

Tema 2:

Ciclo de vida del Software

Marcos López Sanz
Ingeniería del Software de Gestión

➤ ¿Qué es el ciclo de vida del Software?

❑ La norma 12207-2008

❑ Modelos de desarrollo

- ❑ Es una **sucesión de etapas** por las que pasa el software en su desarrollo, desde que se concibe la idea hasta que el software deja de utilizarse (obsolescencia)
- ❑ Cada etapa lleva asociada una serie de **actividades y tareas** que se deben realizar, y una serie de **documentos** que serán la salida de cada una de estas fases y que servirán de entrada a la fase siguiente

- ❑ Según la norma *ISO/IEC Standard 12207:2008: Software life-Cycle processes* propuesta por la ISO (International Organization for Standardization):

“Es un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, explotación y mantenimiento de un producto software, abarcando la vida del sistema desde la definición de requisitos hasta que se deja de utilizar”

- ❑ <http://www.iso.org/iso/home.htm>

❑ ¿Qué es un proceso?

✓ *Un proceso es un conjunto de actividades que se suceden siguiendo una ordenación temporal determinada*

❑ ¿Qué es una actividad?

✓ *Una actividad es un conjunto de tareas*

❑ ¿Qué es una tarea?

✓ *Una acción que transforma unas entradas en unas salidas*

❑ Fases genéricas en el ciclo de vida del SW:

✓ *Fase de definición. Tareas:*

- Ingeniería de sistemas
- Planificación del proyecto del SW
- Análisis de los requisitos

✓ *Fase de desarrollo. Tareas:*

- Diseño del SW
- Generación de código
- Prueba del SW

✓ *Fase de mantenimiento. Cambios:*

- Corrección
- Adaptación
- Mejora
- Prevención

❑ ¿Qué es el ciclo de vida del Software?

➤ La norma 12207-2008

❑ Modelos de desarrollo

❑ Según esta norma las actividades que se pueden llevar a cabo durante el ciclo de vida del SW se pueden agrupar en:

- ✓ *5 procesos principales*
- ✓ *8 procesos de soporte*
- ✓ *4 procesos de organización o generales*

Procesos principales

Adquisición

Desarrollo

Explotación

Suministro

Mantenimiento

Procesos generales

Gestión

Infraestructura

Mejora

Formación

Procesos de Soporte

Documentación

Gestión de la Config.

Aseg. de la calidad

Verificación

Validación

Revisión conjunta

Auditoría

Resolución problemas

Proceso de Desarrollo

*Actividad
de
Análisis*

*Actividad
de
Diseño*

*Actividad
de
Codificación*

*Actividad
de
Pruebas*

*Actividad
de
Integración*

*Actividad
de
Implantación*

Cada una de estas **actividades** está compuesta por diferentes **tareas**

- ❑ **Adquisición:** Actividades y tareas que el comprador, el cliente o el usuario realizan para adquirir un sistema, un servicio o un producto software:
 - ✓ *Preparación y publicación de ofertas*
 - ✓ *Selección del suministrador de SW*

- ❑ **Suministro:** Actividades y tareas del suministrador:
 - ✓ *Preparar contratos como respuesta a una petición de un comprador de un producto SW*
 - ✓ *Identificar los recursos necesarios para llevar a cabo con éxito el desarrollo del producto SW*

- ❑ **Desarrollo:** Actividades y tareas enfocadas a la obtención de un producto Software
 - ✓ *Análisis*
 - ✓ *Diseño*
 - ✓ *Codificación*
 - ✓ *Pruebas*
 - ✓ *Integración*
 - ✓ *Implantación*

- ❑ **Explotación:** Explotación del SW y soporte operativo a los usuarios

- ❑ **Mantenimiento:** Actividades que incluyen modificaciones del producto, tanto del código como de la documentación, debido a errores o a la necesidad de mejora o/y adaptación.
 - ✓ *Migración hacia un nuevo entorno operativo*
 - ✓ *Retirada del producto*

***Procesos de soporte:** dan soporte al resto de procesos y se aplican durante cualquier momento del ciclo de vida del SW*

- ❑ **Documentación:** Registrar la información producida por un proceso o actividad del ciclo de vida:
 - ✓ *Diseñar, editar, distribuir y mantener los documentos producidos durante el desarrollo del SW*
- ❑ **Gestión de la Configuración:** Actividades que controlan las modificaciones y versiones de los elementos.
 - ✓ *Registrar las peticiones de cambios e informar de los estados de éstos.*

- ❑ **Aseguramiento de la calidad:** Actividades para asegurar que los productos cumplen los requisitos especificados y se ajustan a los planes establecidos
- ❑ **Verificación:** Actividades para determinar el buen funcionamiento de un producto software
- ❑ **Validación:** Actividades para determinar si el producto cumple los requisitos previstos

- ❑ **Revisión conjunta:** Actividades que permiten determinar el estado de los productos en una determinada actividad del ciclo de vida o en una cierta fase del proyecto. Puede ser una reunión conjunta con el cliente, el grupo de desarrollo y los clientes potenciales para revisar el trabajo hecho
- ❑ **Auditorías:** Actividades que permiten determinar en unos momentos determinados si se han conseguido los objetivos propuestos: requisitos, cumplimiento del contrato, etc.

- ❑ **Resolución de problemas:** Actividades que permiten analizar y resolver los problemas o disconformidades con los requisitos o con el contrato, que hayan surgido durante el desarrollo, la explotación, el mantenimiento, o en cualquier otro momento.
- ✓ *Disponer de un medio documental que permita asegurar que todos los problemas se han tratado*

Procesos de soporte a la organización: gestión, formación del personal, mejora de los procesos, etc.

- ❑ **Gestión:** Actividades de planificación, seguimiento, control, revisión y evaluación.
- ❑ **Infraestructura:** Actividades para determinar la infraestructura necesaria para un proceso. Incluye HW, SW, instalaciones...
- ❑ **Mejora:** Valorar, medir, controlar, evaluar y mejorar todos los procesos del ciclo de vida.
- ❑ **Formación:** Plan de formación para los empleados.

❑ ¿Qué es el ciclo de vida del Software?

❑ La norma 12207-2008

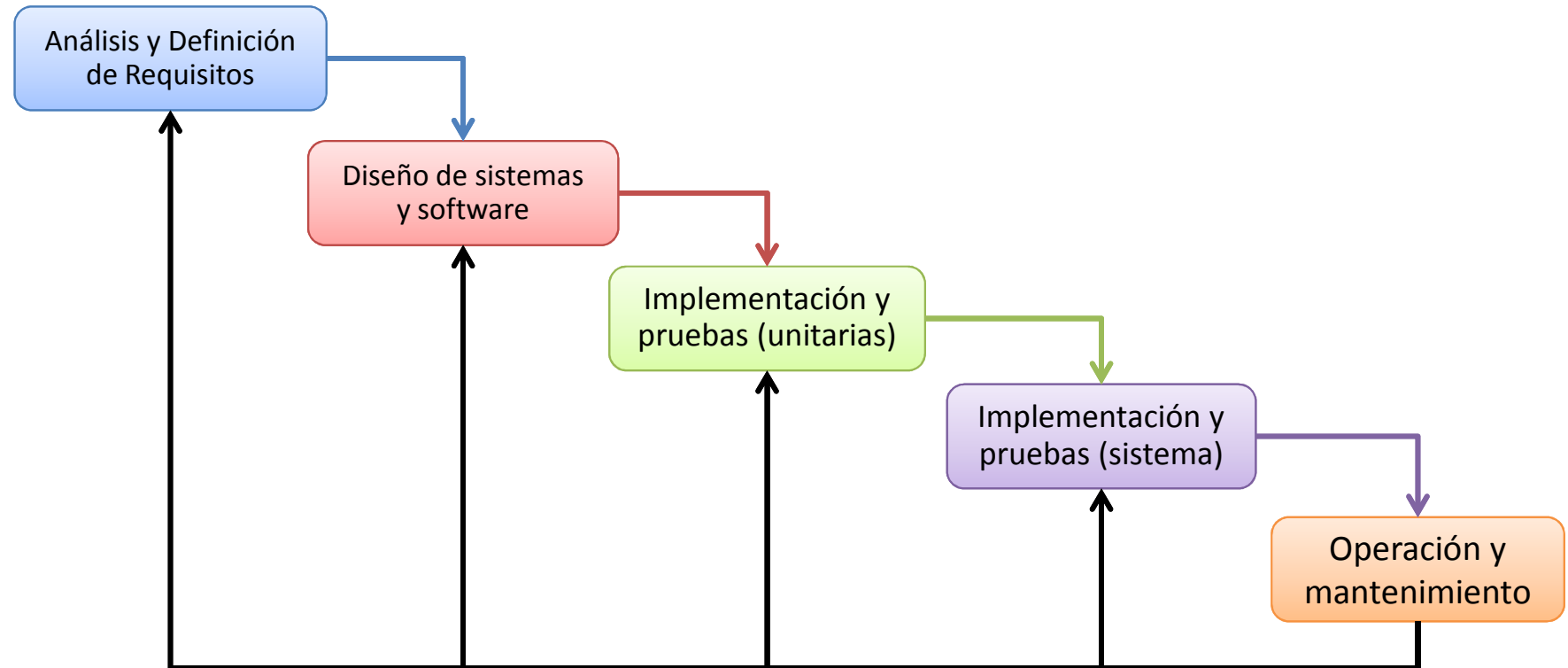
➤ Modelos de desarrollo

❑ ¿Qué es un modelo de desarrollo?

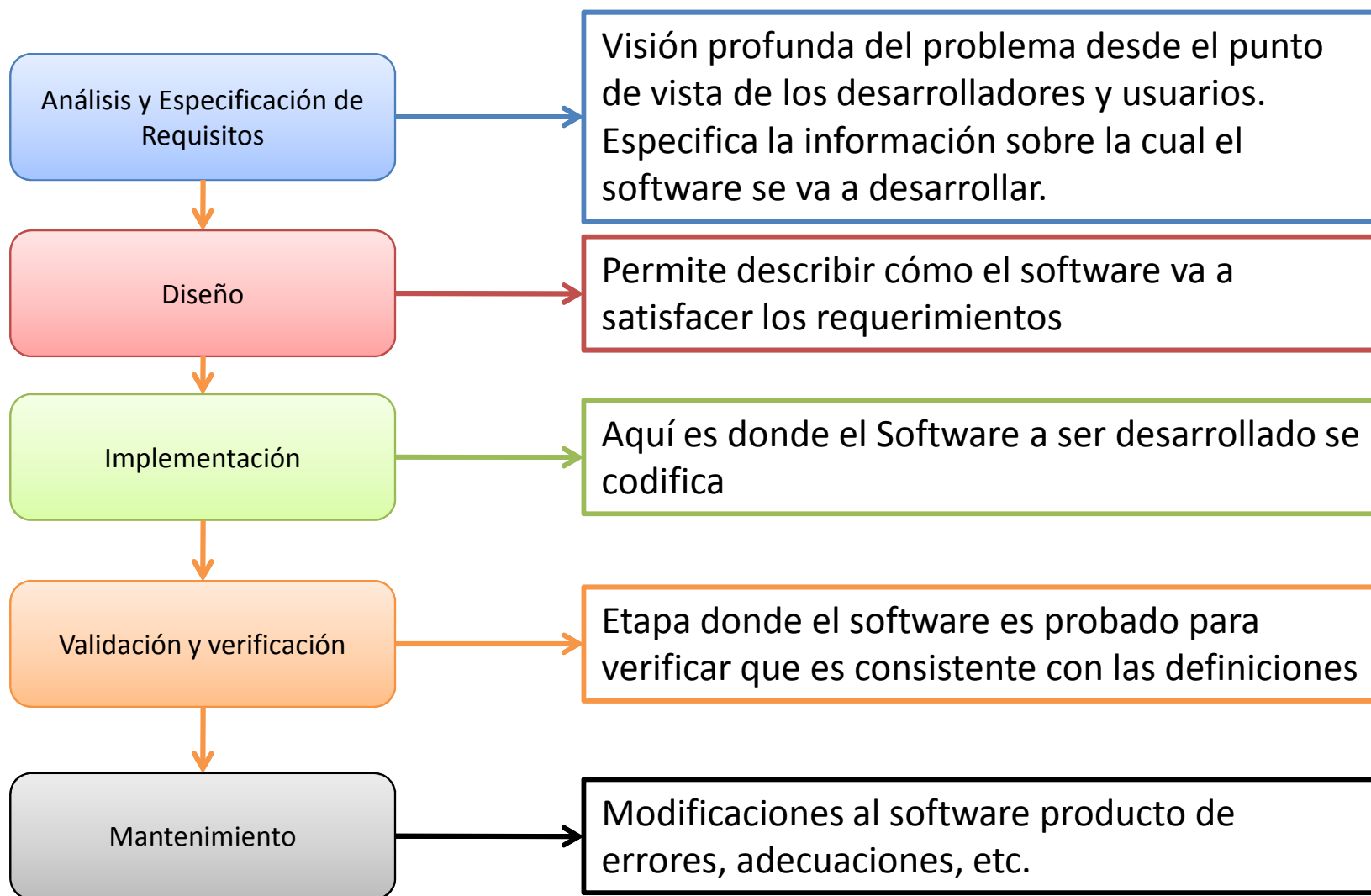
- ✓ *Representación abstracta de un proceso del software*
- ✓ *Son **estrategias de desarrollo** que ayudan a organizar las diferentes etapas y actividades del ciclo de vida del software*
→ *Modelos de ciclo de vida del software*
- ✓ *Estos modelos ayudan al control y a la coordinación del proyecto*
- ✓ *El modelo a utilizar depende del tipo de proyecto*

- ☐ Modelo en Cascada
- ☐ Modelo de Prototipos
- ☐ Modelo en Espiral
- ☐ Modelo Incremental
- ☐ Otros

- ❑ También llamado *ciclo de vida básico o modelo lineal-secuencial*
- ❑ Divide el proceso de desarrollo en un conjunto de etapas **secuenciales**
- ❑ Una etapa no puede empezar hasta que no haya terminado la anterior
- ❑ Al final de cada fase, el personal de desarrollo y los usuarios **revisan** el progreso del proyecto
- ❑ En cada fase se genera todo un conjunto de documentos. Es un *modelo dirigido por documentos* → Son los productos principales en cada etapa



- ❑ Es el modelo más antiguo y más utilizado
- ❑ Ha sido la base de muchos otros modelos



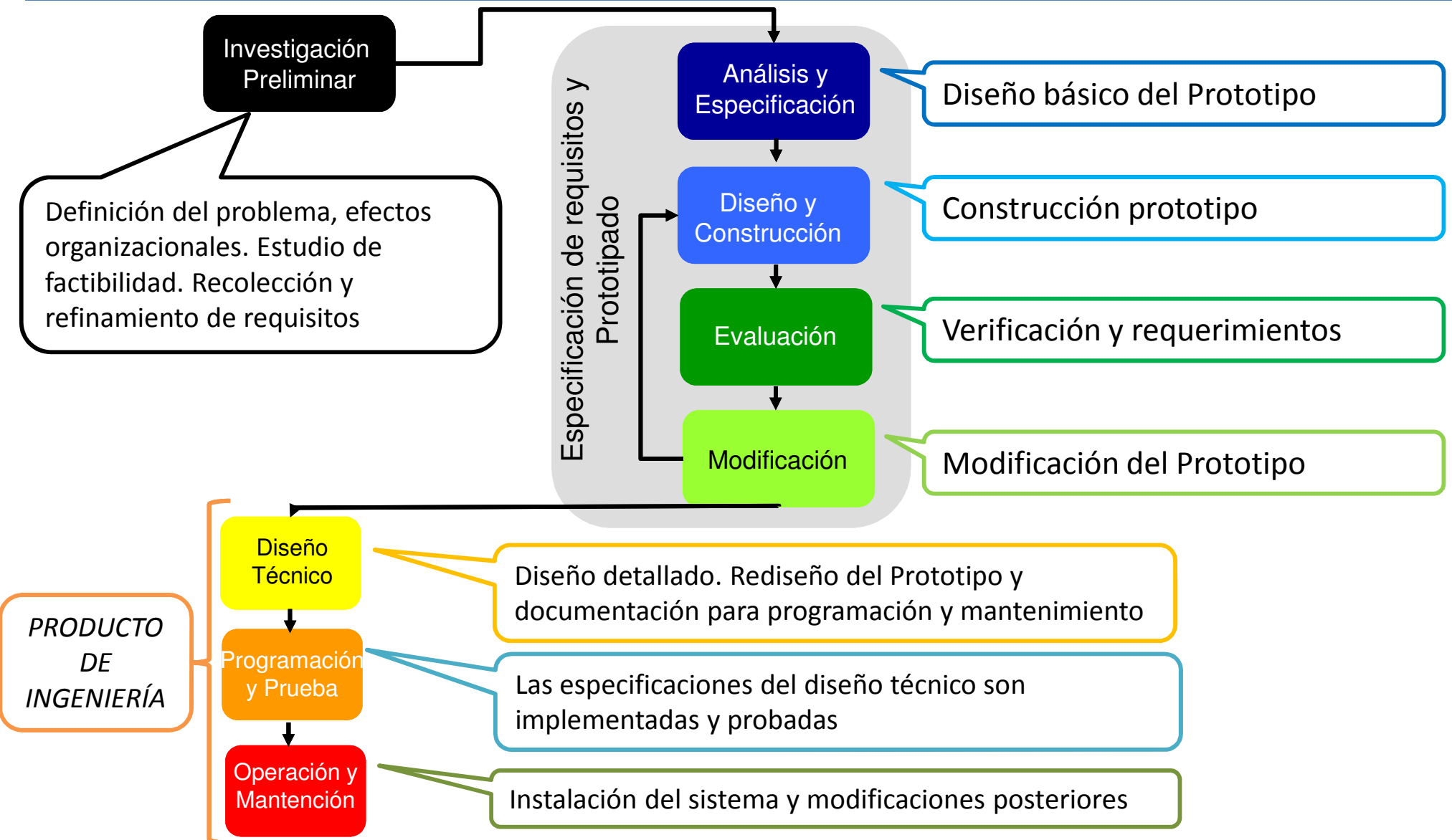
❑ Desventajas:

- ✓ *Definir todos los requisitos al inicio del proceso no es práctico ya que el cliente añade y modifica según le van surgiendo necesidades durante el proceso de desarrollo.*
- ✓ *¡El cliente nunca tiene claro lo que quiere!*
- ✓ *El cliente no ve el producto en funcionamiento hasta el final del proceso.*
- ✓ *La validación de los requisitos iniciales no se hace hasta el final.*
- ✓ *Poca o nula flexibilidad a cambios.*

❑ ¿Cuándo es conveniente? (Ventajas):

- ✓ *Cuando tenemos proyectos complejos pero que se entienden y quedan bien definidos desde un comienzo.*
- ✓ *Cuando el equipo de desarrollo no está muy cualificado o es aún inexperto porque la estructura de trabajo que propone es muy ordenada y ayuda a minimizar esfuerzo.*
- ✓ *Cuando realizamos una migración de software desde un entorno tecnológico obsoleto.*

- ❑ Un **prototipo** es una versión limitada del producto que permite a las partes responsables de su creación probarlo en situaciones reales y explorar su uso
- ❑ Con este modelo hay un acercamiento al cliente. Gracias al prototipo, el cliente puede hacerse una idea de cómo está evolucionando el producto y esto ayuda a refinar los requisitos del sistema
- ❑ Con los prototipos definimos de forma clara y concreta qué quiere el cliente → *los requisitos del sistema*



❑ Desventajas:

- ✓ *El diseño rápido del prototipo hace que los desarrolladores utilicen herramientas que faciliten la rápida generación de código, dejando a un lado aspectos de calidad (eficiencia, fiabilidad, mantenibilidad del código, etc.).*
- ✓ *Probablemente no se tendrá un código óptimo*
- ✓ *Exige disponer de las herramientas adecuadas*

❑ ¿Cuándo es conveniente? (Ventajas):

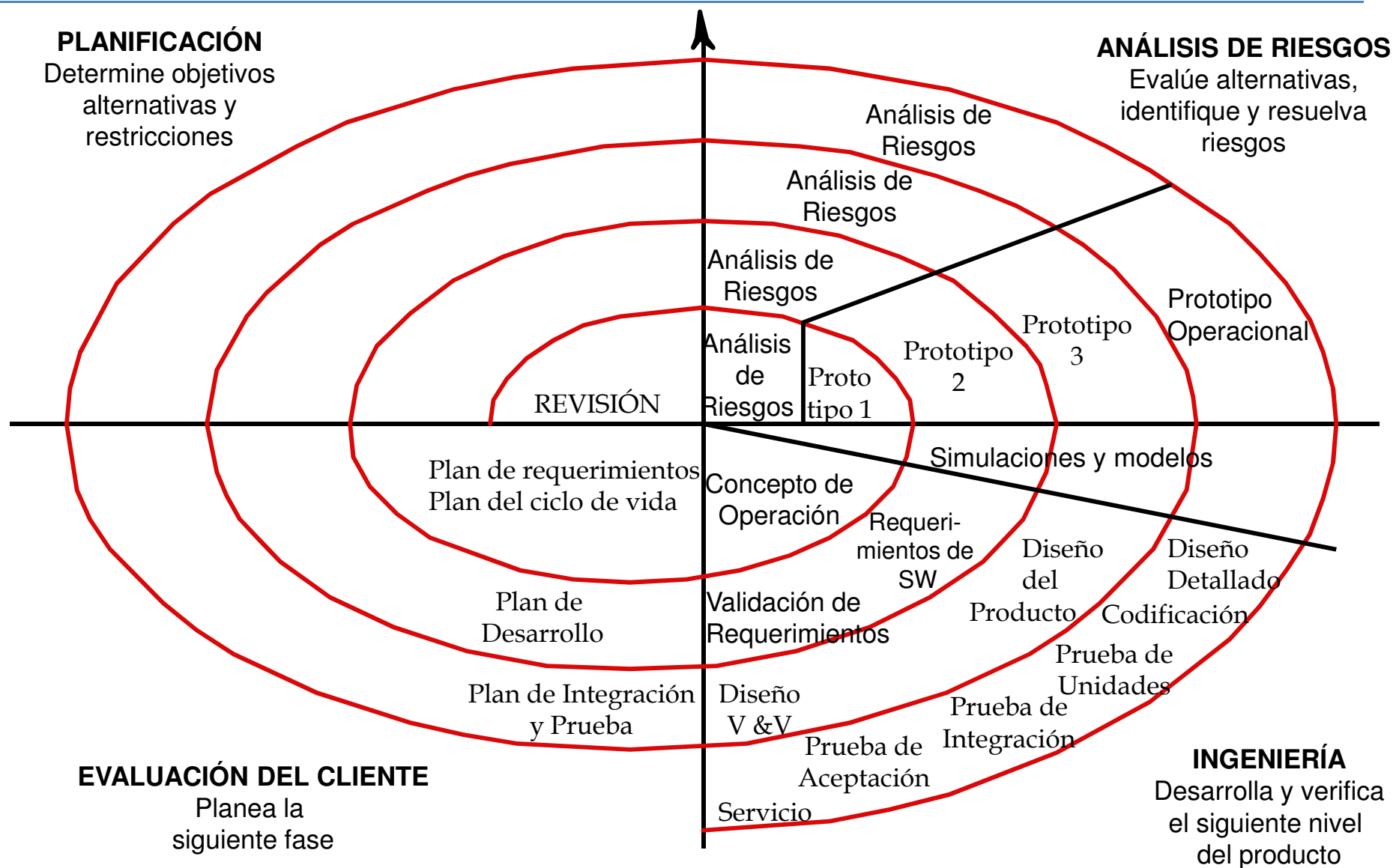
- ✓ *Se recomienda para clientes que quieren ver resultados a corto plazo → Reduce costos y aumenta la probabilidad de éxito*
- ✓ *Cuando el cliente no sabe lo que quiere y los requisitos no están bien definidos desde el principio → Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios*
- ✓ *Cuando los requisitos evolucionan muy rápidamente.*
- ✓ *Para sistemas on-line donde es más importante la parte de la interfaz con el usuario que las funcionalidades del sistema.*
- ✓ *Se recomienda utilizar este modelo sólo para la especificación de requisitos. Mejor continuar el desarrollo utilizando otro modelo*

- ❑ Es una combinación de los dos anteriores (Boehm'88)
- ❑ Es un modelo evolutivo del desarrollo, formado por un conjunto de vueltas de espiral
 - ✓ *En las primeras vueltas el SW es un modelo en papel, la especificación de un producto. Aún no funciona.*
 - ✓ *En las sucesivas vueltas, se desarrolla un prototipo.*
 - ✓ *En la últimas iteraciones se obtienen versiones completas del producto.*
- ❑ Cada ciclo del espiral representa una fase del proyecto software
- ❑ 4 sectores por ciclo:
 - ✓ *Definición de Objetivos*
 - ✓ *Evaluación y reducción de riesgos*
 - ✓ *Desarrollo y Validación*
 - ✓ *Planificación*

- ❑ Con este modelo obtenemos el producto final a partir de piezas más pequeñas.
- ❑ El número de actividades a realizar se incrementa notablemente a medida que nos alejamos del centro de la espiral. Las primeras son menos costosas
- ❑ La evaluación después de cada fase permite cambios.
- ❑ Incorpora el factor Riesgo → es un modelo orientado a riesgos
 - ✓ *Tiene como objetivo vital pensar en las cosas que pueden ir mal en el desarrollo del software y saber cómo resolverlas*

Modelos de desarrollo

Modelo en Espiral



❑ Planificación

- ✓ *Determinar los objetivos*
- ✓ *Determinar las alternativas de desarrollo*
- ✓ *Analizar las restricciones de cada alternativa*

❑ Análisis de riesgos

- ✓ *Ver todos los puntos que pueden fallar*
- ✓ *Evaluar las diferentes alternativas*
- ✓ *Determinar y resolver o minimizar los riesgos*
- ✓ *Identificación de los riesgos para cada alternativa, así como la manera de resolverlos*

❑ Ingeniería

- ✓ *Desarrollo del producto*
- ✓ *En cada iteración el proyecto se va completando*

❑ Evaluación del cliente

- ✓ *Revisión para ver si está de acuerdo, o no, con los resultados obtenidos. Si todo va bien, se pasa a la siguiente fase*
- ✓ *En la revisión participan todas las personas y organizaciones que tienen relación con el producto*
- ✓ *Se planifica la siguiente vuelta. Previsión de los recursos necesarios*

❑ Desventajas:

- ✓ *Es difícil establecer los hitos para determinar si podemos pasar a la siguiente vuelta de espiral*
- ✓ *La evaluación de riesgos es una tarea compleja → Se necesitan expertos en evaluación de riesgos*
- ✓ *Es difícil convencer a los clientes que un modelo evolutivo es controlable*
- ✓ *No se aconseja para proyectos que tienen pocos riesgos. Demasiado coste*
- ✓ *En definitiva, no es un modelo muy real ni claro.*

❑ ¿Cuándo es conveniente? (Ventajas):

- ✓ *Cuando tenemos proyectos complejos, donde el problema no está muy bien definido y conlleva una serie de riesgos*
- ✓ *Proyectos dinámicos*
- ✓ *Proyectos innovadores y ambiciosos*

- ❑ Es un tipo de modelo evolutivo → es iterativo: permiten a los ingenieros desarrollar versiones cada vez más completas
- ❑ Combina elementos del modelo en cascada (aplicados repetidamente) con la filosofía interactiva de la construcción de prototipos
- ❑ Cada secuencia lineal produce un incremento → las entregas de los incrementos se definen al principio del proceso software
- ❑ Cada entrega constituye un producto operacional
- ❑ Es útil cuando el personal o los recursos no están disponibles hasta cierto tiempo dentro del proceso de desarrollo → Se adapta a entornos de alta incertidumbre
- ❑ El proceso no es visible, documentación costosa y difícil planificación

❑ El modelo en espiral: ¿es iterativo e incremental?

- ✓ *Es iterativo (se realizan ciclos repetitivos), pero no necesariamente incremental.*
- ✓ *Para que un modelo se considere incremental los incrementos deben decidirse durante el análisis/elicitación de los requisitos y deben condicionar las iteraciones que se hacen, y no al revés.*
- ✓ *Es decir, durante las iteraciones del modelo de ciclo en espiral los incrementos, de producirse, se establecen en la planificación de la propia iteración, no a priori. Es más, el incremento lo constituyen los productos de la iteración, no conocidos hasta que no se obtienen realmente.*

- ❑ Métodos formales (síntesis automática del Software)
- ❑ Desarrollo orientado a la reutilización (basado en componentes)
- ❑ DRA (*Desarrollo Rápido de Aplicaciones*)
- ❑ Espiral WINWIN
- ❑ Desarrollo concurrente
- ❑ Técnicas de 4ª generación
- ❑ Modelos para Desarrollo OO:
 - ✓ *Modelo de Agrupamiento*
 - ✓ *Modelo fuente*
 - ✓ *Modelo remolino*
 - ✓ *Modelo Pinball, etc.*