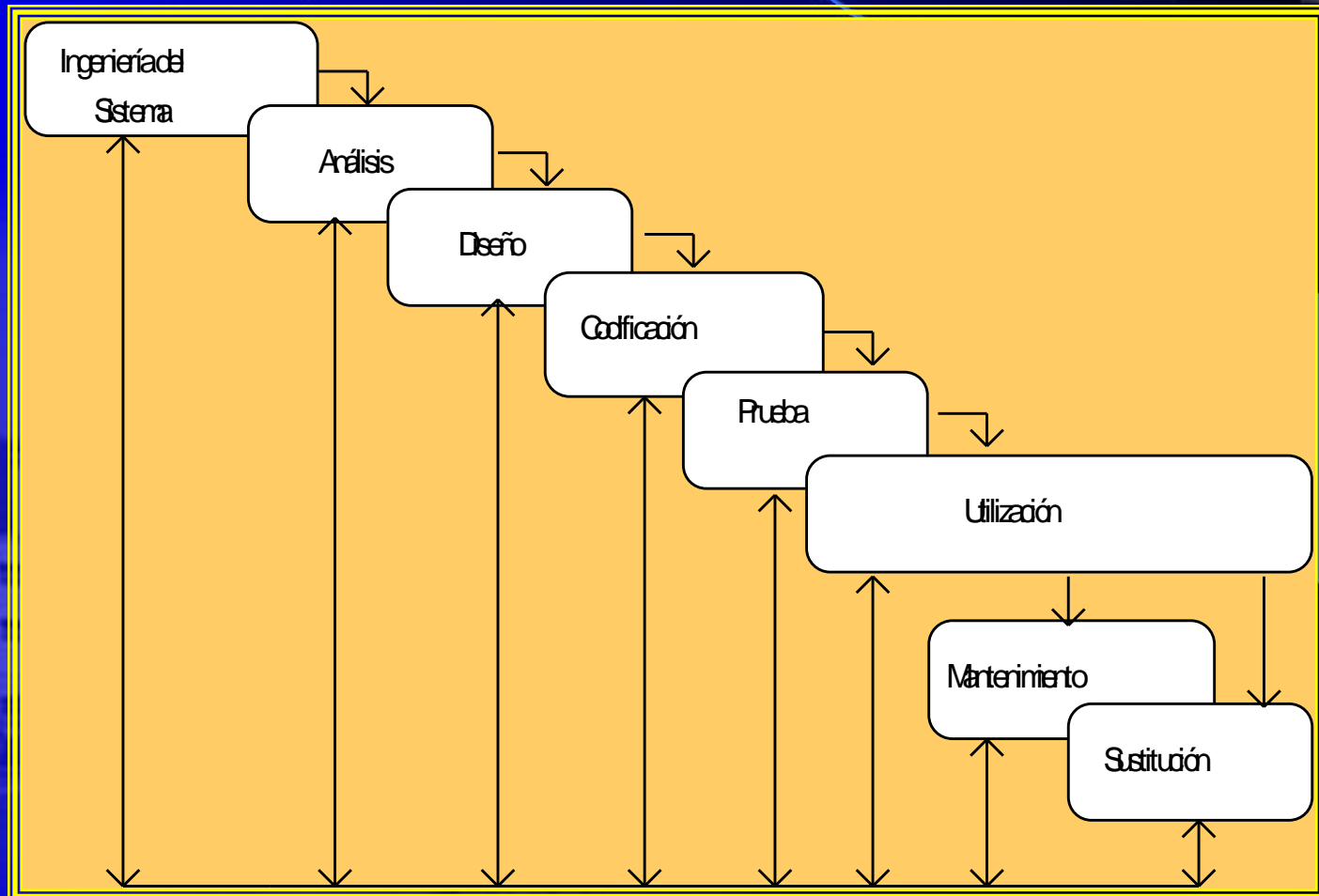
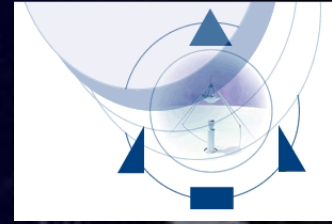


Ciclo de vida en cascada



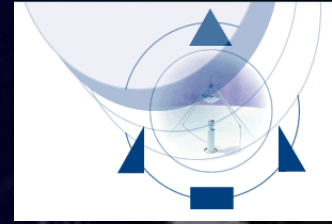
Ciclo de vida en cascada



- Aportaciones

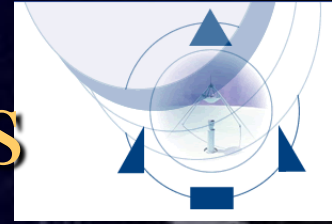
- Es el más simple, el más conocido y el más fácil de usar.
- Sistematiza el desarrollo
 - Define fases de desarrollo diferenciadas
 - Al final de cada fase el personal técnico y los usuarios tienen la oportunidad de revisar el progreso del proyecto.
 - Permite analizar las fechas de entrega y los costes esperados
- Permite generar software eficientemente y de acuerdo con las especificaciones.

Ciclo de vida en cascada

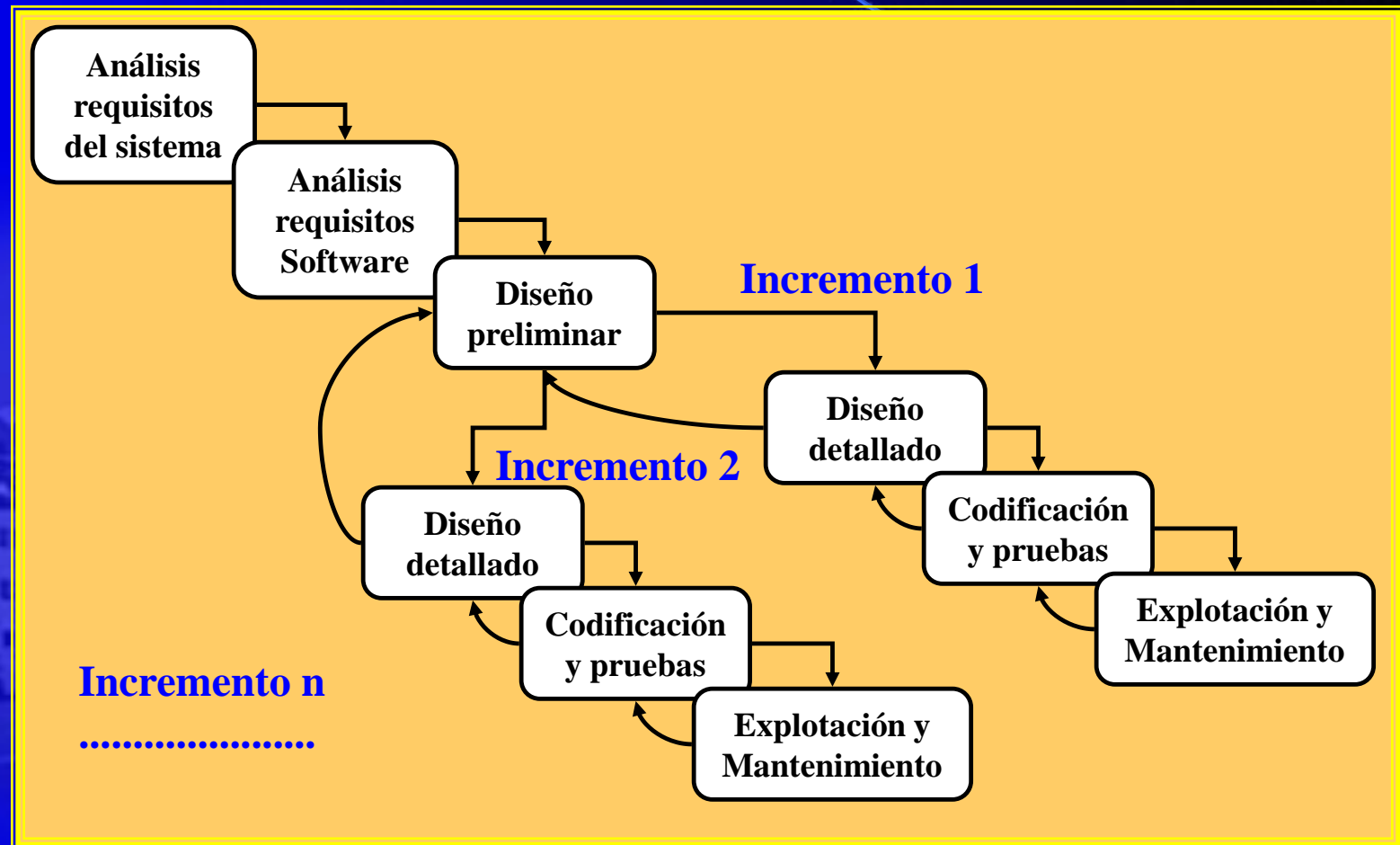


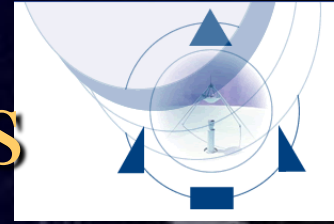
- Problemas/Críticas

- En la realidad el ciclo de vida no es secuencial, sino que hay iteraciones
- No siempre se pueden establecer los requisitos del sistema desde el primer momento
- Hasta el final no hay una versión operativa del programa por lo que es al final de la fase de codificación cuando se detectan la mayoría de los fallos
- Estados de bloqueo, porque hay partes que tienen que esperar por otras.



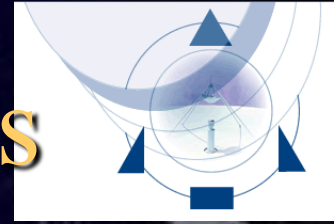
Desarrollo en incrementos





Desarrollo en incrementos

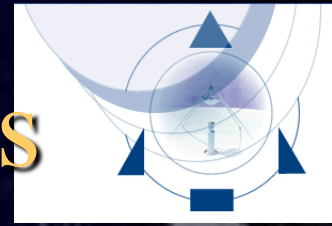
- Avances
 - El software no se piensa como una unidad monolítica sino la integración de resultados sucesivos.
 - Adecuado a entornos de alta incertidumbre.
- Ventajas
 - Modelo iterativo.
 - Mejora la comunicación con el cliente.
- Críticas:
 - Dificultad de ver si los requisitos son válidos.
 - Los errores en los requisitos se detectan tarde.



Construcción de prototipos

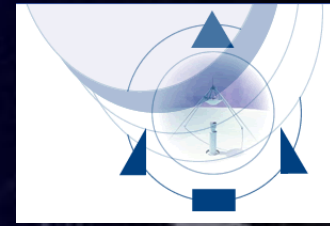


Ingeniería del Software



Construcción de prototipos

- Criterios de selección
 - Mucha interacción con el usuario, algoritmos refinables
 - Mucha Complejidad => Mal candidato.
 - Disposición del cliente a probar los prototipos.
- Ventajas
 - Proporciona un excelente lenguaje de especificación de requisitos.
 - El cliente tiene una idea muy clara de lo que se está haciendo.
 - Parte del trabajo realizado en la especificación puede ser utilizado en el desarrollo.
- Desventajas
 - El prototipo se convierte en el resultado final
 - Se asumen elecciones apresuradas de arquitectura, plataforma de desarrollo; contiene gran cantidad de errores latentes, será poco eficiente y fiable y difícil de mantener



Modelo en Espiral

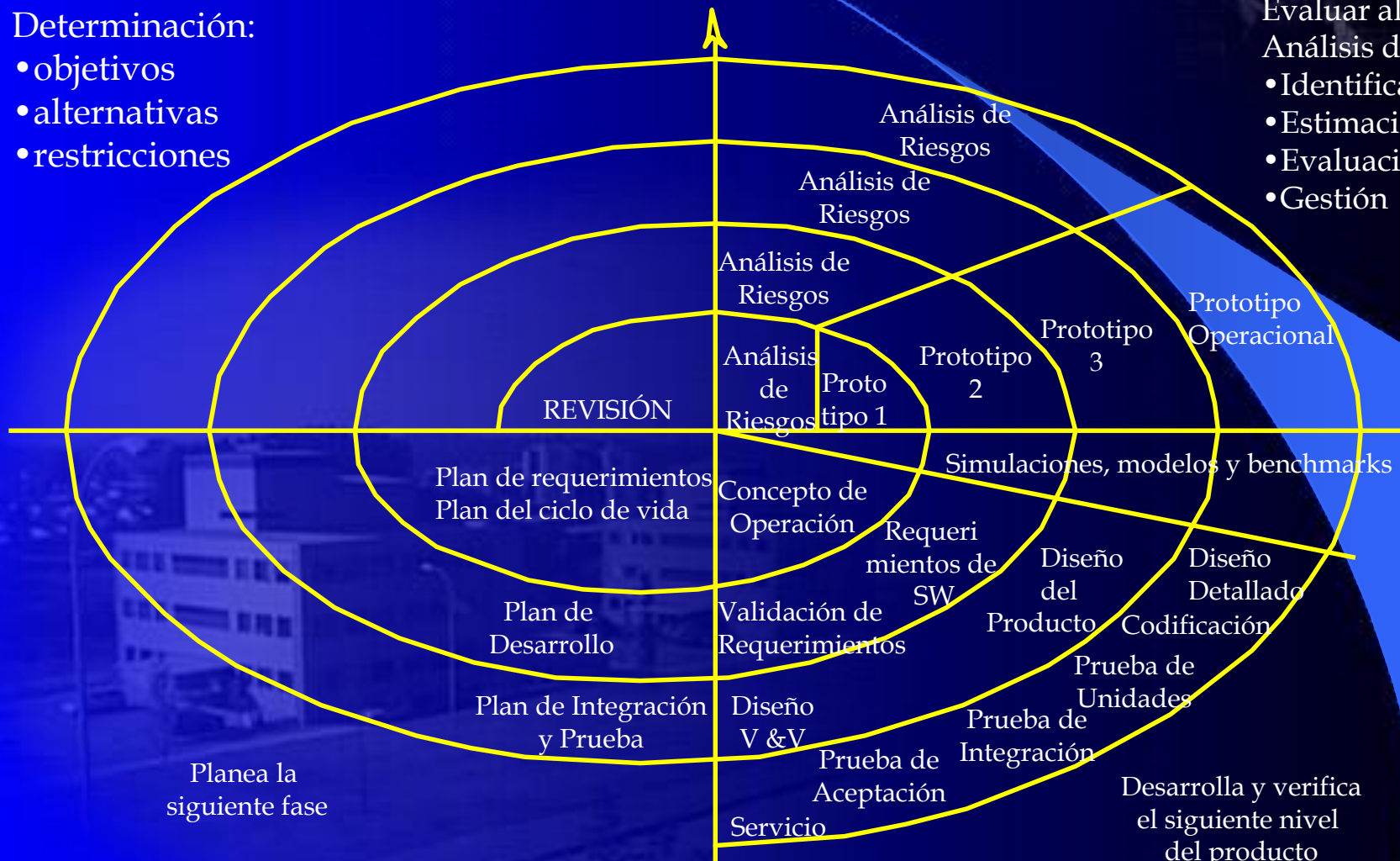
Determinación:

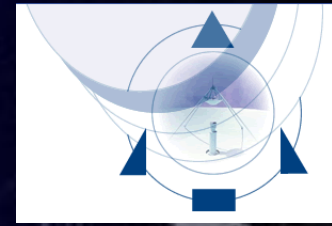
- objetivos
- alternativas
- restricciones

Evaluar alternativas.

Análisis de riesgos:

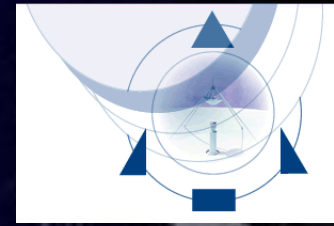
- Identificación
- Estimación
- Evaluación
- Gestión





Modelo en Espiral Cascada + Prototipos

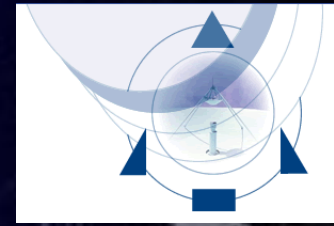
- Proporciona un modelo evolutivo para el desarrollo de sistemas de software complejos
 - Más realista que el ciclo de vida clásico.
 - Mantiene el enfoque sistemático del desarrollo secuencial.
 - Permite la utilización de prototipos, pero cada iteración implica la generación de un producto operativo
 - Incorpora análisis de riesgos.
 - Propone un análisis inicial de alternativas.
 - Define estrategias para la reducción de riesgos (Prototipos).
 - Permite acabar en fases tempranas el desarrollo de un producto final demasiado costoso o arriesgado



Modelo en Espiral

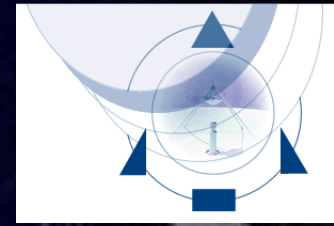
● Limitaciones

- Es difícil de adaptar a un contrato.
- Puede ser difícil de controlar y de convencer al cliente de que es controlable
- Requiere habilidad en la gestión de riesgos.
Necesidad de expertos
- Un riesgo no detectado a tiempo equivale en coste a un requisito mal definido de los modelos secuenciales.



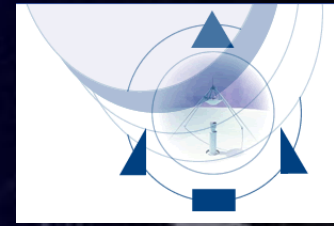
Desarrollo Ágil: XP

- Trata de aliviar el peso de las metodologías reduciendo la presión sobre la calidad de los modelos.
 - Los modelos deben ser SUFICIENTEMENTE buenos
- Trata de superar limitaciones de las metodologías pesadas
 - Incapacidad para adaptarse al cambio
 - Fragilidad asociada a las debilidades humanas



Desarrollo Ágil: XP

- Acelerar al máximo las entregas de software.
- Hacer participe activo al cliente.
 - Evaluación del producto entregado.
 - Nuevas requisitos.
 - Nuevas prioridades.
- Tolerancia en la aplicación de los modelos permitiendo que estos se adapten al equipo y no al revés.
- Admitir la falta de eficiencia en el planteamiento.



Desarrollo Ágil: XP

- Ventajas

- Fácil adaptación a los cambios.
- Reduce el peso de la documentación
- Facilita el análisis con el cliente de problemas mal definidos

- Desventajas

- Es muy sensible al personal
 - Cambios, rotaciones, modificaciones en plantilla
 - Plantilla inexperta, sin habilidades interpersonales o conflictiva
 - Disponibilidad del cliente
- Es una mala solución
 - En problemas bien comprendidos
 - Con necesidades de documentación rigurosa. Contratos. Normas