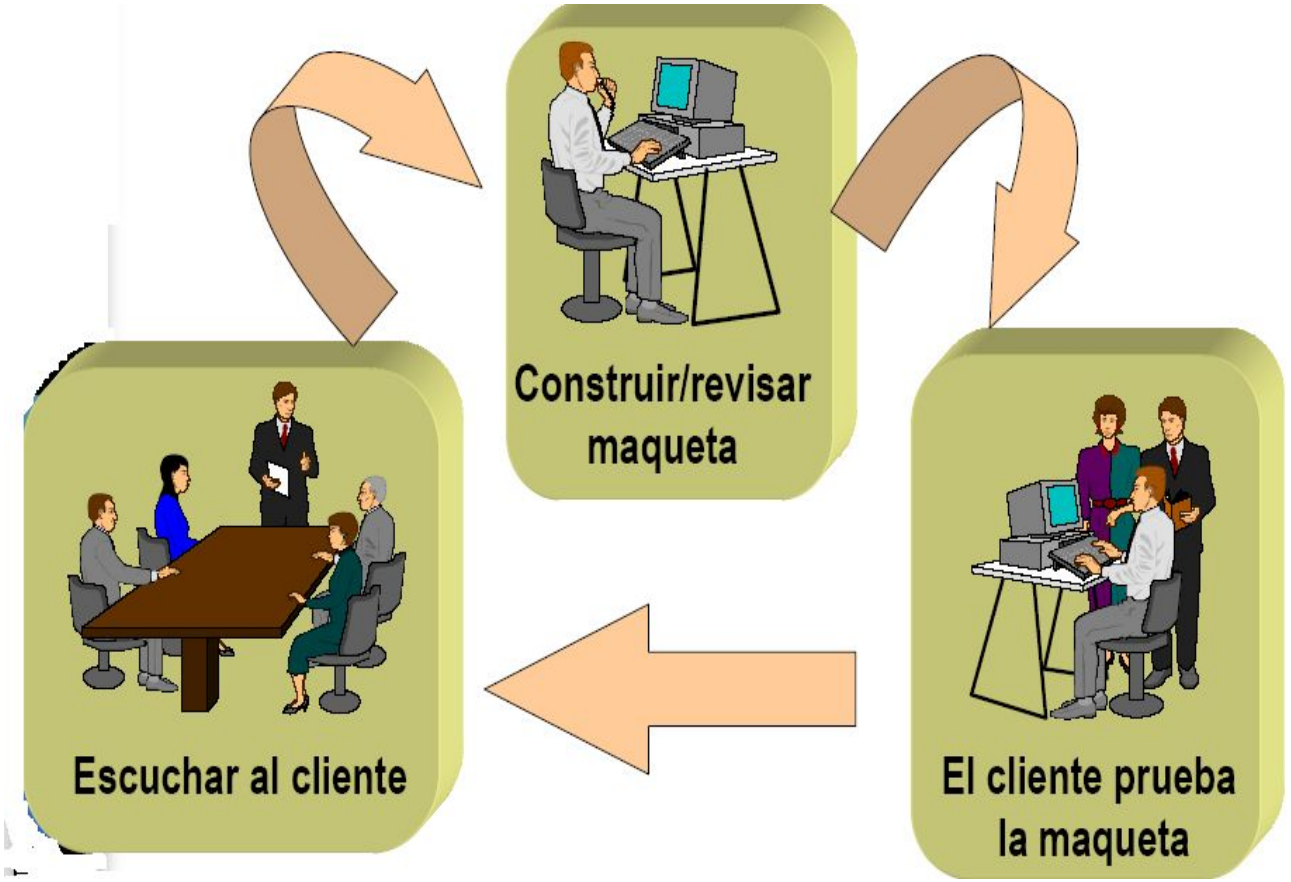
- Usado cuando no se conoce exactamente cómo desarrollar un determinado producto o cuáles son las especificaciones de forma precisa.

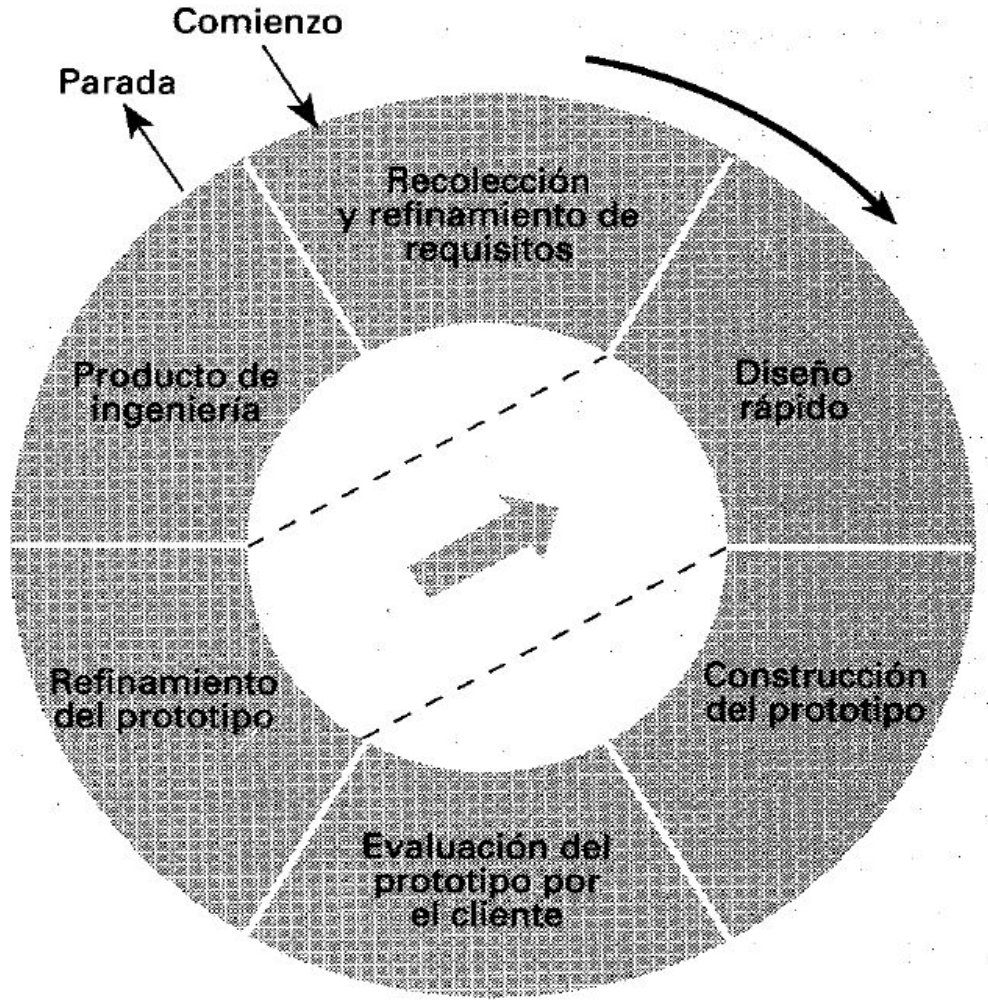
- En estos casos suele recurrirse a definir especificaciones iniciales para hacer un prototipo, o sea, un producto parcial y provisional.

- El objetivo es lograr crear un producto intermedio, antes de realizar el producto final, para conocer mediante el prototipo cómo responderán las funcionalidades previstas para el producto final.

- Antes de adoptar este tipo de ciclo de vida, deberíamos analizar si el esfuerzo por crear un prototipo, realmente vale la pena.

Modelo por Prototipos





Modelo por Prototipos - Usos

- Utilizado mayoritariamente en desarrollo de productos con innovaciones importantes, o en el uso de tecnologías nuevas o poco probadas.
- Es el único apto para desarrollos en los que no se conoce a priori sus especificaciones. Cómo contrapartida, tiene la desventaja de poder ser altamente costoso.

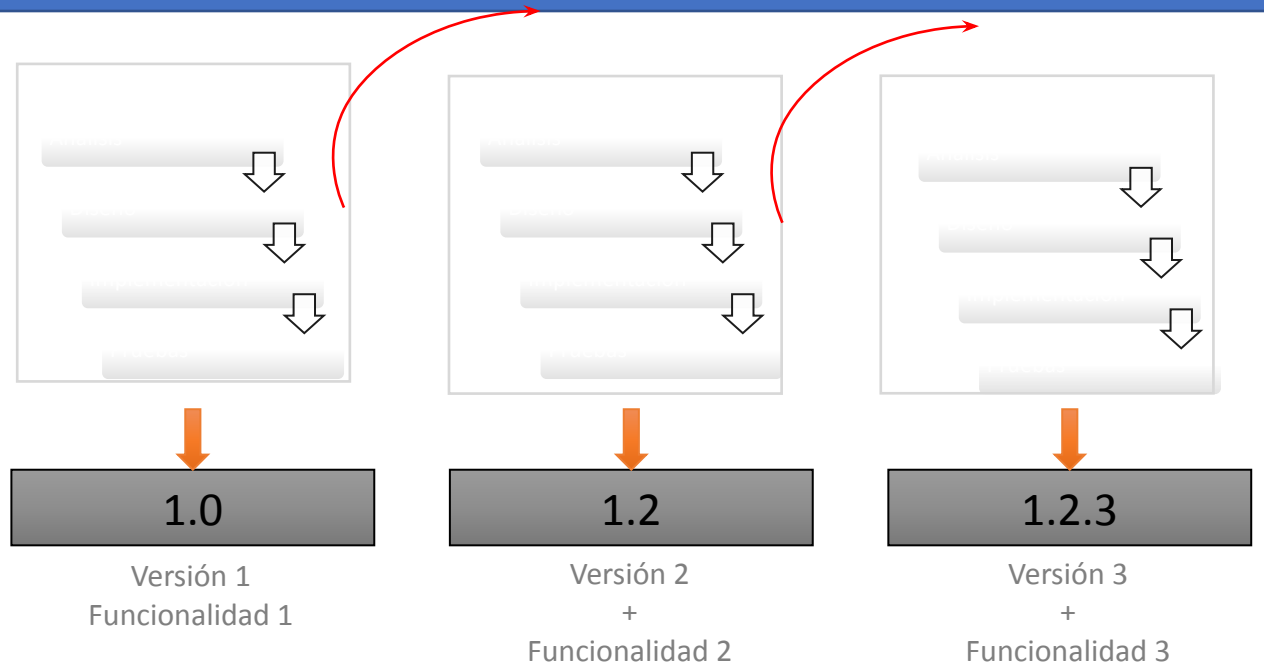
El Prototipado para que sea EFECTIVO

- Debe ser un sistema con el que se pueda experimentar
- Debe ser comparativamente barato (< 10%)
- Debe desarrollarse rápidamente
- Énfasis en la interfaz de usuario
- Equipo de desarrollo reducido
- Herramientas y lenguajes adecuados

Modelo Incremental

- Se basa en la filosofía de construir incrementando las funcionalidades del programa.
- Se realiza construyendo por módulos que cumplen las diferentes funciones del sistema. Esto permite aumentar gradualmente las capacidades del software.
- Facilita el desarrollo, permitiendo a cada miembro del equipo desarrollar un módulo particular (en caso de que sea realizado por un equipo de programadores).
- Similar al ciclo de vida en cascada con iteraciones, aplicándose un ciclo en cada nueva funcionalidad del programa.
- Al final de cada ciclo, se le entrega al cliente la versión que contiene la nueva funcionalidad.
- Nos permite hacer una entrega al cliente antes de acabar el proyecto.

Modelo Incremental
Esquema y Beneficios



Beneficios:

- Construir un sistema pequeño implica menos riesgos que construir uno grande.
- Si se detecta un error grave, sólo desechamos la última iteración.
- No se necesitan todos los requerimientos al principio del proyecto.
- Facilita la aplicación de la filosofía divide & conquistaras.

Modelo en Espiral

- Se basa en una serie de ciclos repetitivos para ir ganando madurez en el producto final.
- Toma los beneficios de los ciclos de vida incremental y por prototipos, pero se tiene más en cuenta el concepto de riesgo que aparece debido a las incertidumbres e ignorancias.
- A medida que el ciclo se cumple (el avance de la espiral), se van obteniendo software que va ganando la satisfacción del cliente.
- A menudo, la fuente de incertidumbres es el propio cliente, que en la mayoría de los casos no sabe con exactitud todas las funcionalidades que debe tener el producto.

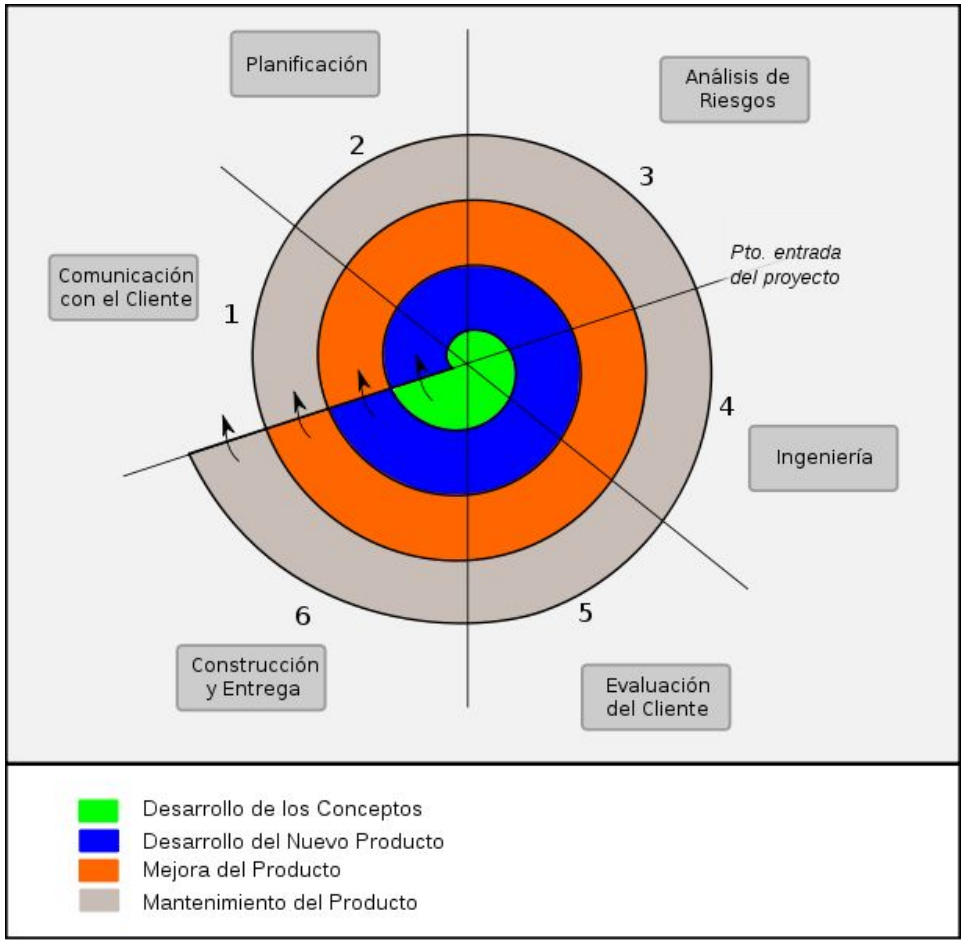
Modelo en Espiral - Etapas

Hay cuatro actividades que envuelven las etapas:

- **Planificación** → Relevamiento de requerimientos iniciales o luego de una iteración.
- **Análisis de riesgos** → De acuerdo con el relevamiento de requerimientos, se decide si se continúa con el desarrollo.
- **Implementación** → Desarrollo de un prototipo basado en los requerimientos.
- **Evaluación** → El cliente evalúa el prototipo, si da su conformidad, termina el proyecto. En caso contrario, incluimos los nuevos requerimientos solicitados por el cliente en la siguiente iteración.

Modelo Espiral - Esquema

Se comienza desde el centro



Modelo de Espiral: Ventajas ~ Desventajas

Ventajas

- Puede comenzarse un proyecto con un alto grado de incertidumbre.
- Bajo riesgo de retraso en caso de detección de errores, ya que se puede solucionar en la próxima rama de la espiral.

Desventajas

- El costo temporal que suma cada vuelta de la espiral.
- La dificultad para evaluar los riesgos.
- Necesidad de la presencia o la comunicación continua con el cliente o usuario.