Procesos de Software

El proceso del Software

- Un conjunto estructurado de actividades necesarias para desarrollar un sistema de software
- Muchos de los procesos de software son diferentes, pero todos implican:
  - Especificación - la definición de lo que el sistema debe hacer;
  - Diseño e implementación - la definición de la organización del sistema y la implementación del sistema;
  - Validación - la comprobación de que hace lo que quiere el cliente;
  - Evolución - el cambio del sistema en respuesta a las necesidades cambiantes de los clientes.
- Un modelo de proceso de software es una representación abstracta de un proceso. Se presenta una descripción de un proceso a partir de una perspectiva particular.

Descripciones de procesos de software

- Cuando describimos y discutimos los procesos, por lo general hablamos de las actividades en estos procesos, como la especificación de un modelo de datos, el diseño de una interfaz de usuario, etc, y el ordenamiento de estas actividades.

- Descripciones de proceso también pueden incluir:
  - Productos, que son los resultados de una actividad del proceso;
  - Roles, que reflejan las responsabilidades de las personas involucradas en el proceso;
  - Pre-y post-condiciones, que son declaraciones que son verdaderas antes y después de una actividad de proceso se ha promulgado o elaborado un producto.

Proceso dirigido por Plan y procesos ágiles.

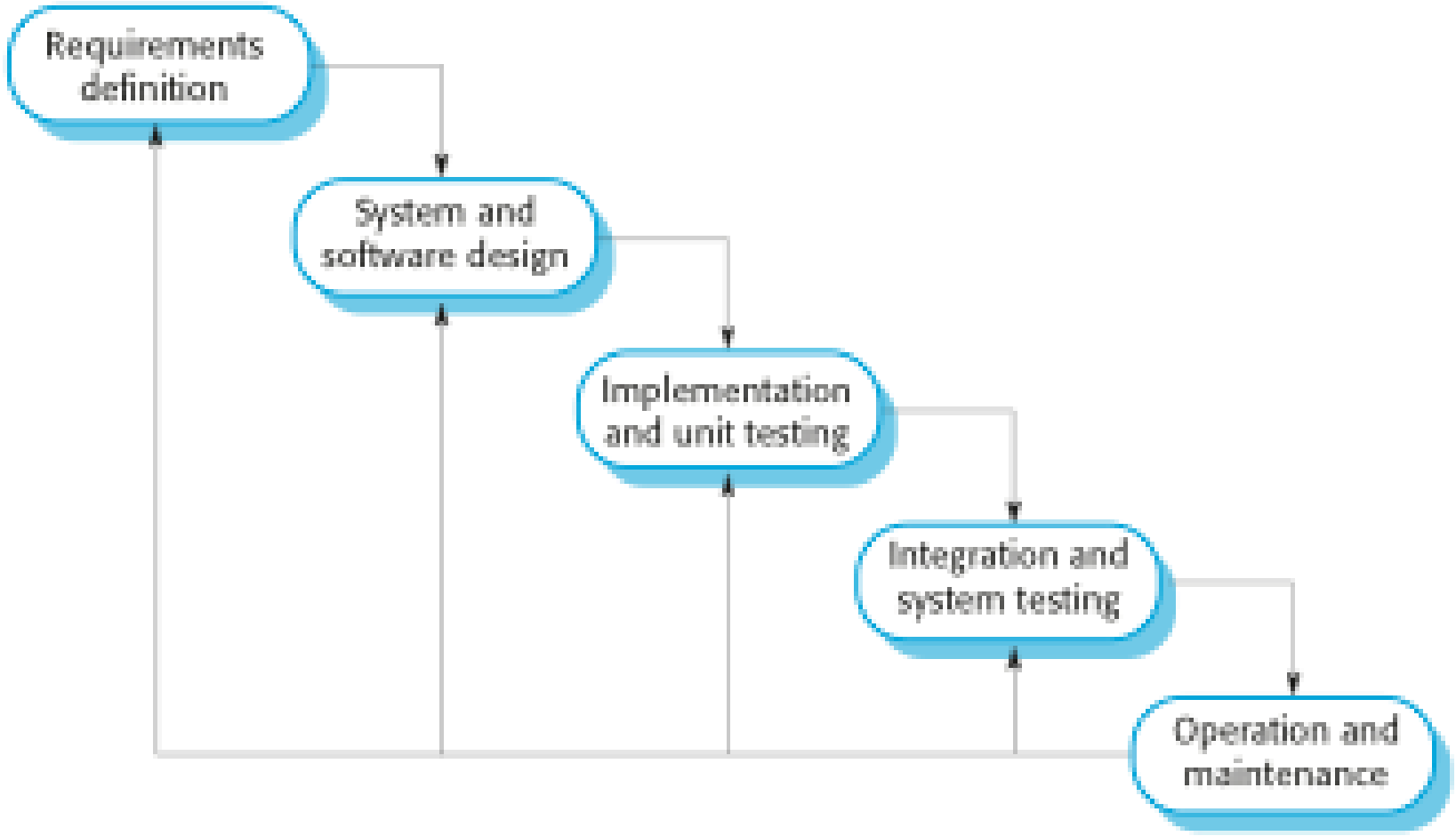
- En un desarrollo dirigido por plan todas las actividades del proceso se planifican con antelación y el progreso se mide en contra de este plan.
- En los procesos ágiles, la planificación es gradual y es más fácil para cambiar el proceso para reflejar los requisitos cambiantes de los clientes.
- En la práctica, la mayoría de los procesos prácticos incluyen elementos de ambos enfoques el dirigido por plan y el ágil.
- No hay procesos de software correctos o incorrectos.

Modelos de procesos de software

- El modelo de cascada
  - Modelo dirigido por Plan. Fases separadas y distintas de especificación y desarrollo.
- El desarrollo incremental
  - Especificación, desarrollo y validación se intercalan. Puede ser el dirigido por plan o ágil.
- Ingeniería de software orientado a reutilización
  - El sistema se ensambla a partir de componentes existentes Puede ser el dirigido por plan o ágil.

En la práctica, la mayoría de los grandes sistemas se desarrollan mediante un proceso que incorpora elementos de todos estos modelos.

Modelo de Cascada



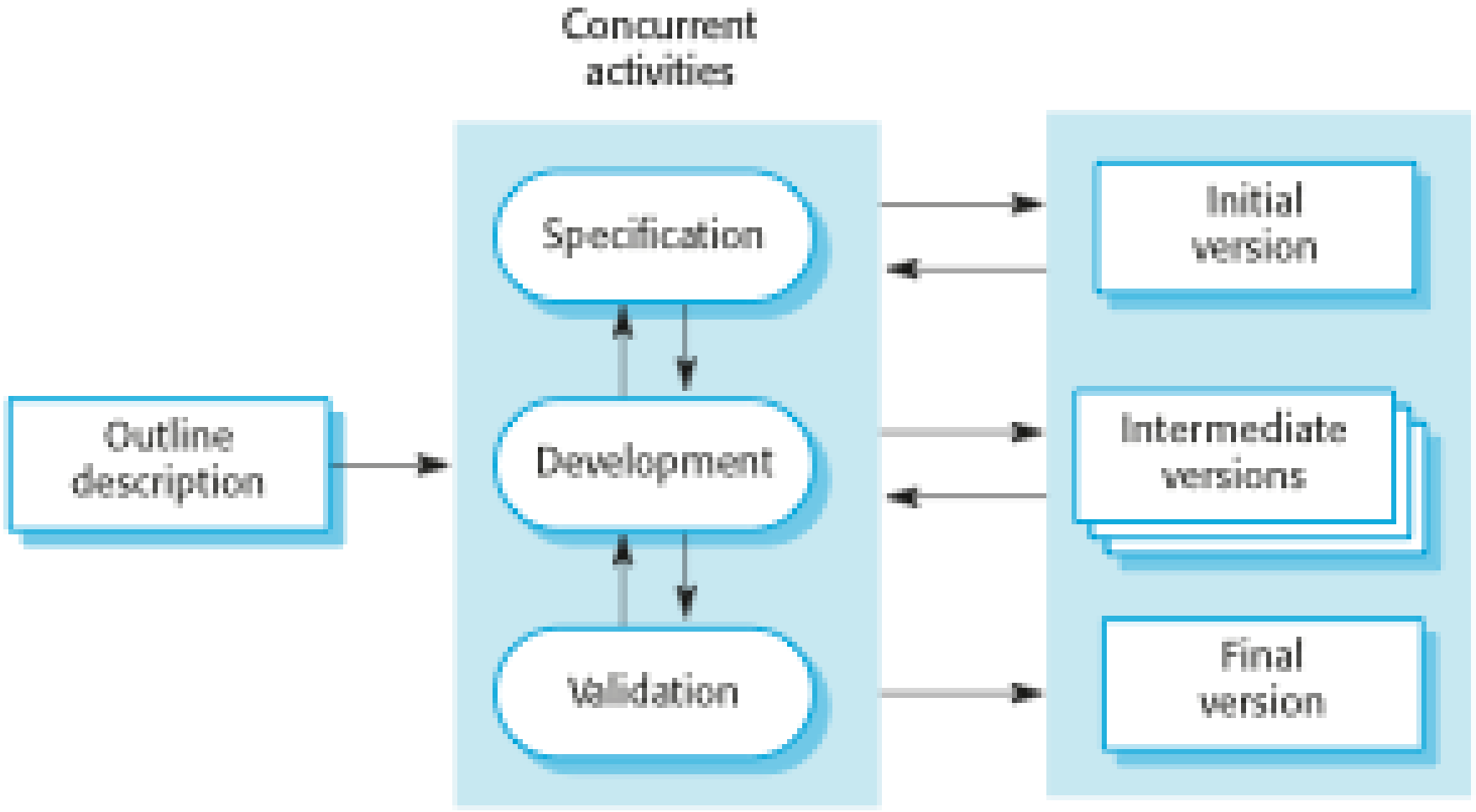
Fases del Modelo de Cascada

- Hay fases identificadas por separado en el modelo de cascada:
  - El análisis de requerimientos y su definición
  - El diseño del sistema y del software
  - Implementación y prueba de unidades
  - Integración y pruebas del sistema
  - Operación y mantenimiento
- El principal inconveniente del modelo de la cascada es la dificultad de acomodar el cambio después de que está en marcha el proceso. En principio, una fase tiene que ser completa antes de pasar a la siguiente fase.

Problemas del Modelo de Cascada

- Inflexible, la división del proyecto en fases estructuradas hace difícil responder a las necesidades cambiantes de los clientes. Por lo tanto, este modelo sólo es apropiado cuando los requisitos son bien entendidos y los cambios serán bastante limitados durante el proceso de diseño.
- POCOS SISTEMAS tienen requisitos estables.
- El modelo de cascada se utiliza sobre todo para los grandes proyectos de ingeniería de sistemas especialmente si un sistema se desarrolla en varios lugares. En estas circunstancias el modelo de cascada ayuda a coordinar el trabajo

El desarrollo incremental



Beneficios de desarrollo incremental

- El costo de atender las necesidades cambiantes de los clientes se reduce.
  - La cantidad de análisis y la documentación que tiene hacerse de nuevo es mucho menor que la que se requiere con el modelo de cascada.
- Es más fácil obtener retroalimentación de los clientes en el trabajo de desarrollo.
  - Los clientes pueden hacer comentarios sobre el avance del desarrollo del software y probar lo que se ha implementado.
- Más rápida entrega y despliegue de software de utilidad para el cliente.
  - Los clientes pueden usar y obtener valor a partir del software mas rápidamente de lo que es posible con un proceso de cascada.

Problemas de desarrollo incremental

- El proceso no es visible.
  - Los gerentes necesitan entregas regulares para medir el progreso. No es rentable producir documentos que reflejen todas las versiones del sistema.
- Estructura del sistema tiende a degradarse a medida que se añaden nuevos incrementos.
  - Se gasta menos tiempo y dinero en la refactorización para mejorar el software, lo que tiende a corromper su estructura. La incorporación de nuevos cambios se vuelve cada vez más difícil y costoso.

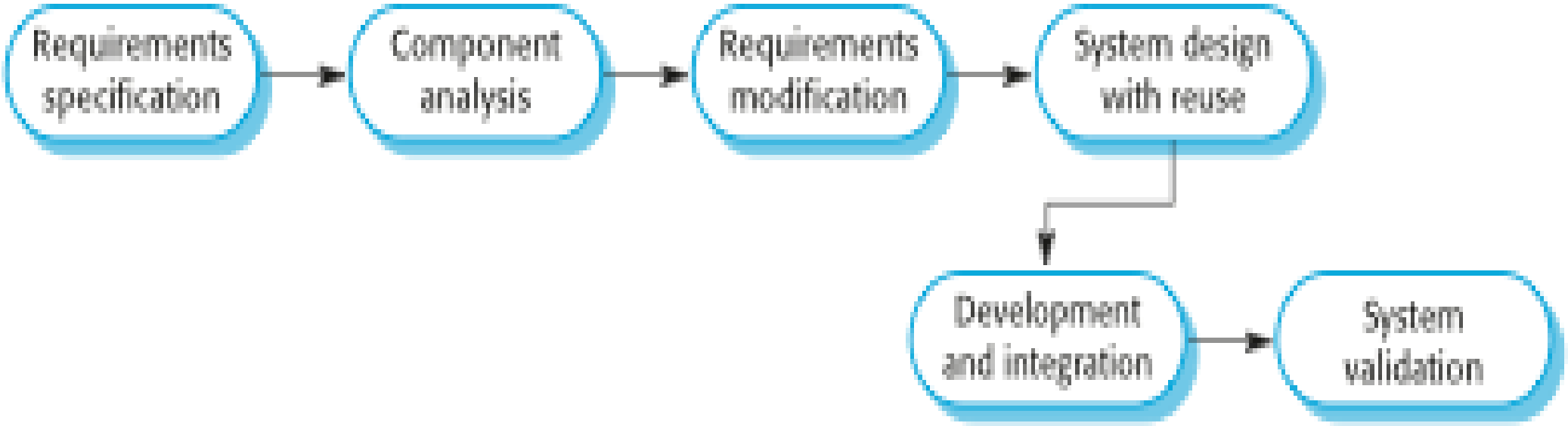
Ingeniería de software orientado a Reutilización

- Se basa en la reutilización sistemática de código, los sistemas se integran a partir de componentes o sistemas existentes.

- Etapas del proceso
  - Análisis de requerimiento;
  - Análisis de los componentes;
  - Modificación de requerimientos;
  - Configuración del sistema con la reutilización;
  - Desarrollo e integración.

- La reutilización es ahora el enfoque estándar para la construcción de muchos tipos de sistemas.

Ingeniería de software orientado a Reutilización



Tipos de componentes de software

- Los servicios Web que se desarrollan de acuerdo a los estándares de servicio y que están disponibles para la invocación remota.
- Colecciones de objetos que se desarrollan como un paquete para ser integrado con un marco de componentes tales como. NET o J2EE.
- Sistemas autónomos de software (COTS) que están configurados para su uso en un entorno particular.

Actividades de proceso

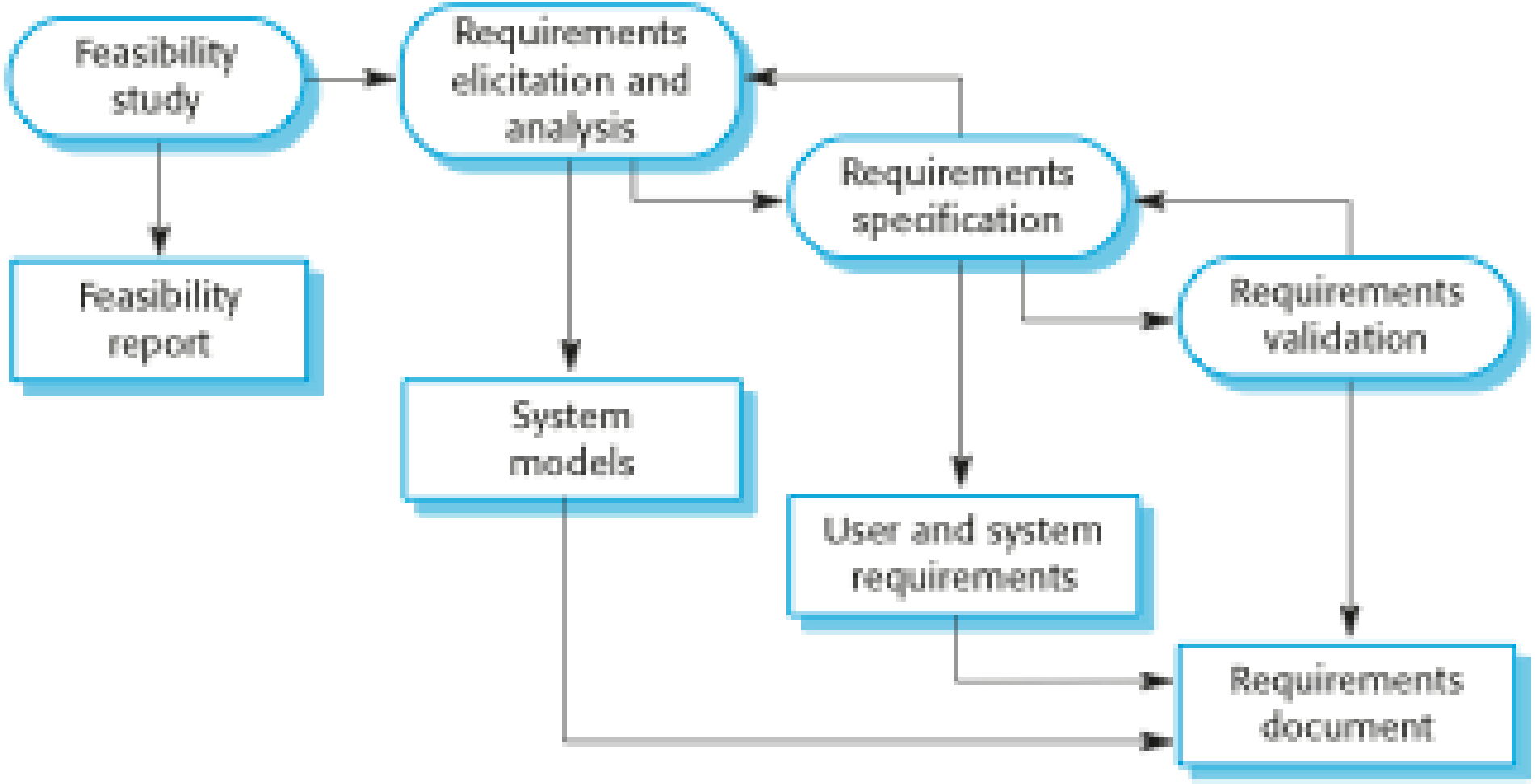
- Procesos de software reales son secuencias intercalados de actividades técnicas, de colaboración y de gestión con el objetivo general de la especificación, diseño, implementación y prueba de un sistema de software.

- Las cuatro actividades básicas del proceso son: especificación, desarrollo, validación y evolución y están organizados de manera diferente según el proceso de desarrollo. En el modelo de cascada, se organizan en secuencia, mientras que en el desarrollo incremental son intercalados.

Especificaciones de Software

- El proceso de establecer qué servicios son necesarios y las limitaciones de funcionamiento y desarrollo del sistema.
- Ingeniería de Requisitos o Requerimientos
  - Estudio de factibilidad
    - Es técnicamente y financieramente factible para construir el sistema?
  - Requerimientos, obtención y análisis
    - Que requieren los diferentes actores del sistema o esperan del sistema ?
  - Especificación de Requerimientos
    - Definición de los requisitos en detalle
  - Validación de Requerimientos
    - Comprobación de la validez de los requisitos

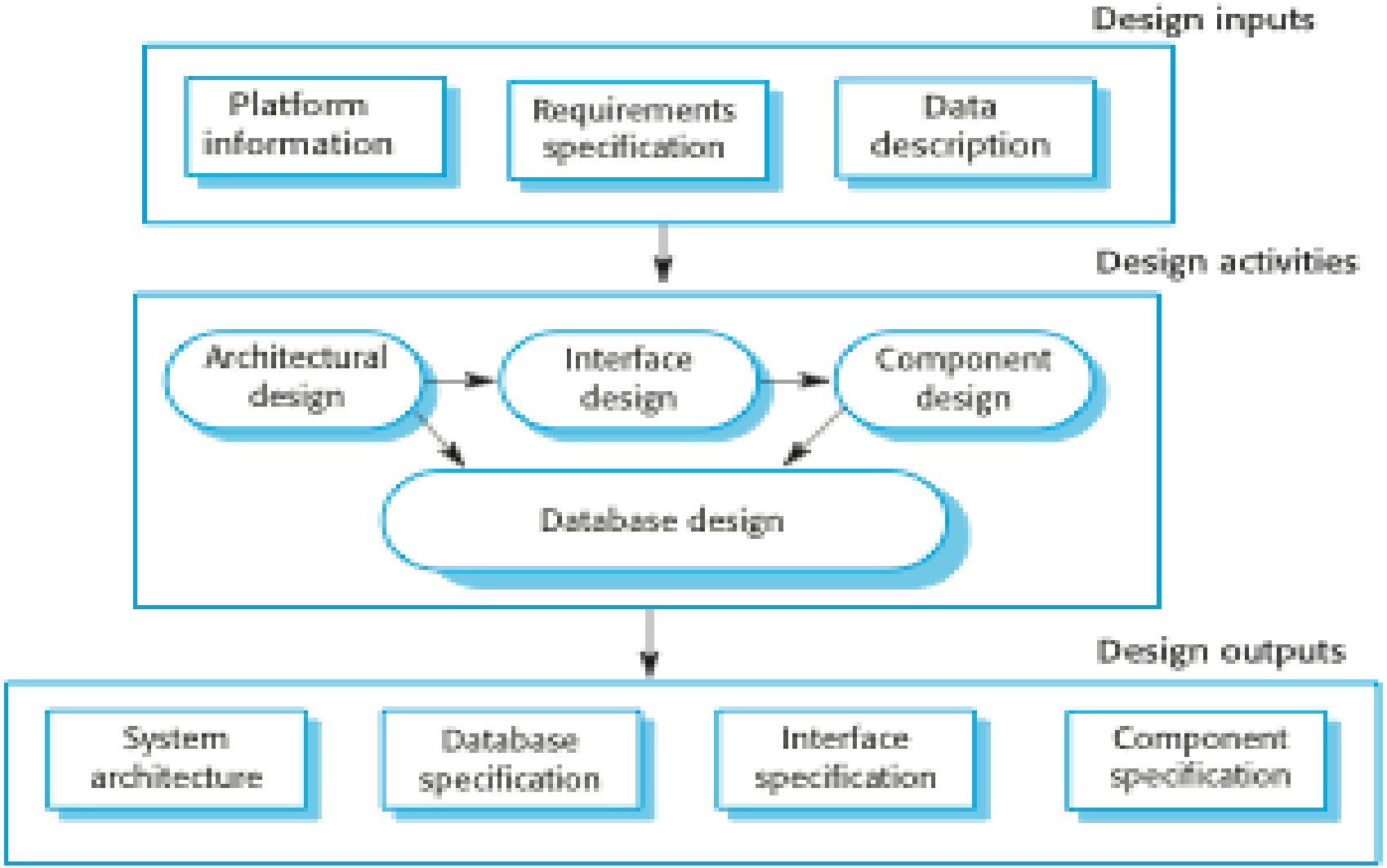
El proceso de ingeniería de requerimientos



El diseño de software y la implementación

- Proceso de conversión entre la especificación del sistema en un sistema ejecutable.
- El diseño de software
  - Diseñar una estructura de software que da cuenta de la especificación;
- implementación
  - Traducir esta estructura en un programa ejecutable;
- Las actividades de diseño e implementación están estrechamente relacionados y pueden ser intercaladas.

Un modelo general del proceso de diseño



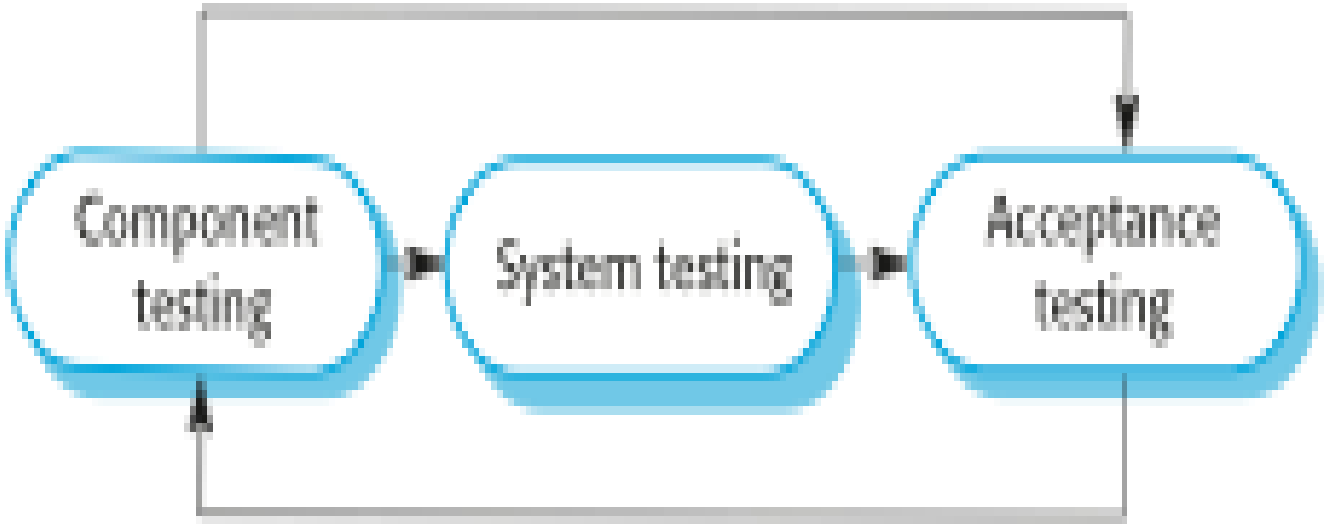
Actividades de Diseño

- **Diseño arquitectónico**, donde se identifica la estructura general del sistema, los componentes principales (a veces llamados subsistemas o módulos), sus relaciones y la forma en que se distribuyen.
- **Diseño de la interfaz**, donde se definen las interfaces entre los componentes del sistema.
- **Diseño de componentes**, donde se toma cada componente del sistema y el diseño de cómo se va a operar.
- **Diseño de base de datos**, donde se diseña la estructura de datos del sistema y de cómo éstos han de estar representados en una base de datos.

Validación del Software

- **Verificación y validación** (V & V) está destinado a demostrar que un sistema cumple con su especificación y cumple con los requisitos del cliente.
- Involucra procesos de control, revisión y prueba del sistema.
- Las pruebas del sistema implica ejecutar el sistema con casos de prueba que se derivan de la especificación utilizando datos reales.
- La prueba es la actividad de V & V más utilizada.

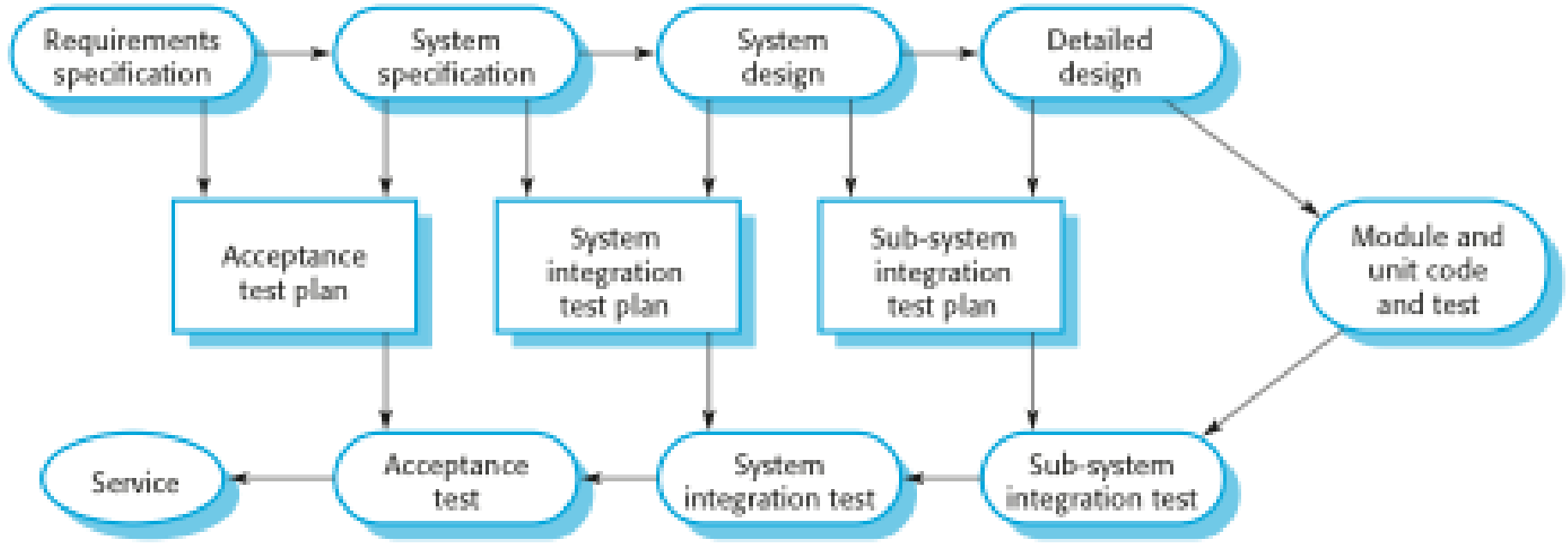
Etapas de la prueba



Etapas de prueba

- Pruebas de Desarrollo o componente.
  - Los componentes individuales se prueban de forma independiente;
  - Los componentes pueden ser funciones, objetos o agrupaciones coherentes de estas entidades.
- Las pruebas del sistema
  - Pruebas del sistema como un todo. El ensayo de las propiedades emergentes es particularmente importante.
- Las pruebas de aceptación
  - Las pruebas realizadas por el cliente para verificar que el sistema cumple con sus necesidades.

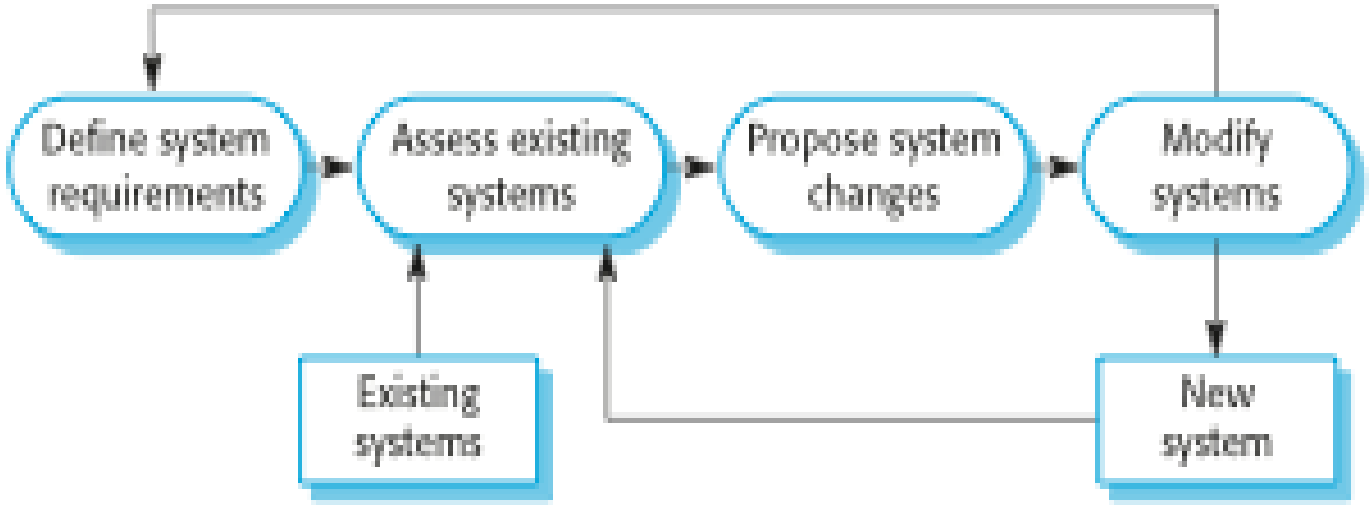
Fases de prueba en un proceso de software



La evolución del software

- El software es inherentemente flexible y puede cambiar.
- Las circunstancias cambiantes de negocios hacen que el software que soporta la empresa también deba evolucionar y cambiar.
- Si bien se habla de desarrollo y evolución (mantenimiento) como etapas diferentes, la diferencia es cada vez más irrelevante, cada vez son menos los sistemas completamente nuevos.

Evolución del Sistema



El problema del cambio

- El cambio es inevitable en todos los grandes proyectos de software.
  - Cambios en el negocio conducen a requisitos nuevos y modificaciones del sistema
  - Las nuevas tecnologías abren nuevas posibilidades de mejorar las implementaciones
  - Cambio de plataformas requieren cambios en las aplicaciones
- Los costos del cambio incluyen tanto la reelaboración (por ejemplo, requisitos de re-analizar), como los costos de implementación de nuevas funcionalidades

La reducción de los costos de rehacer

- **Evitar el Cambio**, donde el proceso de software incluye actividades que pueden anticipar posibles cambios para evitar repetir el trabajo.
  - Por ejemplo, un prototipo del sistema puede ser desarrollado para mostrar algunas de las características clave del sistema para los clientes.

- **Tolerancia al Cambio**, en el que el proceso está diseñado de modo que los cambios pueden afrontarse con un costo relativamente bajo.
  - Esto implica alguna forma de desarrollo incremental. Los cambios propuestos pueden implementarse en incrementos que aún no se han desarrollado. Sólo un único incremento (una pequeña parte del sistema) debe ser alterado para incorporar el cambio.

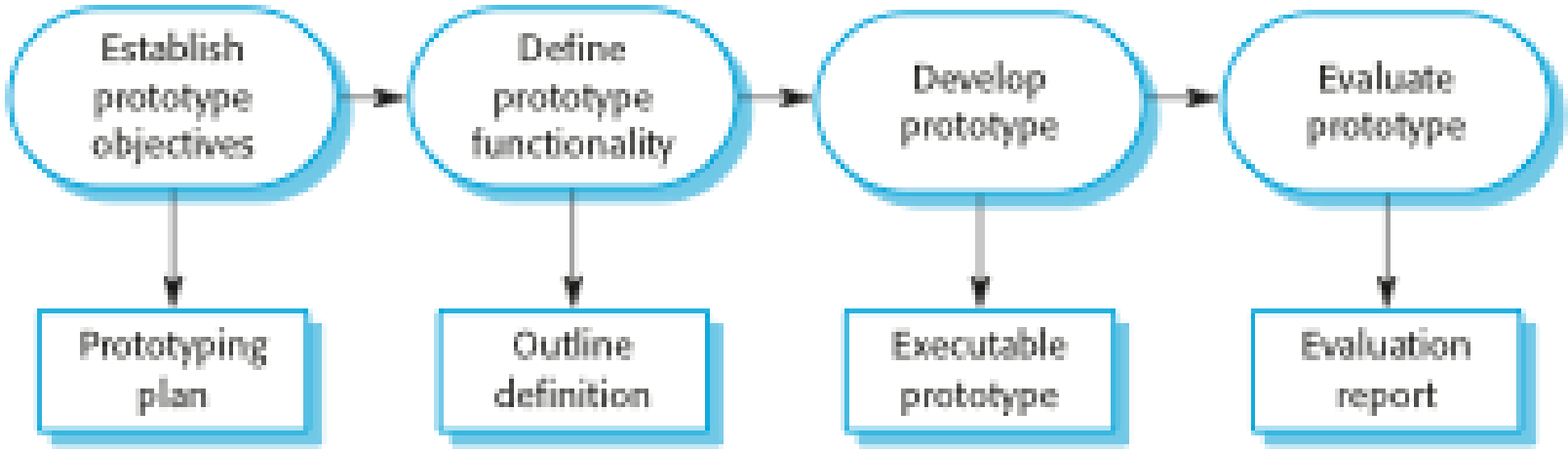
Software Prototipado

- Un prototipo es una versión inicial de un sistema que se utiliza para demostrar conceptos y probar opciones de diseño.
- Un prototipo se puede utilizar en:
  - El proceso de ingeniería de requerimientos para ayudar con la obtención de requisitos y validación;
  - En los procesos de diseño para explorar opciones y desarrollar un diseño de interfaz de usuario;
  - En el proceso de pruebas.

Beneficios del prototipado

- Mejora de la usabilidad del sistema.
- Una aproximación más exacta a las necesidades reales de los usuarios.
- Mejora de la calidad del diseño.
- Mejora de la capacidad de mantenimiento.
- Reduce del esfuerzo de desarrollo.

El proceso de desarrollo de prototipos



Desarrollo de prototipos

- Prototipo debe centrarse en las áreas del producto que no se conocen bien

- La comprobación de errores y recuperación pueden no estar incluidos en el prototipo

- Centrarse en los requisitos funcionales y no en los no funcionales tales como la fiabilidad y la seguridad

Prototipos desechables

- Los prototipos deben desecharse después de desarrollo ya que no son una buena base para un sistema de producción:
  - Puede ser imposible para ajustar el sistema para cumplir con los requisitos no funcionales;
  - Los prototipos son normalmente indocumentados;
  - La estructura del prototipo se suele degradarse a través de un cambio rápido;
  - El prototipo probablemente no va a cumplir con los estándares de calidad normal de la organización.

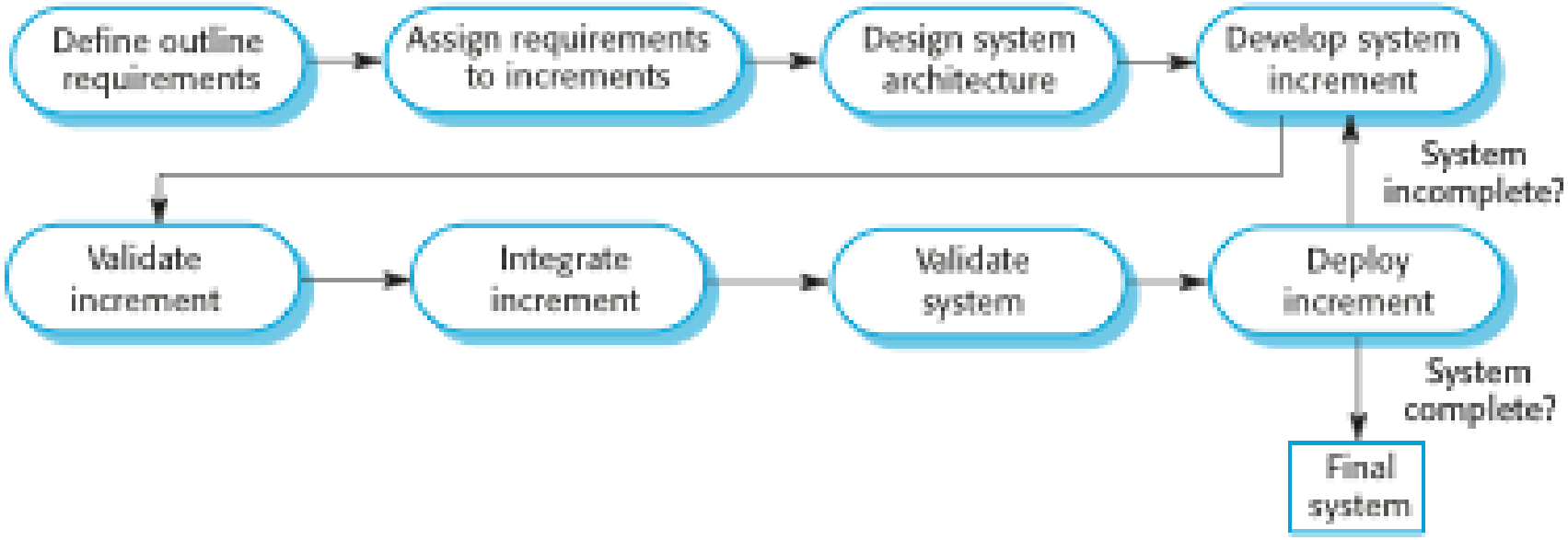
Entrega incremental

- En lugar de entregar el sistema en una sola vez, el desarrollo y la entrega se desglosan en incrementos, con cada incremento se entrega de parte de la funcionalidad requerida.
- Requisitos de usuario se priorizan y se incluyen los requisitos de más alta prioridad en incrementos tempranos.
- Una vez que se inicia el desarrollo de un incremento, los requisitos están congelados, aunque los requisitos para incrementos posteriores pueden seguir evolucionando.

El desarrollo incremental y la entrega

- El desarrollo incremental
  - Desarrollar el sistema en incrementos y evaluar cada incremento antes de continuar con el desarrollo del siguiente incremento;
  - Aproximación normal utilizado en los métodos ágiles;
  - La evaluación puede ser hecha por el usuario / cliente.
- Entrega incremental
  - Implementar un incremento para su uso por los usuarios finales;
  - Una evaluación más realista sobre el uso práctico del software;
  - Difícil de implementar para el caso de sustitución de sistemas existentes, los incrementos tienen menos funcionalidad que el sistema está reemplazando.

Entrega incremental



Ventajas entrega incremental

- El sistema está disponible antes, con cada entrega se agrega funcionalidad al sistema.
- Las primeras entregas actúan como un prototipo, esto es útil para el esclarecimiento de requisitos para incrementos posteriores.
- Menor riesgo de fracaso del proyecto en general.
- Los servicios de mayor prioridad se entregan antes y tienden a recibir la mayor cantidad de pruebas.

Problemas de entrega incremental

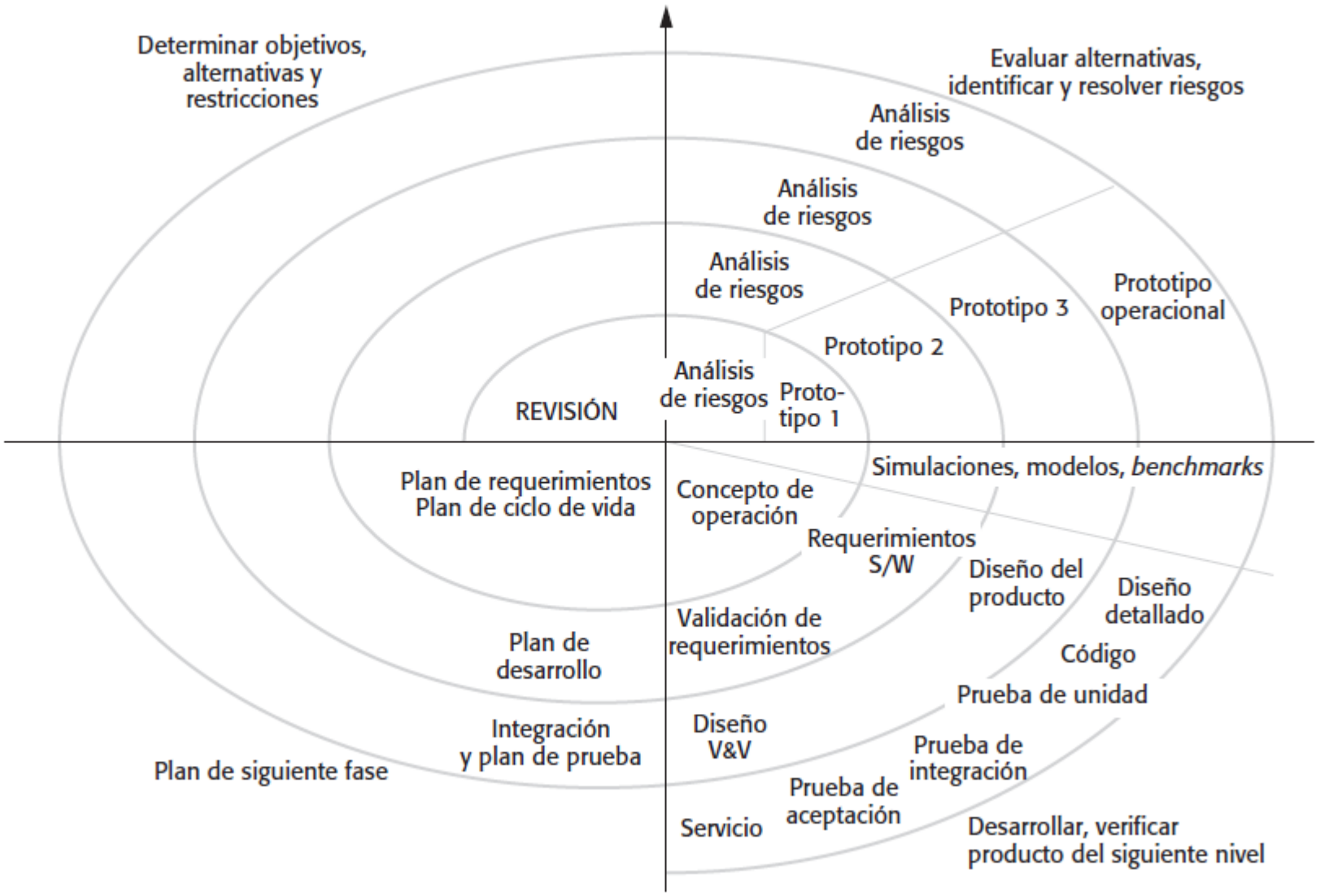
- La mayoría de los sistemas requieren un conjunto de servicios básicos que se utilizan por diferentes partes del sistema.
  - Dado que los requisitos no están definidos en detalle hasta que se llegue al incremento de su ejecución, puede ser difícil identificar las funcionalidades comunes que son necesarias para todos los incrementos.

- La esencia de los procesos iterativos es que la especificación se desarrolla en conjunto con el software.
  - Sin embargo, esto entra en conflicto con el modelo de adquisición de muchas organizaciones, donde la especificación completa del sistema es parte del contrato de desarrollo del sistema.

Modelo en Espiral de Boehm

- Proceso se representa como una espiral en lugar de como una secuencia de actividades con retroceso.
- Cada bucle en la espiral representa una fase en el proceso.
- No hay fases fijas tales como las especificaciones o el diseño - bucles en la espiral se eligen en función de lo que se requiere.
- Los riesgos son evaluados de forma explícita y se resuelven durante todo el proceso.

Modelo en espiral



Sectores del modelo en espiral

- Establecimiento de objetivos
  - Se identifican los objetivos específicos para la fase.
- La evaluación y la reducción de riesgos
  - Los riesgos son evaluados y se ponen en marcha actividades para reducir los riesgos clave.
- Desarrollo y validación
  - Se elige un modelo de desarrollo para el sistema que puede ser cualquiera de los modelos genéricos.
- Planificación
  - El proyecto es revisado y se continua con la siguiente fase de la espiral.

Actividades a realizar en grupo:

- Cleanroom software engineering.
- Capability Maturity Model for Software (CMM)