

# *Reutilización del Software*

---

- \* Construcción de software en base a componentes reutilizables.

# ***Reutilización***

---

*Software*

# *Tipos de Componentes Reutilizables*

---

- \* Reutilización de Sistemas de Aplicación
- \* Reutilización de Subsistemas
- \* Reutilización de Módulos o objetos
- \* Reutilización de Función

# *Practica de la reutilización*

---

## \* Reutilización de un Sistema de Aplicación

- \* Ampliamente difundida. Es una práctica común para desarrolladores de sistemas (p.ej. Microsoft) hacer sus productos disponibles para diferentes plataformas

## \* Reutilización de Subsistemas y módulos

- \* Practicado informalmente por ingenieros individuales que reutilizan trabajos previos.

## \* Reutilización de Funciones

- \* Es común en algunos dominios de aplicaciones (Por ej. Ingeniería) en donde han sido establecidas bibliotecas de dominio específico de funciones reusables. Reutilizar es la razón principal, del porque los lenguajes como FORTRAN son aun empleados

# *Aspectos de reutilización*

---

- \* Desarrollo de Software con reutilización
  - \* El desarrollo de software dada una base de componentes reusables
- \* Desarrollo de Software para reutilización
  - \* Diseño de componentes de software genérico para reutilización

# *Desarrollo de Software con reutilización*

---

- \* Se intenta maximizar el empleo de componentes existentes
- \* Estos componentes pueden ser adaptados en nuevas aplicaciones
- \* Pocos componentes necesitan ser especificados, diseñados y codificados
- \* Los costos de desarrollo deben por tanto ser reducidos

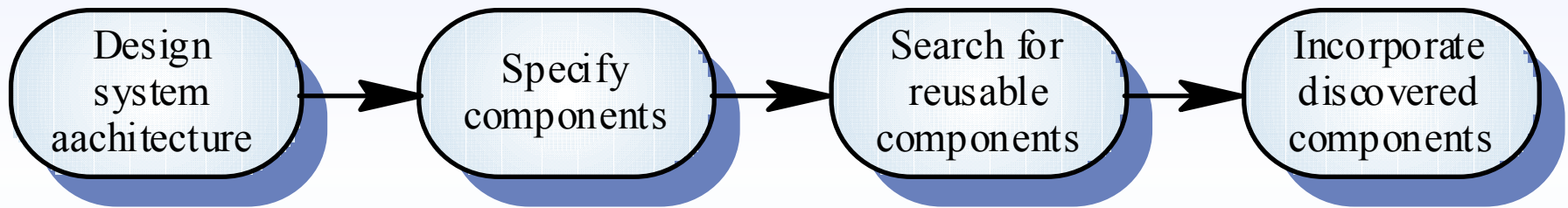
# *Ventajas adicionales*

---

- \* La confiabilidad de un Sistema se ve incrementada
- \* El riesgo es reducido
- \* El empleo efectivo puede ser hecho por especialistas
- \* Los estándares organizacionales pueden estar embebidos dentro de componentes reutilizables
- \* El tiempo de desarrollo de Software puede reducirse

# *Proceso de desarrollo con reutilización*

---





# *Requerimientos para la Reutilización*

---

- \* Debe ser posible encontrar componentes para reutilización apropiados en una base de datos de componentes
- \* Los componentes a reutilizar deben ser fáciles de comprender, para poder encontrar uno de acuerdo a nuestras necesidades, y fiables
- \* Los componentes pueden tener asociados documentación, discutiendo COMO los componentes pueden ser reutilizados y los costos potenciales de su uso

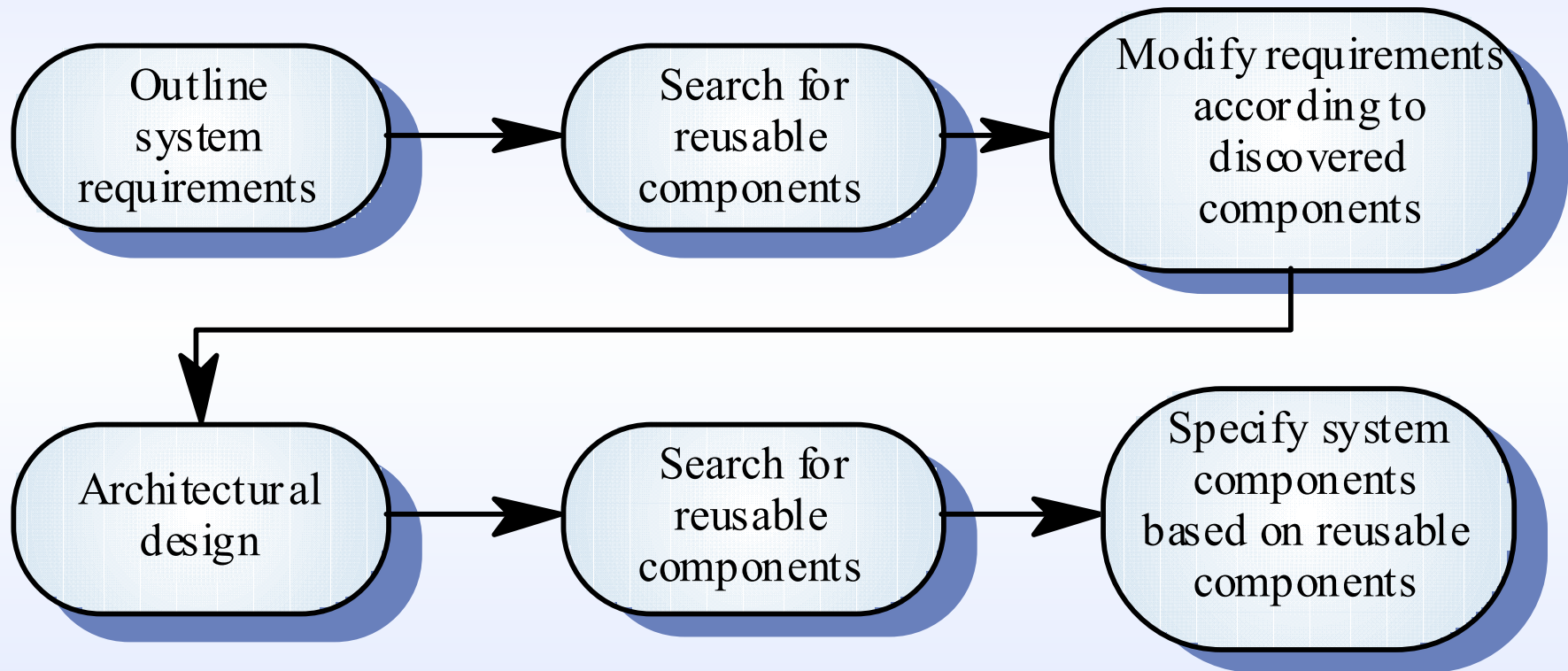
# *Desarrollo manejando reutilización*

---

- \* En lugar de considerar la reutilización después de que el software ha sido especificado, la especificación debe tomar en cuenta la existencia de componentes reutilizables
- \* Este enfoque es muy común en el diseño de sistemas electrónicos, eléctricos y mecánicos.
- \* Si se adopta para software, debe incrementar significativamente la proporción de componentes reutilizados

# *Desarrollo manejando reutilización*

---



# *Problemas con la reutilización*

---

- \* Es difícil cuantificar los costos y los beneficios del desarrollo con reutilización
- \* Las herramientas CASE no soportan el desarrollo con reutilización. Ellas no pueden ser integradas con sistemas con una biblioteca de componentes
- \* Algunos Ingenieros de software prefieren reescribir, en lugar de emplear componentes reutilizables
- \* Las técnicas actuales para la clasificación de componentes, catalogarlos y recuperarlos, son inmaduras. El costo de encontrar componentes adecuados es alto.

# *Desarrollo de Software para reutilización*

---

- \* Los componentes de Software no son automáticamente reutilizables. Ellos deben ser modificados para emplearlos en un amplio rango de aplicaciones.
- \* El desarrollo de Software para reutilización es un proceso de desarrollo, en el cual, se toman componentes y documentos existentes y se reutilizan para un amplio rango de aplicaciones.

# *Desarrollo para reutilización*

---

- \* El costo de desarrollo de componentes reutilizables es más alto, que el costo de especificaciones equivalentes. Debe pensarse en el costo en reutilizabilidad a nivel organizacional en vez de a nivel del costo de un proyecto
- \* Los componentes genéricos pueden ser menos eficientes en espacio y pueden tener un tiempo de ejecución más grande que sus especificaciones equivalentes

# *Mejora de la reutilización*

---

## **\* Generalización del Nombre**

- \* Los nombres en un componente pueden ser modificados, es decir, no existe una reflexión directa de una entidad de aplicación específica.

## **\* Generalización de la Operación**

- \* Las operaciones pueden ser adicionadas para proporcionar funcionalidad extra y las operaciones específicas pueden ser eliminadas de una aplicación.

## **\* Generalización de Excepciones**

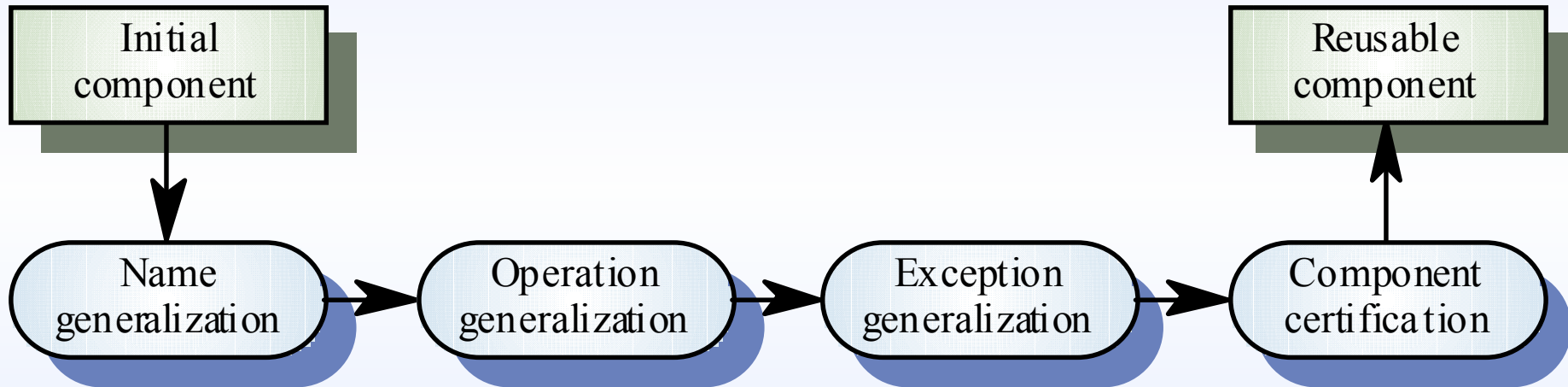
- \* Las excepciones de una aplicación específica son eliminadas y el manejo de excepciones se añade para incrementar la robustez de los componentes.

## **\* Certificación del Componente**

- \* El componente es certificado como reutilizable.

# *Proceso de mejora de la reutilización*

---





# *Ejemplo de componente reutilizable*

---

- \* Una lista ligada de elementos, donde cada elemento mantiene un apuntador al siguiente en la lista.
- \* Las operaciones de las listas ligadas son usualmente independientes del tipo de elemento en la lista

# *Adaptación de componentes*

---

- \* Una funcionalidad extra puede ser añadida a un componente. Cuando se añade, el nuevo componente estará disponible para su reutilización.
- \* La funcionalidad innecesaria debe ser removida de un componente para mejorar o reducir su desempeño o reducir sus requerimientos de espacio.
- \* La implementación de algunas operaciones de componentes pueden ser modificadas. Esto sugiere que la decisión de generalización original puede ser incorrecta.

# Resumen

---

- \* La reutilización de Software involucra la utilización de componentes desarrollados en alguna aplicación para otra aplicación diferente.
- \* La reutilización sistemática puede reducir costos, los riesgos y mejora la factibilidad del software.
- \* El desarrollo con reutilización puede basarse en una biblioteca de componentes reutilizables.
- \* Los componentes deben generalizarse para su reutilización