



# Ingeniería de Software I

Modelos de Proceso

# Modelos de Proceso

---

# Proceso

---

- ❑ Cuando proveemos un servicio o creamos un producto, siempre seguimos una secuencia de pasos para realizar un conjunto de tareas.
- ❑ Las tareas son realizadas usualmente en el mismo orden.
- ❑ Por ejemplo, no se puede cocinar una torta antes de que todos los ingredientes sean mezclados.
- ❑ Se puede pensar que un “conjunto ordenado de tareas” como un proceso.



# ¿Qué es un proceso de software?

- ❑ Es un conjunto de actividades y resultados asociados que producen un producto de software.



# ¿Qué es un proceso de software?

---

Actividades fundamentales de los procesos:

- ☐ Especificación del software
- ☐ Desarrollo del software
- ☐ Validación del software
- ☐ Evolución del software

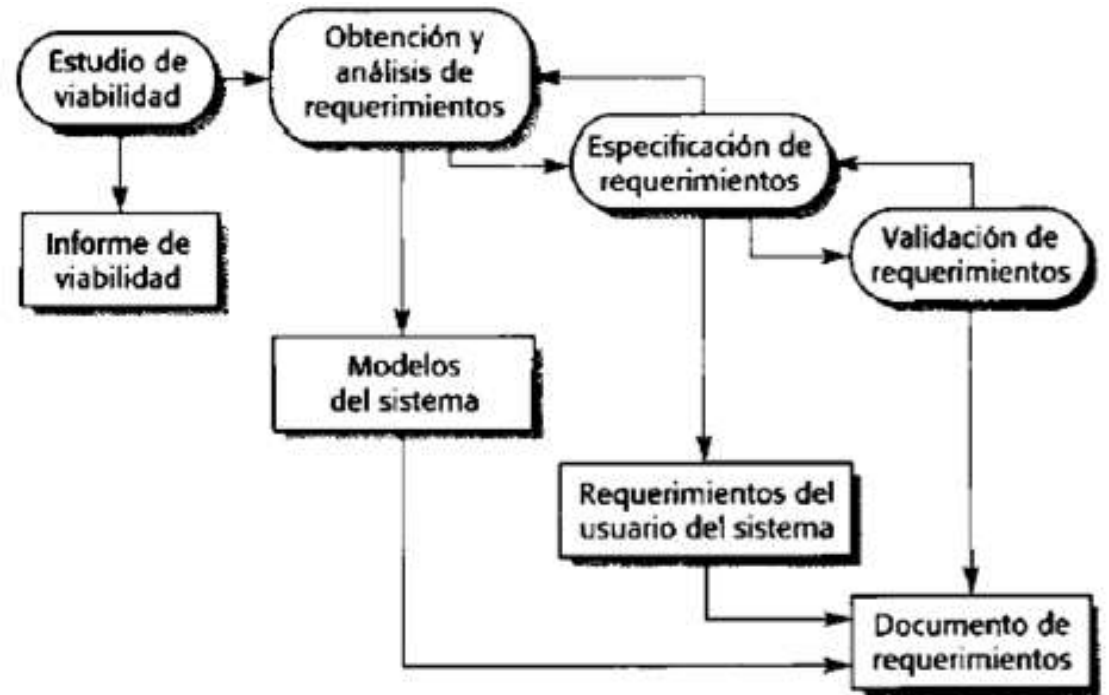


# Actividades fundamentales de los procesos

## ❑ Especificación del software

Consiste en el proceso de comprender y definir el sistema, así como la identificación de las restricciones para el desarrollo del sistema.

También llamada, *Ingeniería de Requerimientos*





# Actividades fundamentales de los procesos

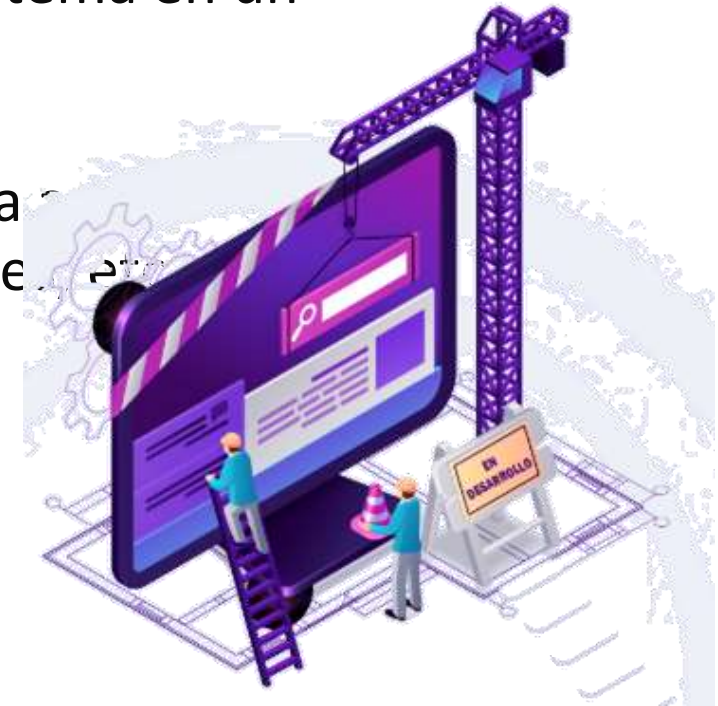
---

## ❑ Desarrollo del software

Corresponde al proceso de convertir una especificación del sistema en un sistema ejecutable.

Incluye los procesos de diseño y programación.

Se crea una descripción de la estructura del software que se va a implementar, los modelos y estructuras de datos, las interfaces.



# Actividades fundamentales de los procesos

---

## ❑ Validación del software

Se realiza para mostrar que un sistema cumple tanto con sus especificaciones como con las expectativas del cliente.

La prueba del sistema con datos de prueba simulados, es una de las formas de validación.

Pero tambien incluye inspecciones y revisiones en distintas





# Actividades fundamentales de los procesos

---

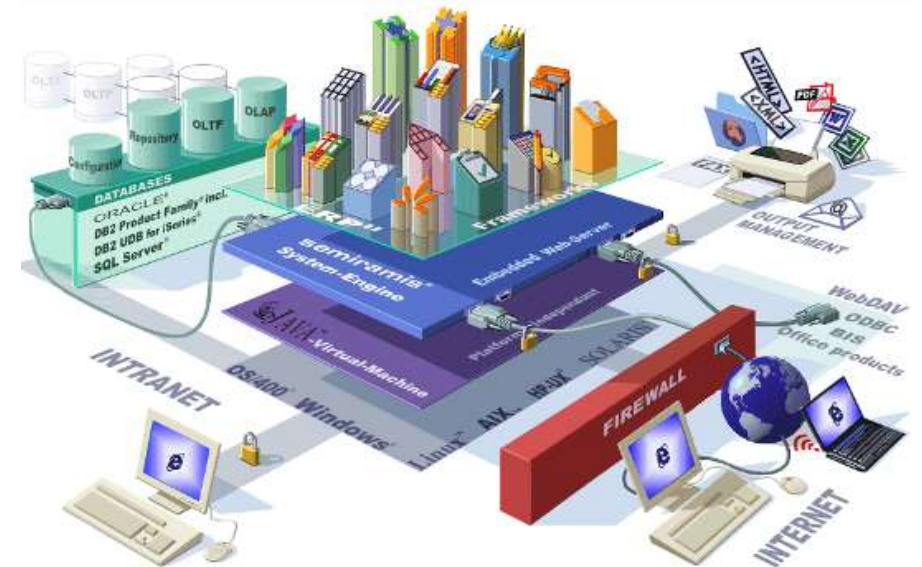
## ❑ Evolución del software

El mantenimiento es una actividad a tener en cuenta en el proceso de desarrollo de software. Eso implica también cambios y mejoras



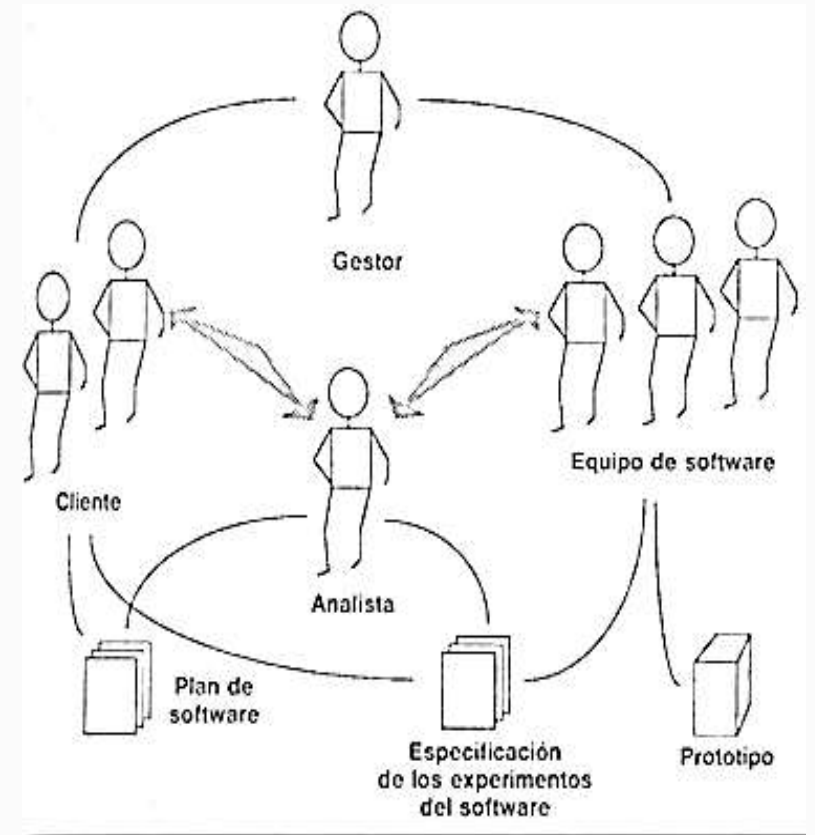
# ¿Qué es un modelo de proceso de software?

- ❑ Es una representación simplificada de un proceso de software que presenta una visión de ese proceso.
- ❑ Estos modelos pueden incluir actividades que son partes de los procesos y productos de software, y el papel de las personas involucradas.



# ¿Qué es un modelo de proceso de software?

Marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso (*norma ISO 12207-1*) [ISO/IEC, 1995]



# Modelos de Proceso

---

## Características

- ☐ Establece todas las actividades.
- ☐ Utiliza recursos, está sujeto a restricciones y genera productos intermedios y finales.
- ☐ Puede estar compuesto por subprocessos.
- ☐ Cada actividad tiene entradas y salidas definidas.
- ☐ Las actividades se organizan en una secuencia.
- ☐ Existen principios que orientan sobre las metas de cada actividad.
- ☐ Las restricciones pueden aplicarse a una actividad, recurso o producto.

# Modelos de Proceso

---

## Ciclo de vida

Proceso que implica la construcción de un producto

## Ciclo de vida del Software

Describe la vida del producto de software desde su concepción hasta su implementación, entrega, utilización y mantenimiento

## Modelos de proceso de software

Es una representación abstracta de un proceso del software.

Modelo de proceso

Paradigma de software

Ciclo de vida del software



Términos Equivalentes

# Modelos de Proceso

---

## Modelos **prescriptivos**

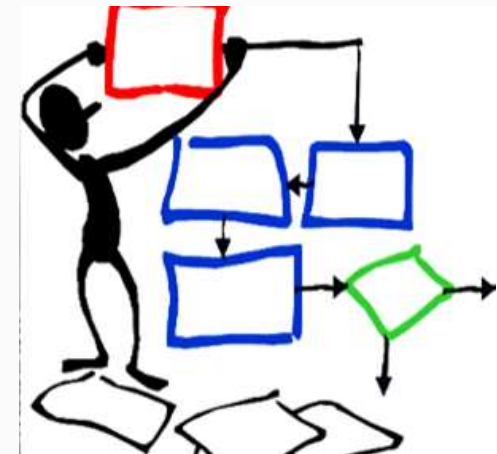
Prescriben un conjunto de elementos del proceso: actividades del marco de trabajo, acciones de la ingeniería del software, tareas, aseguramiento de la calidad y mecanismos de control.

Cada modelo de proceso prescribe también un “flujo de trabajo”, es decir de qué forma los elementos del proceso se interrelacionan entre sí.

## Modelos **descriptivos**

Descripción en la forma en que se realizan en la realidad

Ambos modelos deberían ser iguales





# Modelos de Proceso

---

## Modelos tradicionales

Formados por un conjunto de fases o actividades en las que e no tienen en cuenta la naturaleza evolutiva del software

- ☐ Clásico, lineal o en cascada
- ☐ Modelo en V
- ☐ Basado en prototipos



## Modelos evolutivos

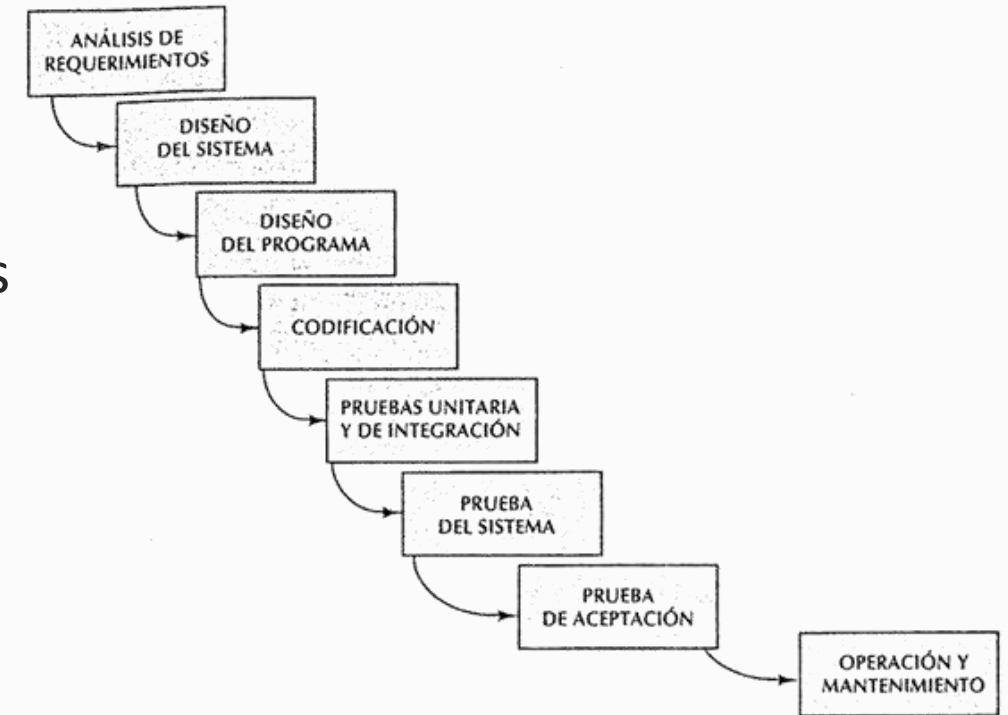
Son modelos que se adaptan a la evolución que sufren los requisitos del sistema en función del tiempo

- ☐ En espiral
  - ☐ Evolutivo
  - ☐ Incremental
- Desarrollo por fases

## Procesos ágiles

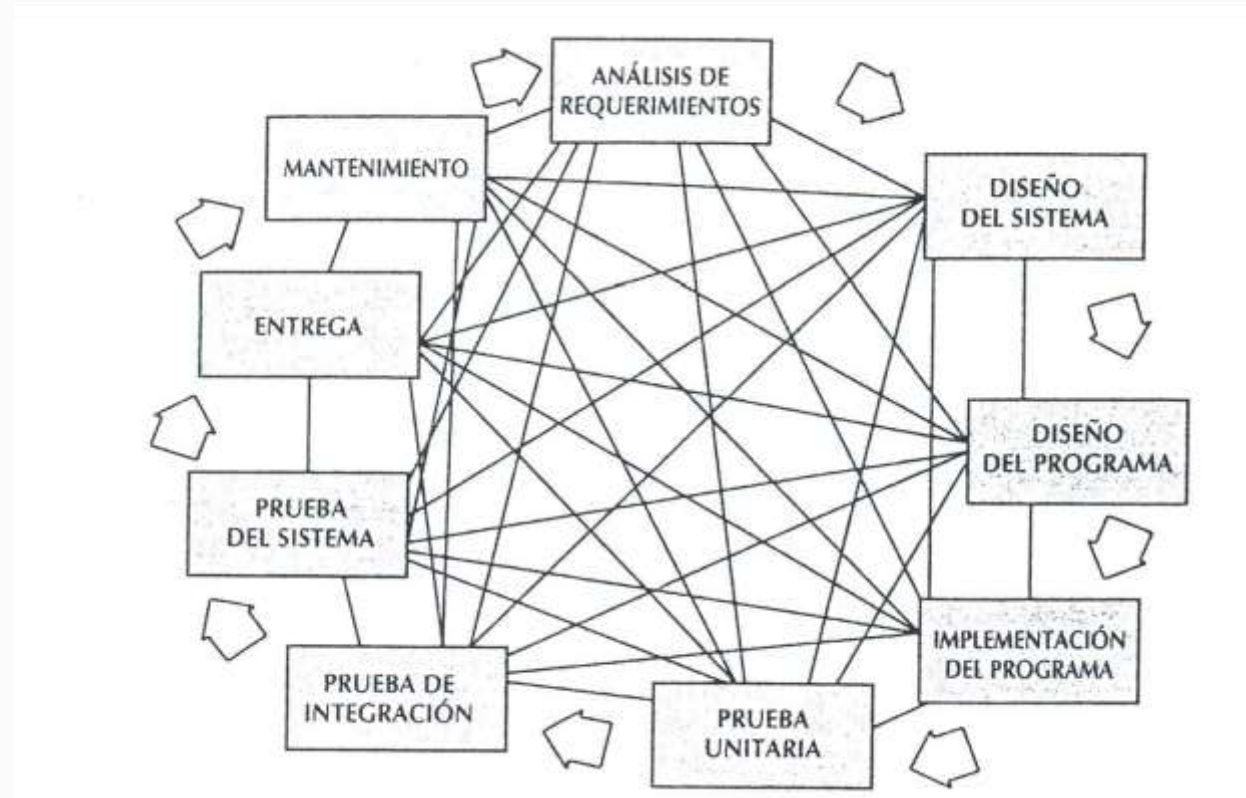
# Modelo en cascada

- ❑ Las etapas se representan cayendo en cascada
- ❑ Cada etapa de desarrollo se debe completar antes que comience la siguiente
- ❑ Útil para diagramar lo que se necesita hacer
- ❑ Su simplicidad hace que sea fácil explicarlo a los clientes



# Modelo de la realidad en comparación con cascada

- ❑ Modelo de la realidad (sin control entre las etapas)



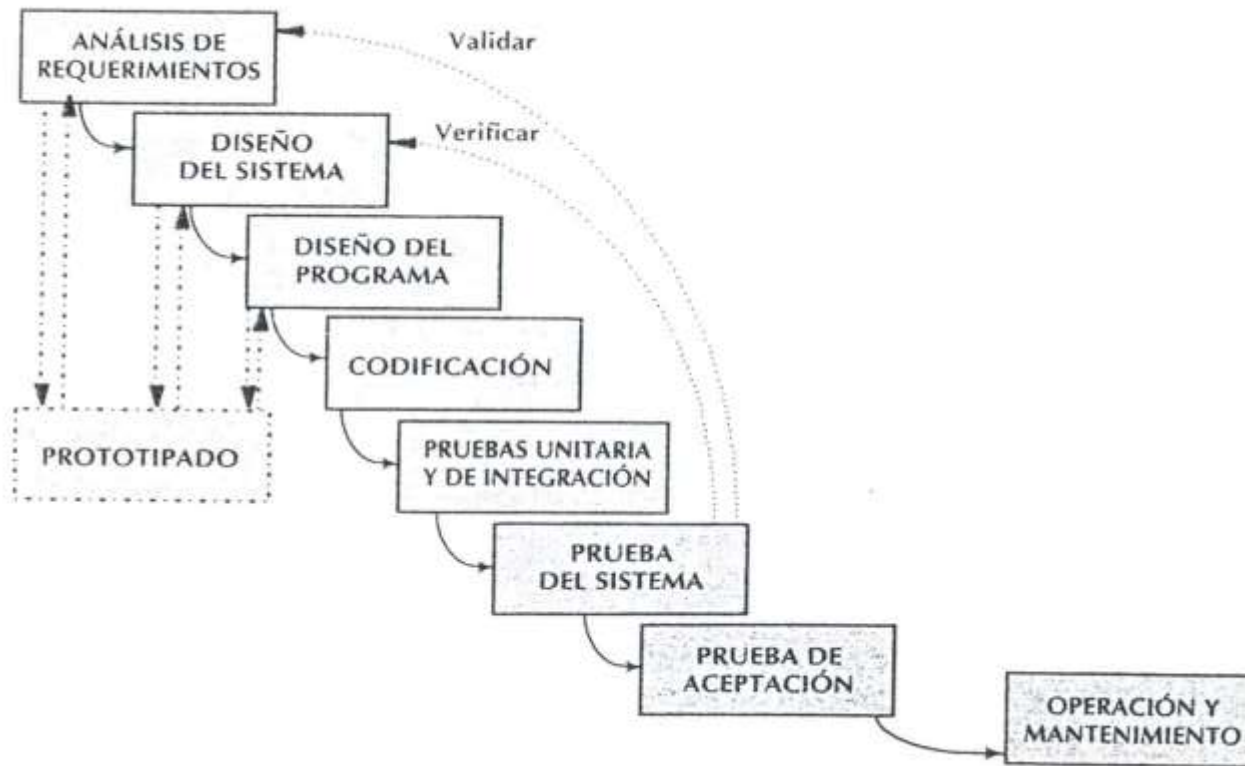
# Modelo en cascada

---

## Dificultades:

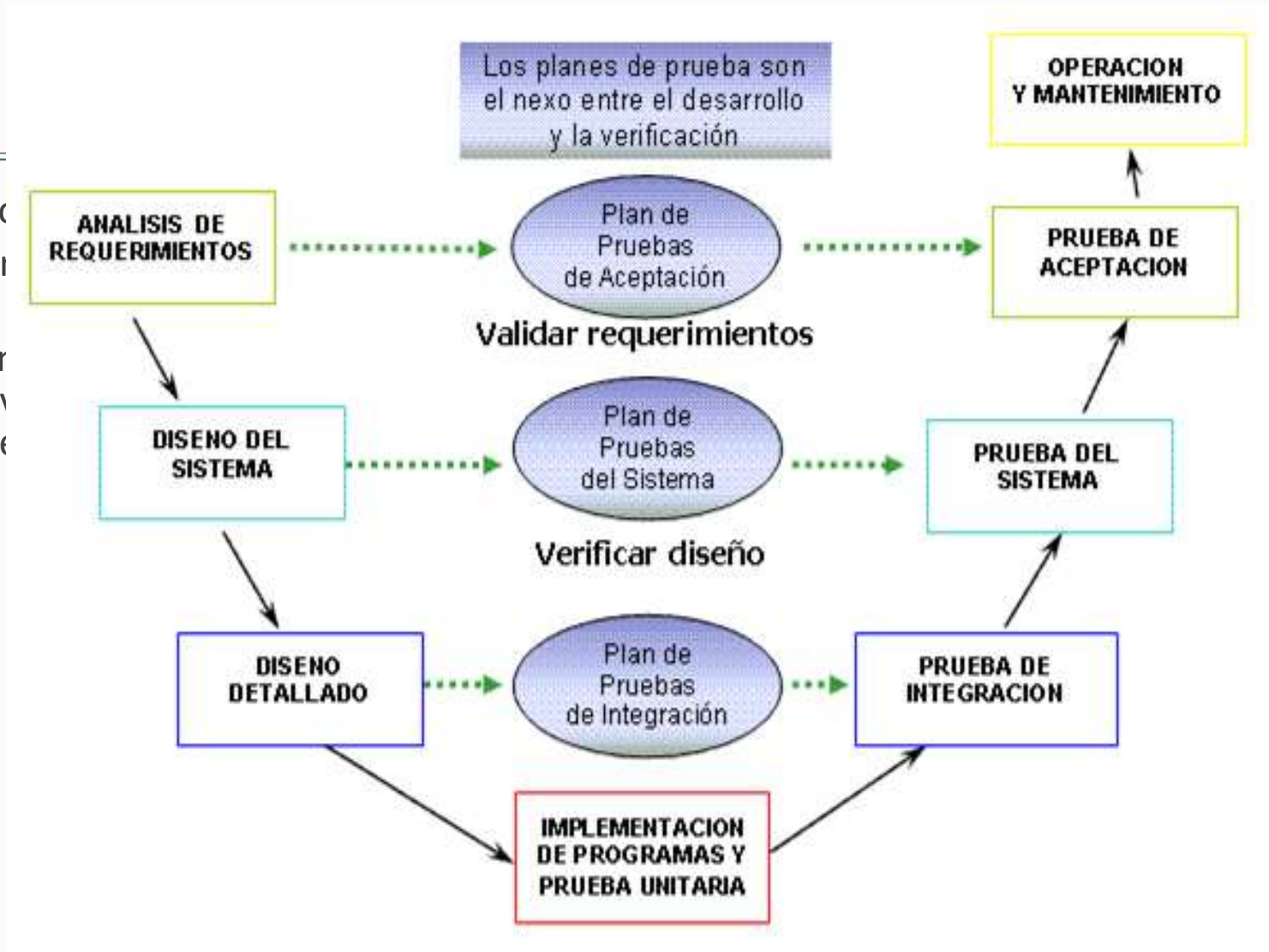
- ❑ No existen resultados concretos hasta que todo este terminado.
- ❑ Las fallas más triviales se encuentran al comienzo del período de prueba y las más graves al final.
- ❑ La eliminación de fallas suele ser extremadamente difícil durante las últimas etapas de prueba del sistema.
- ❑ Deriva del mundo del hardware y presenta una visión de manufactura sobre el desarrollo de software.
- ❑ La necesidad de pruebas aumenta exponencialmente durante las etapas finales.
- ❑ "CONGELAR" una fase es poco realista.
- ❑ Existen errores, cambios de parecer, cambios en el ambiente.

# Modelo en cascada con prototipo



# Modelo

- ❑ Demuestra cómo
- ❑ Sugiere que la pi programa
- ❑ La vinculación en la verificación y para solucionar e



del  
rante  
te



# Modelo de prototipos

---

Un prototipo es un producto parcialmente desarrollado que permite que clientes y desarrolladores examinen algunos aspectos del sistema propuesto, y decidan si éste es adecuado o correcto para el producto terminado.

Esta es una alternativa de especificación para tratar mejor la incertidumbre, la ambigüedad y la volubilidad de los proyectos reales.



# Modelo de prototipos- Tipos

## Evolutivos

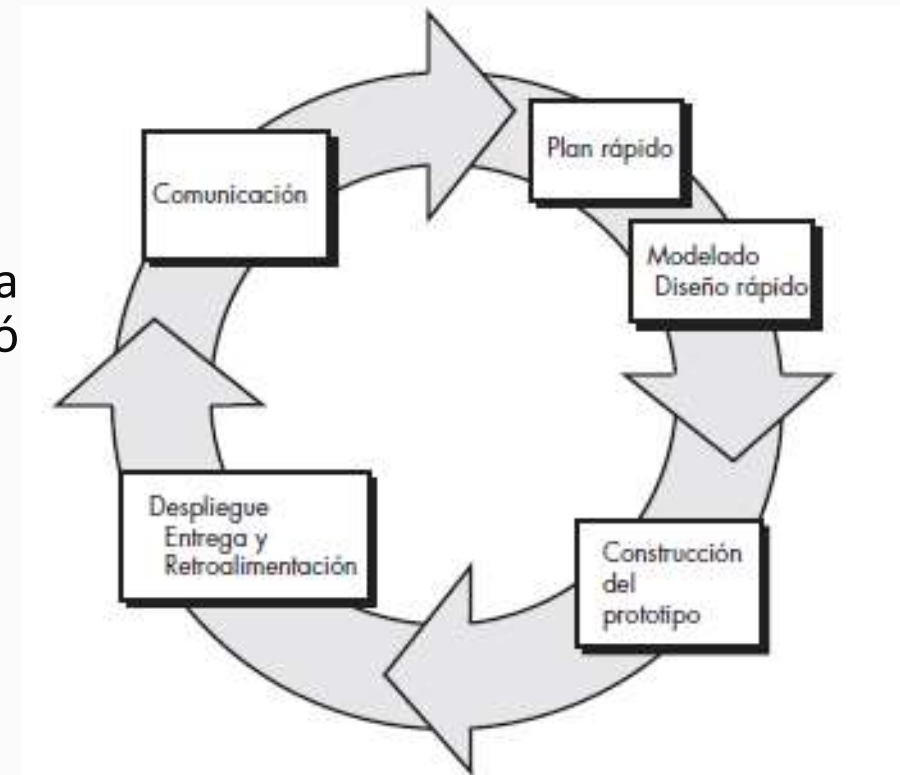
El objetivo es obtener el sistema a entregar.

Permite que todo el sistema o alguna de sus partes se construyan rápidamente para comprender o aclarar aspectos y asegurar que el desarrollador, el usuario y el cliente tengan una comprensión unificada tanto de lo que se necesita como de lo que se propone como solución.

## Descartables

No tiene funcionalidad

Se utilizan herramientas de modelado



# Modelo de prototipos

---

	Descartable	Evolutivo
Enfoque de desarrollo	Rápido y sin rigor	Riguroso
Que construir	Solo las partes problemáticas	Primero las partes bien entendidas. Sobre una base sólida
Objetivo ultimo	Desecharlo	Lograr el sistema

```

graph LR
    IR[Initial Requirements] --> Design[Design]
    Design --> Prototyping[Prototyping]
    Prototyping --> CE[Customer Evaluation]
    CE --> Dev[Development]
    Dev --> Test[Test]
    Test --> Maintain[Maintain]
    CE --> Design
    CE --> Prototyping
    Test --> Dev
    Review[Review & Updating]
  
```

The diagram illustrates the Software Engineering process as a sequence of stages with feedback loops. The stages are: Initial Requirements, Design, Prototyping, Customer Evaluation, Development, Test, and Maintain. The process starts with Initial Requirements, leading to Design, then Prototyping, Customer Evaluation, Development, Test, and finally Maintain. There are feedback loops from Customer Evaluation back to Design and Prototyping, and from Test back to Development. A 'Review & Updating' stage is shown below the main flow, connected to the Customer Evaluation and Test stages.

# Proyectos candidatos

- ❑ Usuarios que no examinarán los modelos abstractos
- ❑ Usuarios que no determinarán sus requerimientos inicialmente
- ❑ Sistemas con énfasis en los formatos de E/S más que en los detalles algorítmicos
- ❑ Sistemas en los que haya que explorar aspectos técnicos
- ❑ Si el usuario tiene dificultad al tratar con los modelos gráficos para modelar los requerimientos y el comportamiento
- ❑ Si se enfatiza el aspecto de la interfaz humana

# Modelo de prototipos

---

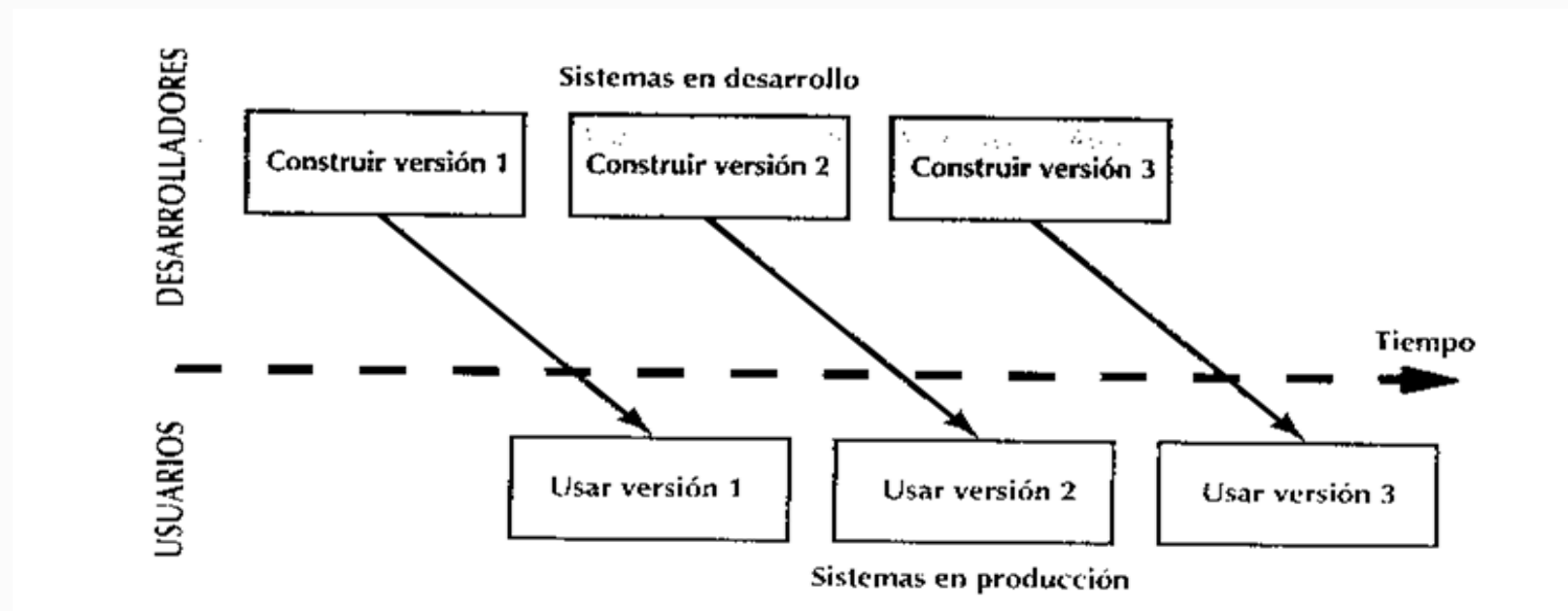
Para asegurar el éxito:

- ❑ 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 de desarrollo por fases

Se desarrolla el sistema de tal manera que puede ser entregado en piezas. Esto implica que existen dos sistemas funcionando en paralelo: el sistema operacional y el sistema en desarrollo.





# Modelo de desarrollo por fases

---

## Tipos de modelos de desarrollo por fases

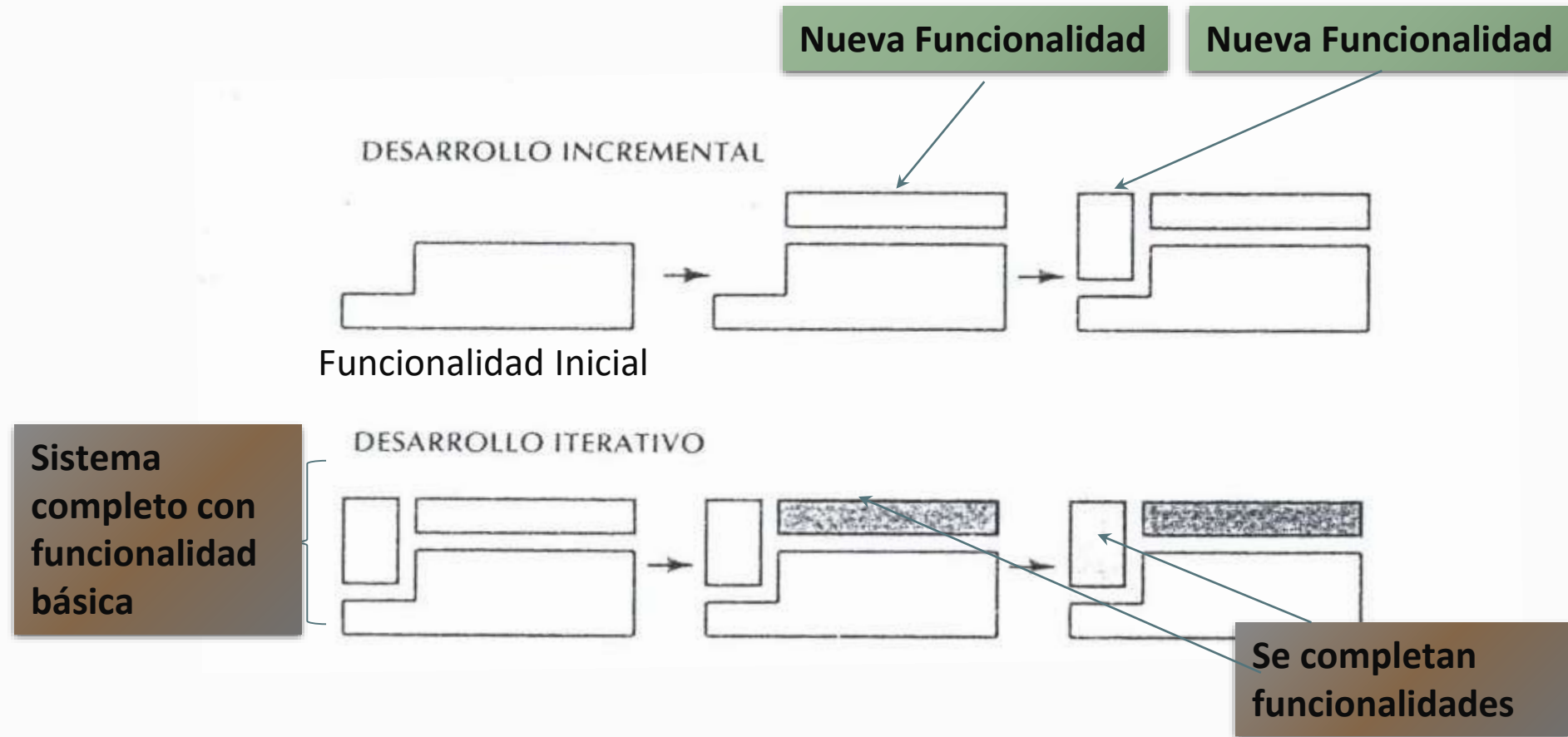
### **Incremental**

El sistema es particionado en subsistemas de acuerdo con su funcionalidad. Cada entrega agrega un subsistema.

### **Iterativo**

Entrega un sistema completo desde el principio y luego aumenta la funcionalidad de cada subsistema con las nuevas versiones.

# Modelos de desarrollo por fases



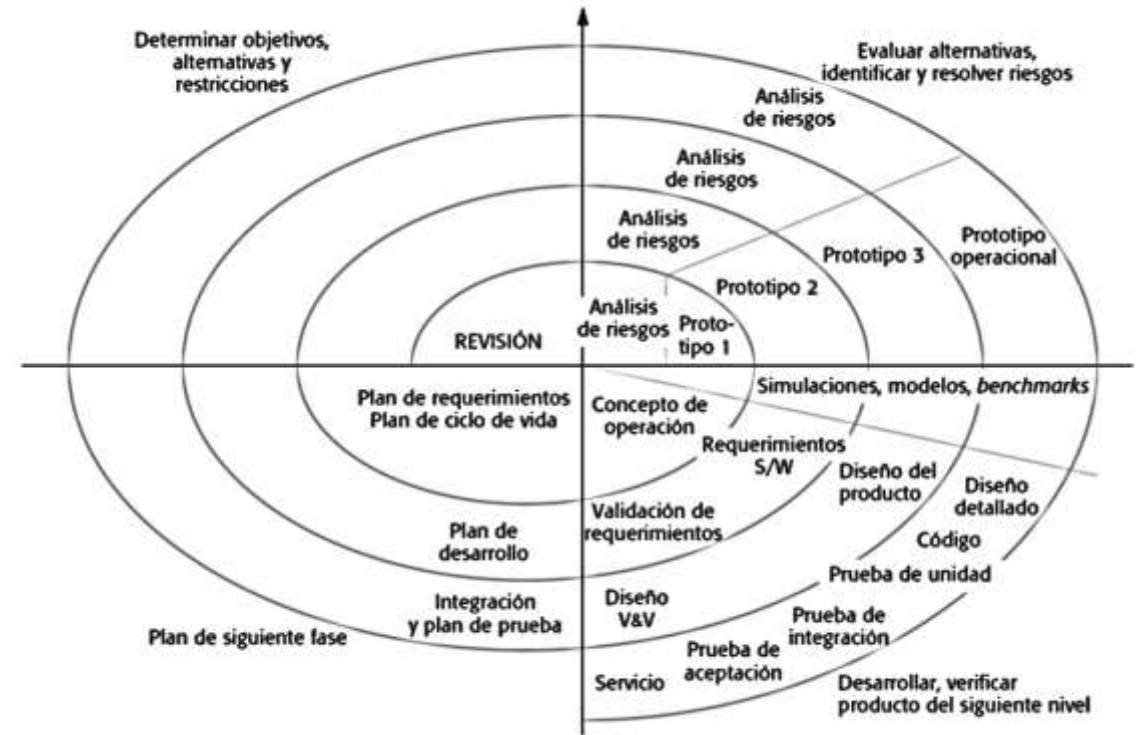
Pfleeger

# Modelo en espiral (Boehm)

- ❑ Combina las actividades de desarrollo con la gestión del riesgo
- ❑ Trata de mejorar los ciclos de vida clásicos y prototipos.
- ❑ Incorpora objetivos de calidad
- ❑ Elimina errores y alternativas no atractivas al comienzo
- ❑ Permite iteraciones, vuelta atrás y finalizaciones rápidas
- ❑ Cada ciclo empieza identificando:
  - ❖ Los objetivos de la porción correspondiente
  - ❖ Las alternativas

## Restricciones

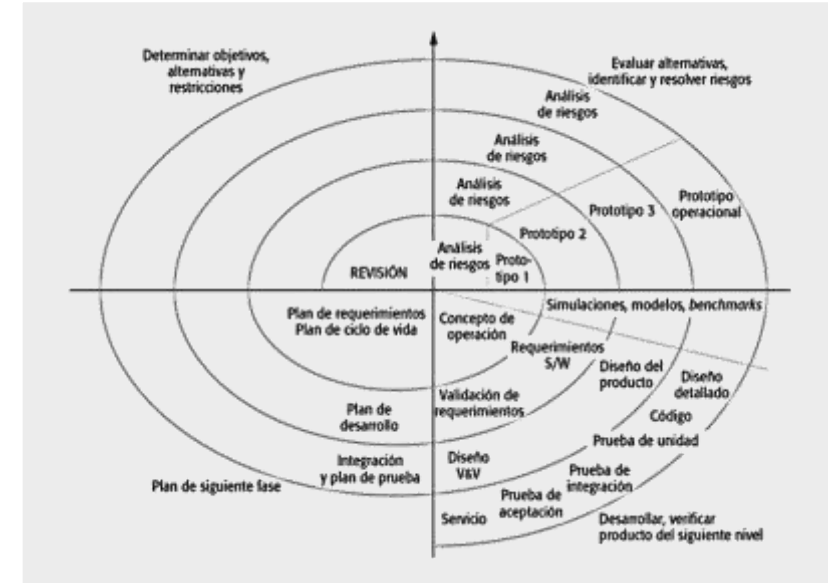
- ❑ Cada ciclo se completa con una revisión que incluye todo el ciclo anterior y el plan para el siguiente



# Modelo en espiral (Boehm)

Cada ciclo en la espiral se divide en cuatro sectores:

1. *Establecimiento de objetivos* . Se identifican restricciones, se traza un plan de gestión, se identifican riesgos
2. *Valoración y reducción del riesgo*. Se analiza cada riesgo identificado y se determinan acciones.
3. *Desarrollo y validación*. Se determina modelo de desarrollo.
4. *Planeación* . El proyecto se revisa y se toma decisiones para la siguiente fase



# Bibliografía

---

## ❑ Libros Utilizados en la Teoría

Sommerville, Capitulo 2 , Ingeniería de Software , Pearson 2011

Pressman, Capítulo 2, Ingeniería de Software Un enfoque práctico, Mc Graw Hill 2021

Pfleeger, Capitulo 2 Software Engineering. Theory and Practice . Prentice Hall 2010