Modelo de Procesos de Software

- **Lineal Secuencial.**
- **Prototipo.**
- **Desarrollo de Aplicaciones Rápida.(D.R.A).**

Lineal Secuencial

- **Análisis de los requisitos del software.**
- **Diseño.**
- **Generación de código**
- **Pruebas**
- **Mantenimiento**

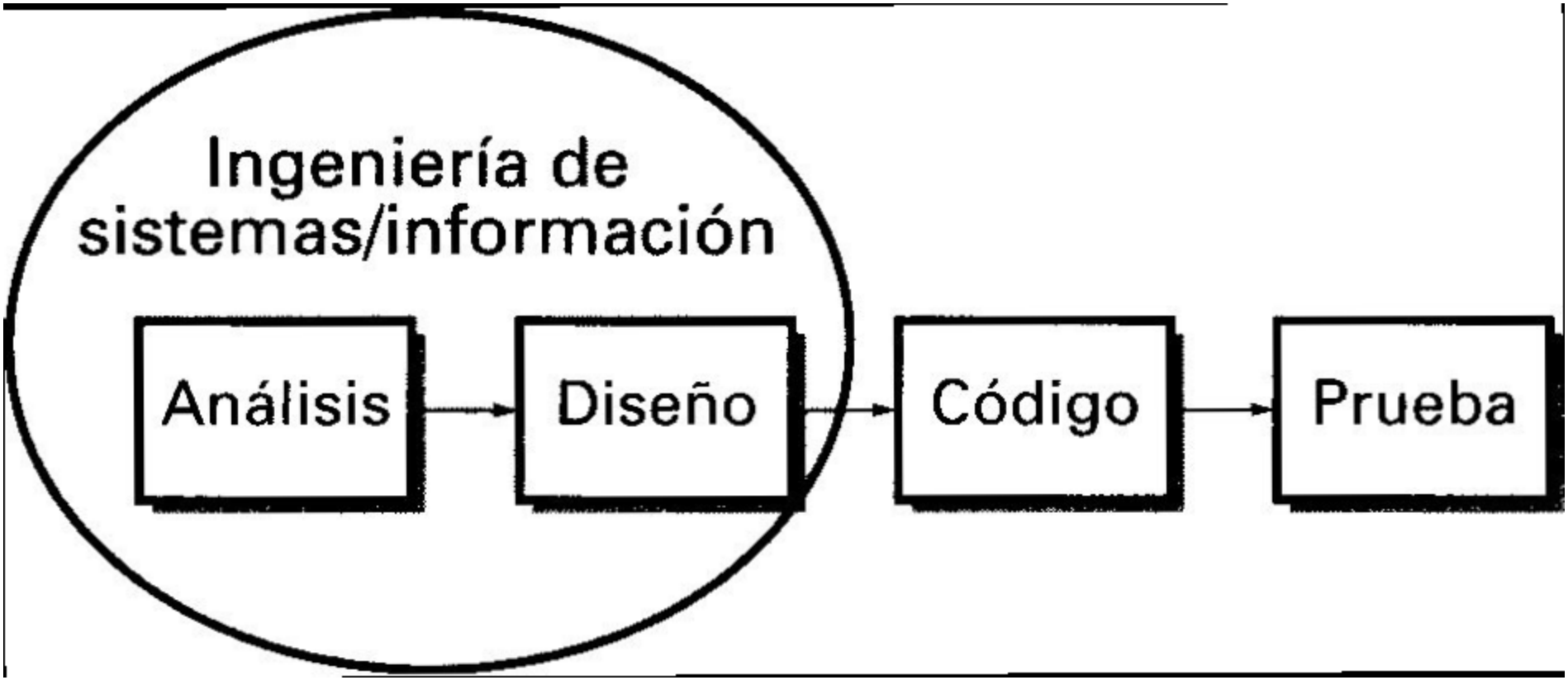
Lineal Secuencial

¿Por qué falla algunas veces el modelo lineal?

- Los proyectos reales raras veces siguen el modelo secuencial que propone el modelo.
- A menudo es difícil que el cliente exponga explícitamente todos los requisitos
- El cliente debe tener paciencia.

El modelo lineal

- Proporciona una plantilla en la que se encuentran métodos para análisis, diseño, codificación, pruebas y mantenimiento.
- El ciclo de vida clásico sigue siendo el modelo de proceso extensamente utilizado por la ingeniería del software.
- Pese a tener debilidades, es significativamente mejor que un enfoque hecho al azar para el desarrollo del software.

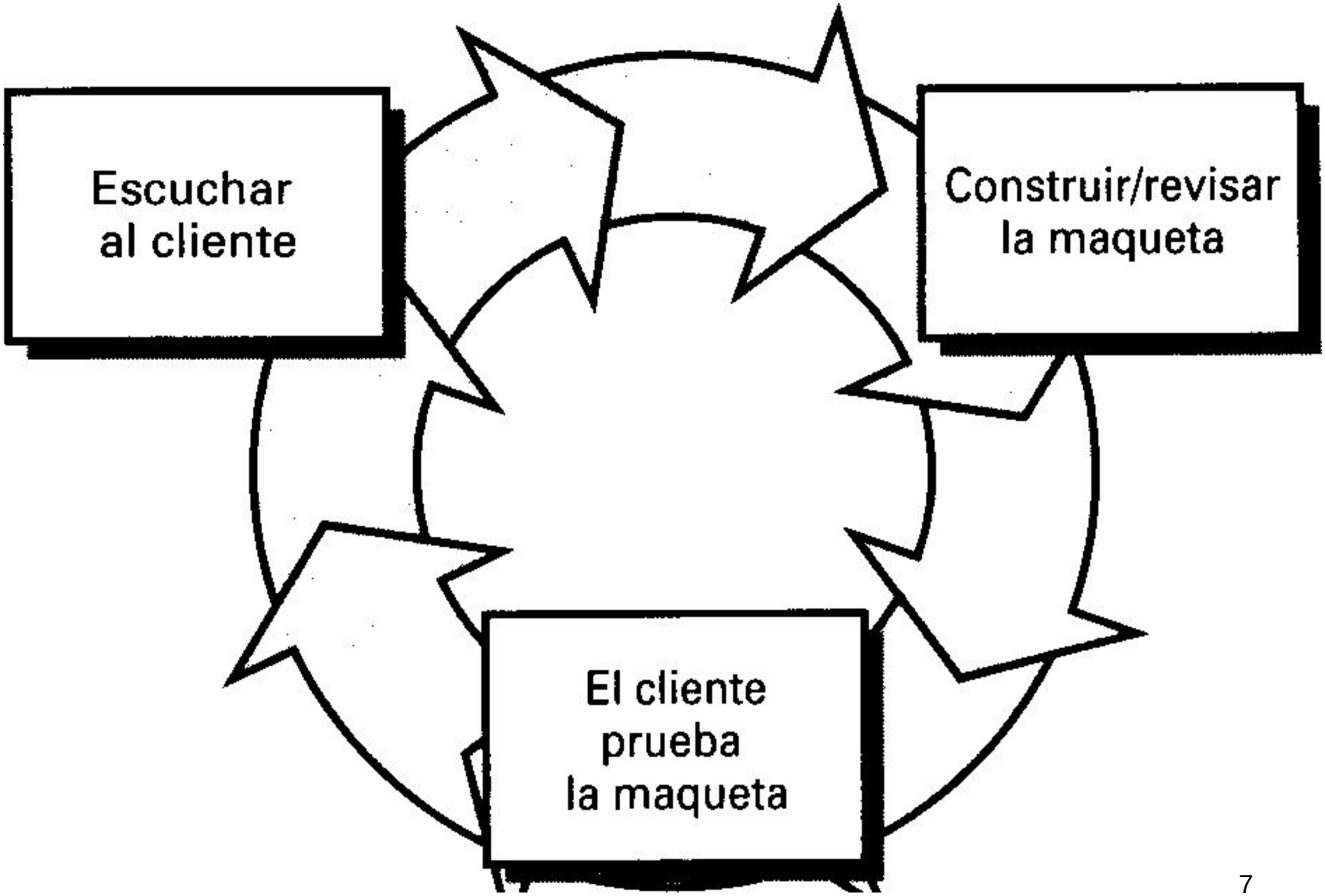

Ingeniería de sistemas/información
Análisis
Diseño
Código
Prueba

PROTOTIPOS

- Un cliente, a menudo, define un conjunto de objetivos generales para el software, pero no identifica los requisitos detallados de entrada, proceso o salida.
- En otros casos, el responsable del desarrollo del software puede no estar seguro de la eficacia de un algoritmo, de la capacidad de adaptación de un sistema operativo, o de la forma en que debería tomarse la interacción hombre máquina.

UN *PARADIGMA DE CONSTRUCCIÓN DE PROTOTIPOS PUEDE OFRECER* EL MEJOR ENFOQUE.

PROTOTIPOS


Escuchar al cliente
Construir/revisar la maqueta
El cliente prueba la maqueta

PROTOTIPOS

- El prototipo puede servir como «primer sistema».
- El cliente ve lo que parece ser una versión de trabajo del software.

Modelo Desarrollo de Aplicaciones Rápidas(D.R.A)

DRA adaptación a «alta velocidad» del modelo lineal secuencial. Se logra el desarrollo rápido utilizando una construcción basada en componentes.

SI

- Se comprenden bien los requisitos
- Se limita el ámbito del proyecto

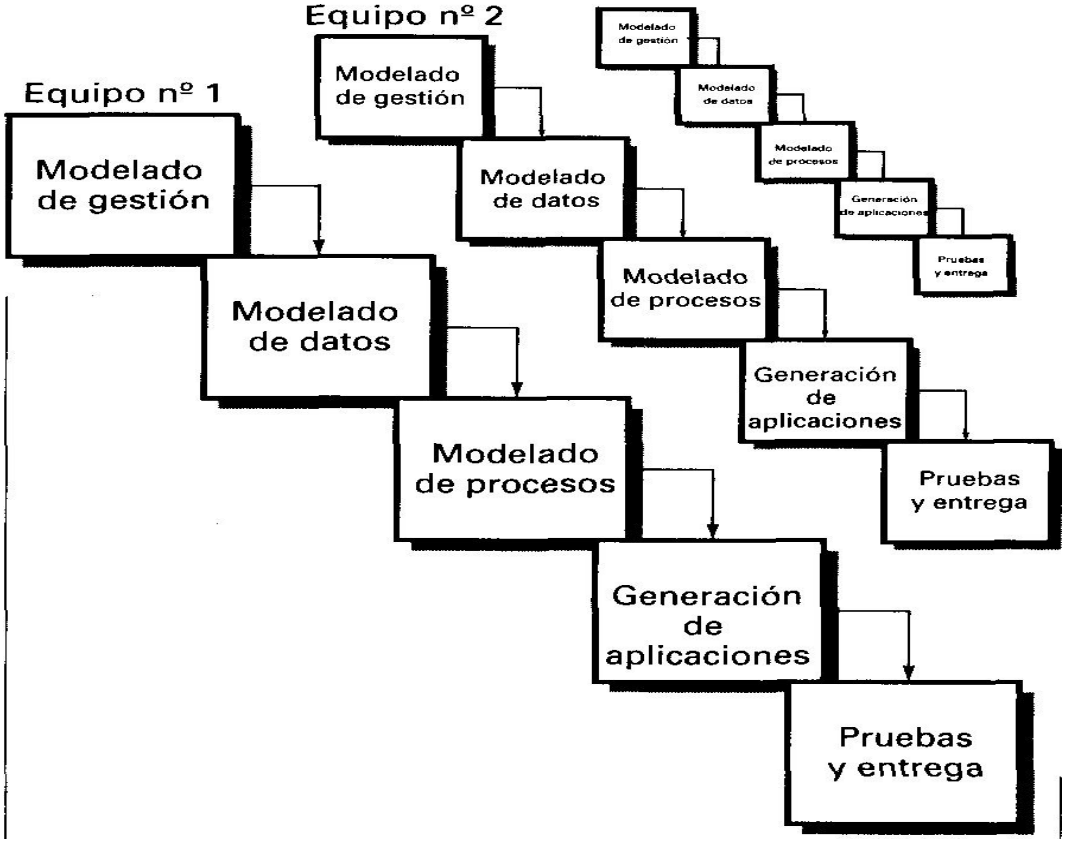
DRA permite crear un «sistema completamente funcional»

Modelado de datos. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.

Modelado del proceso. Los objetos de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos.

Modelo Desarrollo de Aplicaciones Rápidas(D.R.A)

- Utilización de técnicas de cuarta generación
- El proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario).
- Se utilizan herramientas para facilitar la construcción del software.



Modelo de Desarrollo Rápido. (D.R.A)

Problemas:

- requiere recursos humanos suficientes como para crear el número correcto de equipos DRA.
- DRA requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado.
- No todos los tipos de aplicaciones son apropiados para DRA:
  - **Si un sistema no se puede modularizar** adecuadamente, la construcción de los componentes necesarios para **DRA será problemático.**
  - **Si está en** juego el alto rendimiento, y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistemas, el enfoque **DRA puede que no funcione.**
  - **DRA no es adecuado cuando los riesgos técnicos son** altos.

Modelos Evolutivos

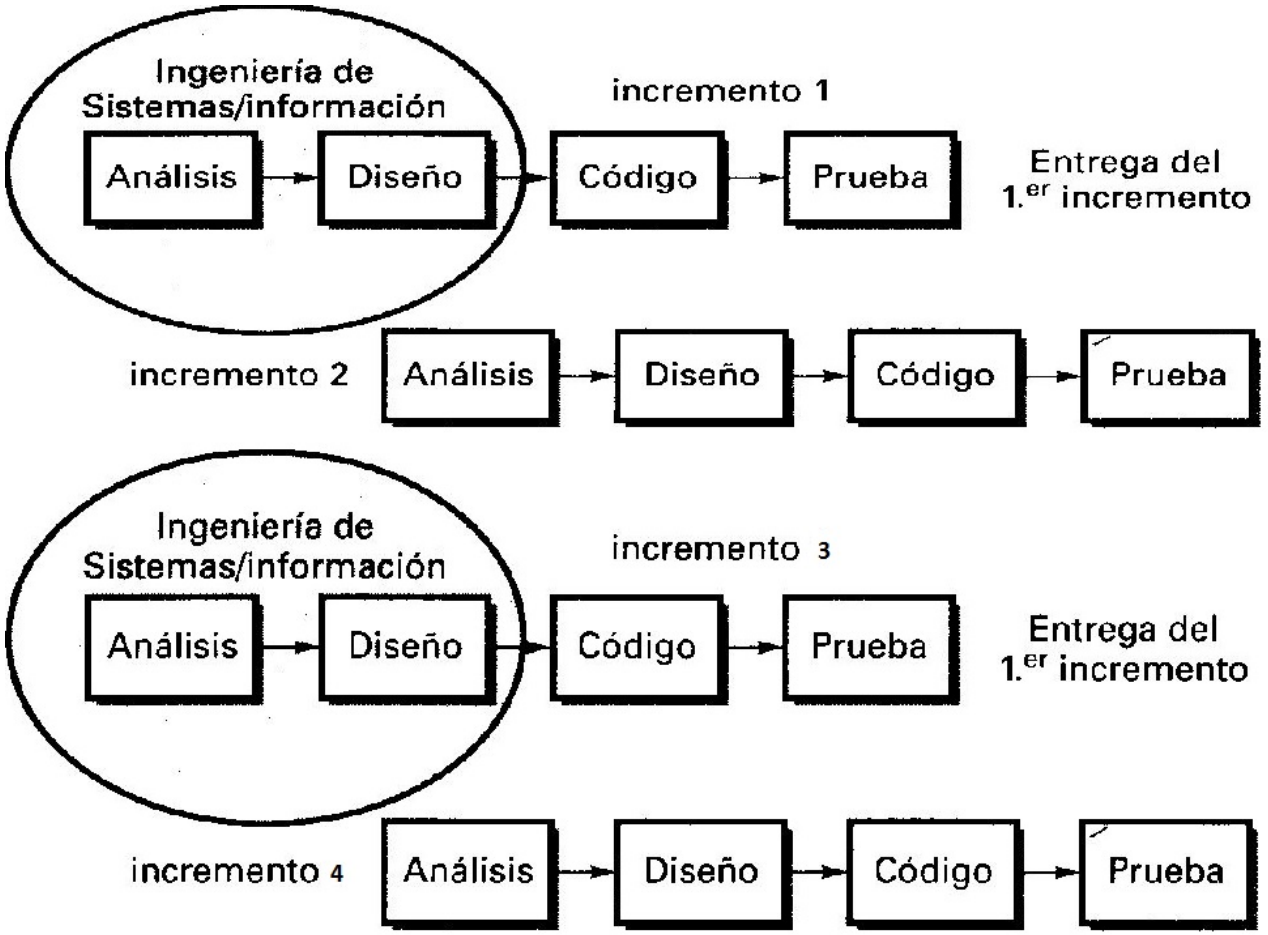
- **Modelo Incremental.**
- **Modelo Espiral.**

Modelos Incremental

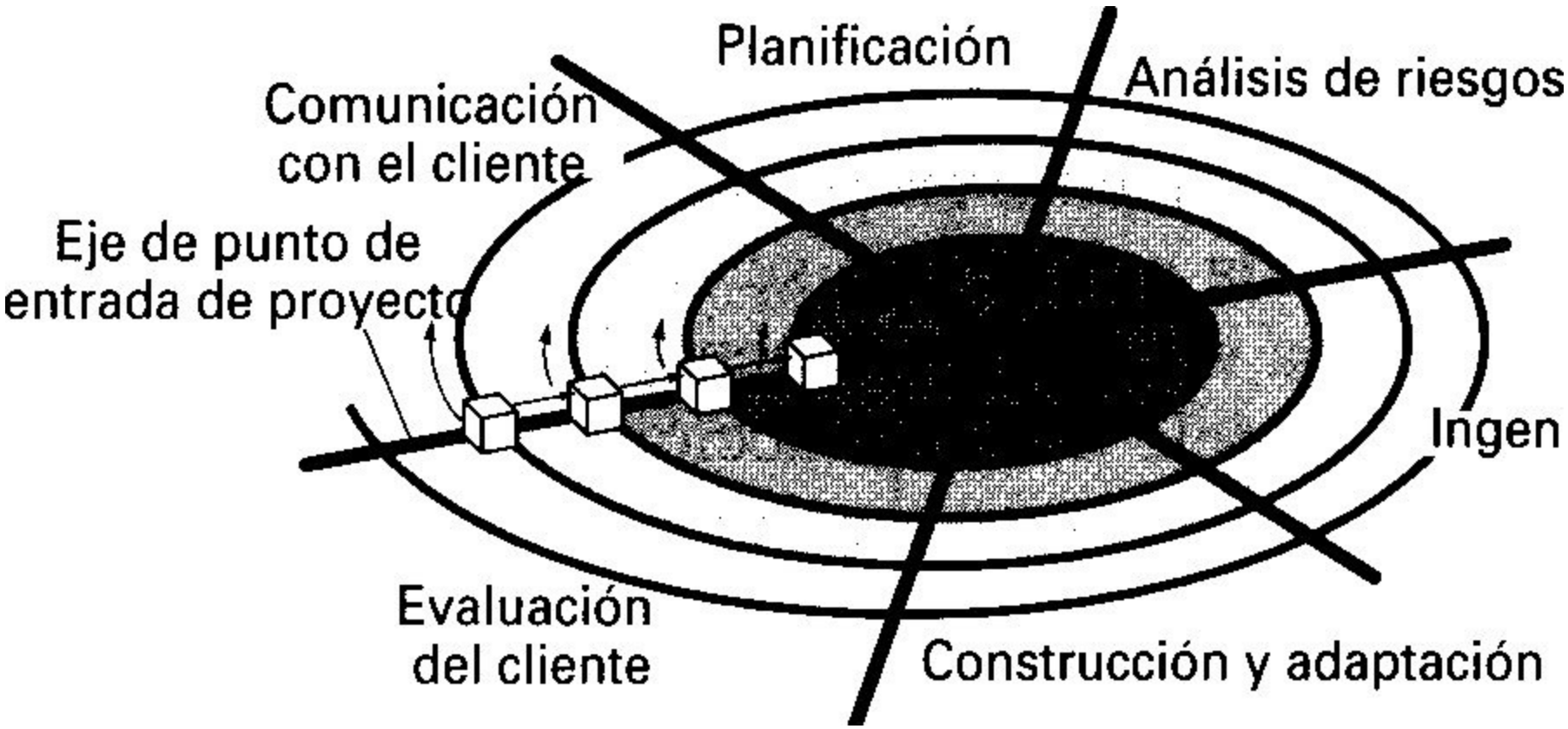
El *modelo incrernental combina elementos del modelo* lineal secuencial (aplicados repetidamente) con la filosofía interactiva de construcción de prototipos.

Modelos Incremental

A partir de la Etapa de Diseño se empieza con análisis y se van realizando cada entrega del incremento hasta llegar al 4ta entrega del incremento.



Modelo Espiral



Modelo Espiral

El modelo Espiral tiene seis regiones:

- **Comunicación con el cliente-** las tareas requeridas para establecer comunicación entre el desarrollador y el cliente.
- **Planificación-** las tareas requeridas para definir recursos, el tiempo y otra información relacionadas con el proyecto.
- **Análisis de riesgos-** las tareas requeridas para evaluar riesgos técnicos y de gestión.
- **Ingeniería**- las tareas requeridas para construir una o más representaciones de la aplicación.
- **Construcción y acción-** las tareas requeridas para construir, probar, instalar y proporcionar soporte al usuario (por ejemplo: documentación y práctica).
- **Evaluación del cliente-** las tareas requeridas para obtener la reacción del cliente según la evaluación de las representaciones del software creadas durante la etapa de ingeniería e implementada durante la etapa de instalación.

Características del Modelo en Espiral

- Cada una de las regiones están compuestas por un conjunto de tareas del trabajo que se adaptan a las características del proyecto que va a emprenderse.
- Para proyectos pequeños, el número de tareas de trabajo y su formalidad es bajo.
- Para proyectos mayores y más críticos cada región de tareas contiene
- tareas de trabajo que se definen para lograr un nivel más alto de formalidad.
- Cuando empieza este proceso evolutivo, el equipo de ingeniería del software gira alrededor de la espiral en la dirección de las agujas del reloj, comenzando por el centro. El primer circuito de la espiral puede producir el desarrollo de una especificación de productos; los pasos siguientes en la espiral se podrían utilizar para desarrollar un prototipo y progresivamente versiones.
- A diferencia del modelo de proceso clásico que termina cuando se entrega el software, el modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.

El Modelo en Espiral:

- Utiliza Prototipos
- Mantiene el enfoque sistemático de los pasos sugeridos por el ciclo de
- vida clásico, pero lo incorpora al marco de trabajo iterativo que refleja de forma más realista el mundo real.
- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto, y, si se aplica adecuadamente, debe reducir los riesgos antes de que se conviertan en problemáticos.

Modelo en Espiral WINWIN

- Aborda la comunicación con el cliente.
- Las mejores negociaciones se esfuerzan en obtener «victoria-victoria». Esto es, el cliente gana obteniendo el producto o sistema que satisface la mayor parte de sus necesidades y el desarrollador gana trabajando para conseguir presupuestos y lograr una fecha de entrega realista.

Modelo en Espiral WINWIN

El modelo en espiral WINWIN de Boehm define un conjunto de actividades de negociación al principio de cada paso alrededor de la espiral.

1. Identificación del sistema o subsistemas clave de los «directivos»
2. Determinación de las «condiciones de victoria» de los directivos.
3. Negociación de las condiciones de «victoria» de los directivos para reunirlas en un conjunto de condiciones «victoria-victoria» para todos los afectados (incluyendo el equipo del proyecto de software).
4. Valida las definiciones del producto y del proceso.
5. Definir el siguiente nivel del proceso y producto incluyendo peticiones
6. Valida las definiciones del producto y el proceso
7. Revisión, compromiso

Modelo en Espiral WINWIN

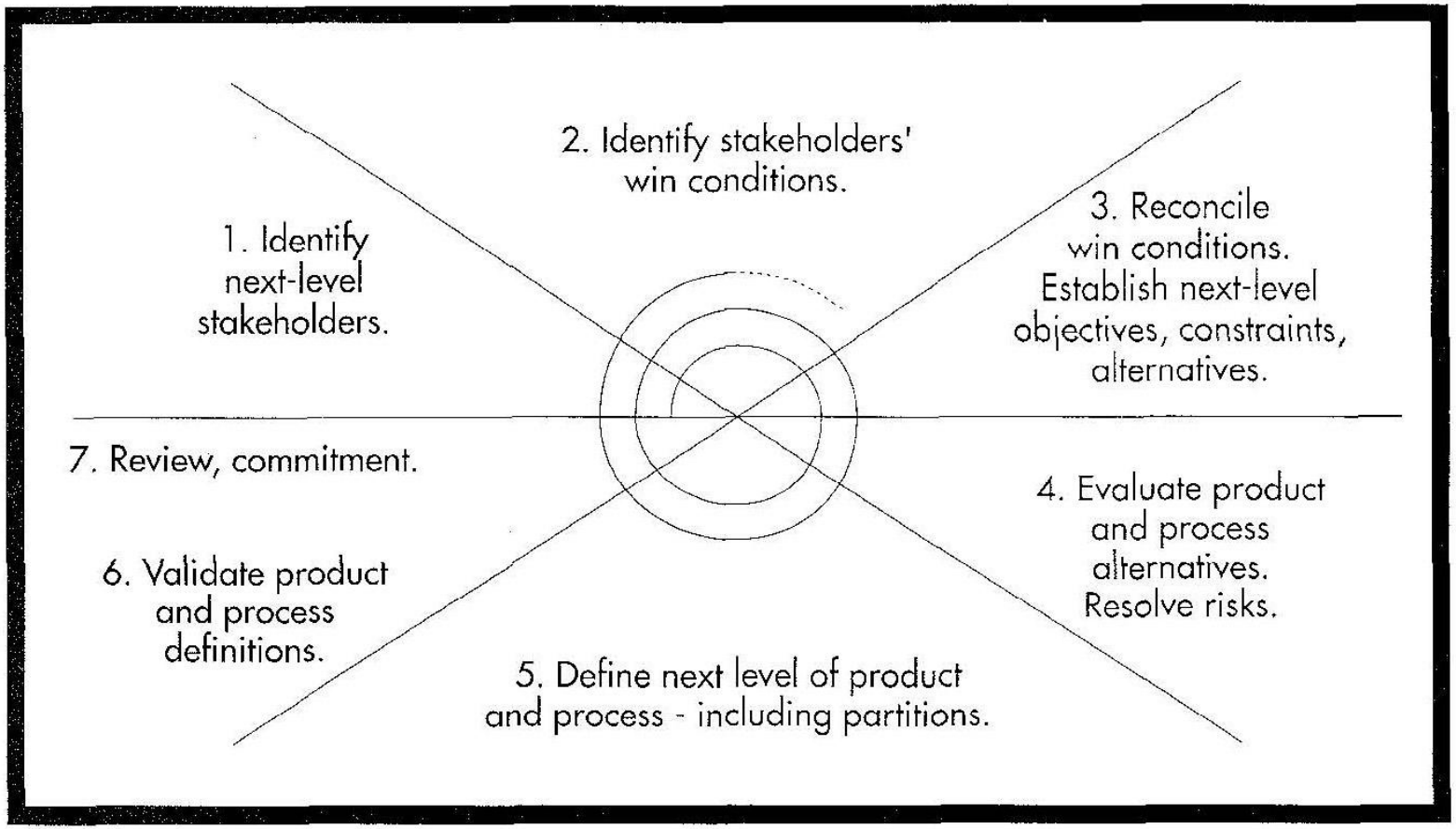


Figure 1. The Win-Win spiral model.

Modelo en Espiral WINWIN

- Primer punto de fijación: *objetivos del ciclo de vida (**OCV**),*
- Segundo punto de fijación: *arquitectura del ciclo de vida (**ACV**)*
- *Tercer punto de fijación: capacidad operativa inicial* (**COI**)

Modelo de Desarrollo Concurrente

La dimensión de componentes se afronta con dos actividades:

- **diseño**
- **realización**

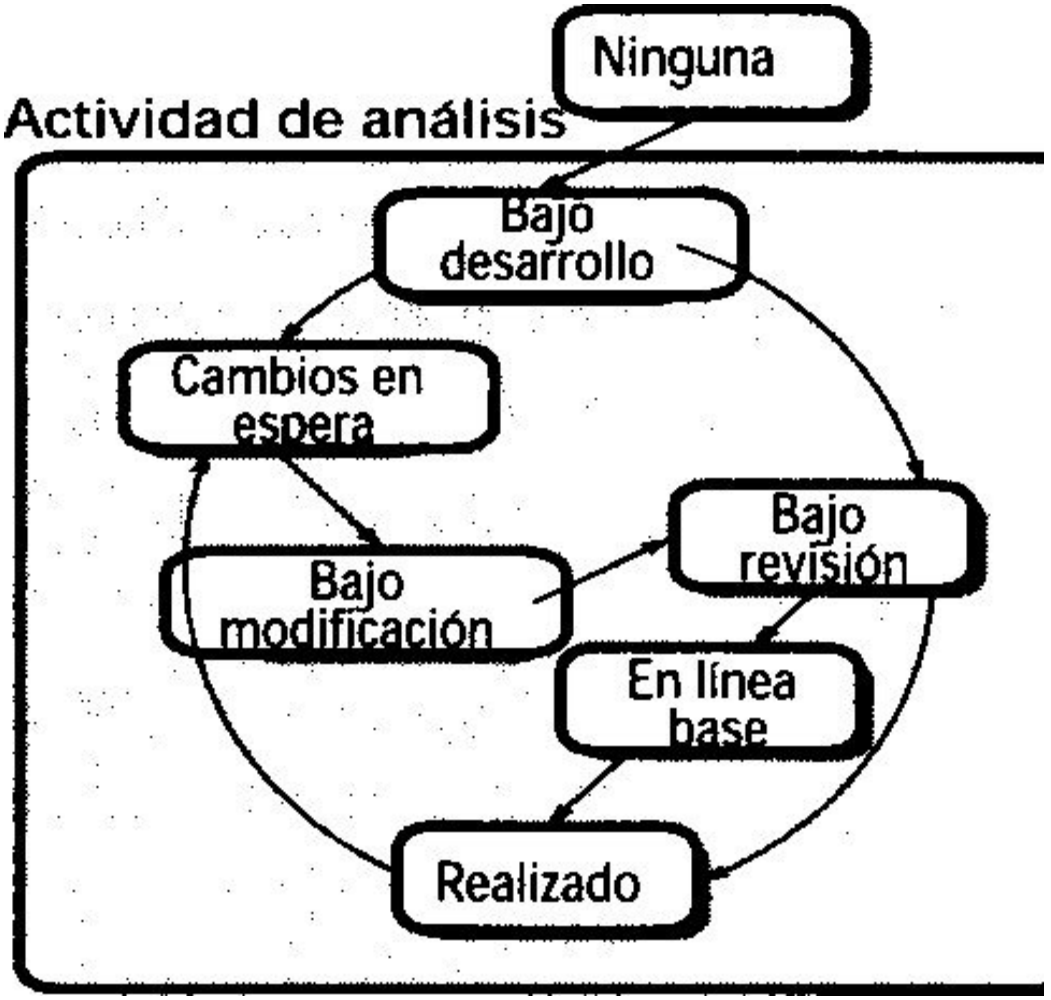
La concurrencia se logra de dos formas:

(1) **las actividades de sistemas y de componentes** ocurren simultáneamente y pueden modelarse con el enfoque orientado **a objetos descrito anteriormente.**

(2) una aplicación cliente/servidor típica se implementa con muchos componentes, cada uno de los cuales se pueden diseñar y realizar concurrentemente**.**

En realidad, el modelo de proceso concurrente es aplicable a todo tipo de desarrollo de software y proporciona una imagen exacta del estado actual de un proyecto.

Modelo de Desarrollo Concurrente



Desarrollo Basado en Componentes