

# Ingeniería de Software II

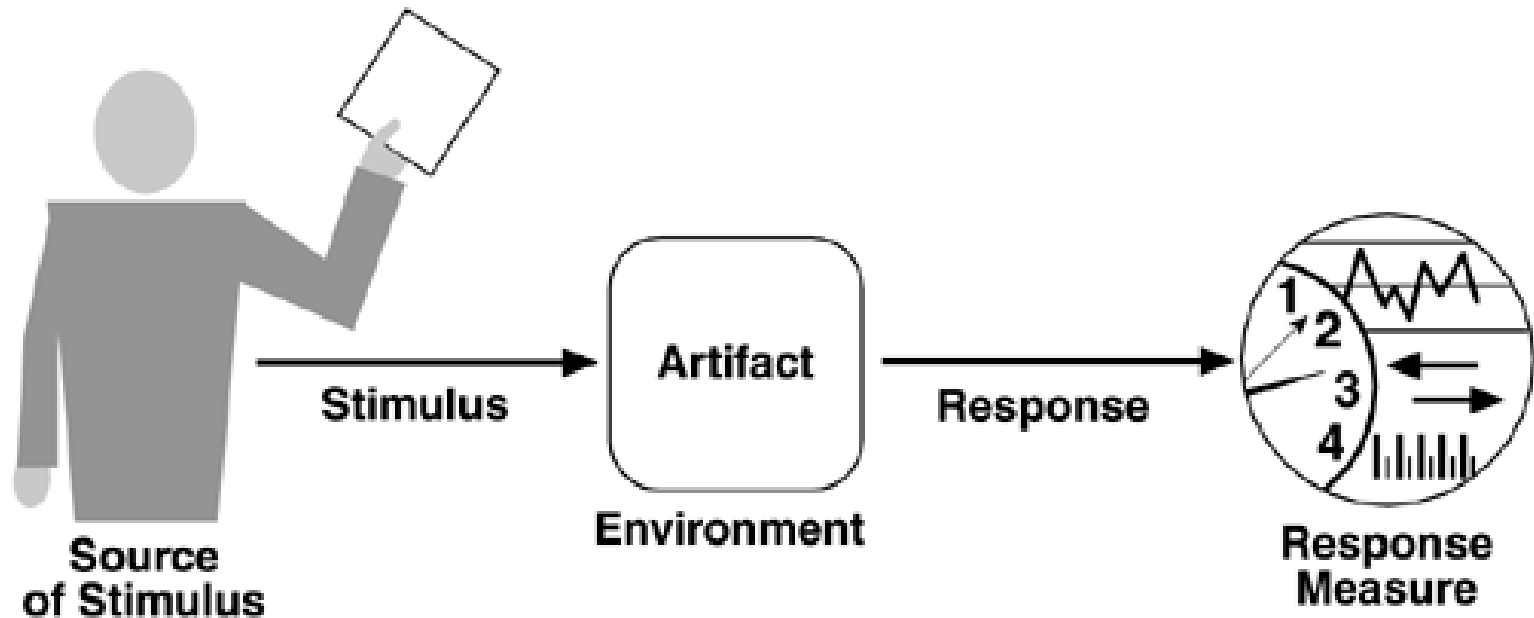
## Primer Cuatrimestre de 2010

Clase 8: Arquitecturas. Introducción a las tácticas para atributos de calidad.  
Escenarios generales.

Buenos Aires, 22 de Abril de 2010

# Recordando – Escenarios

Los usamos para especificar Atributos de Calidad



Los escenarios de Atributos de Calidad están formados por:

**Fuente del estímulo:** Interna o externa

**Estímulo:** condición que debe ser tenida en cuenta al llegar al sistema

**Entorno:** condiciones en las cuales ocurre el estímulo

**Artifact:** el sistema o partes de él afectadas por el estímulo

**Response:** qué hace el sistema ante la llegada del estímulo

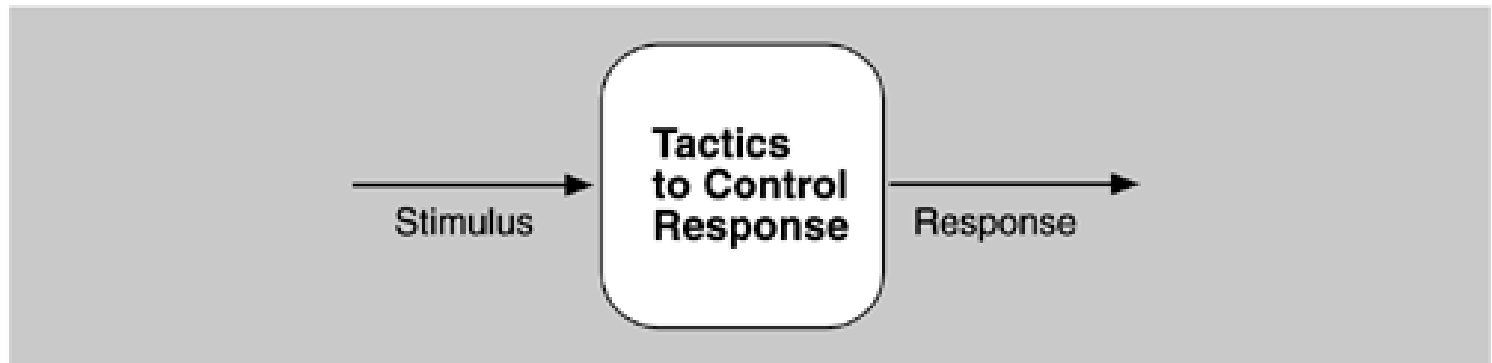
**Response measure:** cuantificación de un atributo de la respuesta

# Escenarios generales y concretos

- ▶ Escenarios generales de atributos de calidad, o simplemente “escenarios generales”:
  - ▶ Son independientes de un sistema específico y por lo tanto pueden, potencialmente, pertenecer a cualquier sistema
  - ▶ Son una guía para saber qué preguntar o pensar en escenarios candidatos
  - ▶ Están ordenados por atributo de calidad
- ▶ Escenarios concretos de atributos de calidad:
  - ▶ Son específicos a un sistema, instanciando cada uno de sus atributos

# Tácticas

- ▶ Buscan controlar las **respuestas** a determinados **estímulos**
- ▶ Las tácticas existen desde hace mucho tiempo
- ▶ Existe una gran cantidad de tácticas
- ▶ Cada táctica representa una **decisión** de diseño



# Tácticas

- ▶ Una táctica es una decisión de diseño que influencia el control sobre la respuesta un atributo de calidad
- ▶ Una colección de tácticas forman una estrategia de arquitectura
- ▶ Una táctica puede refinar otras tácticas: redundancia puede ser refinada en redundancia de datos o de procesos

***Es la bolsa de trucos del arquitecto***

# Estilos y tácticas

¿Existe una relación entre los estilos arquitectónicos de software y las tácticas?

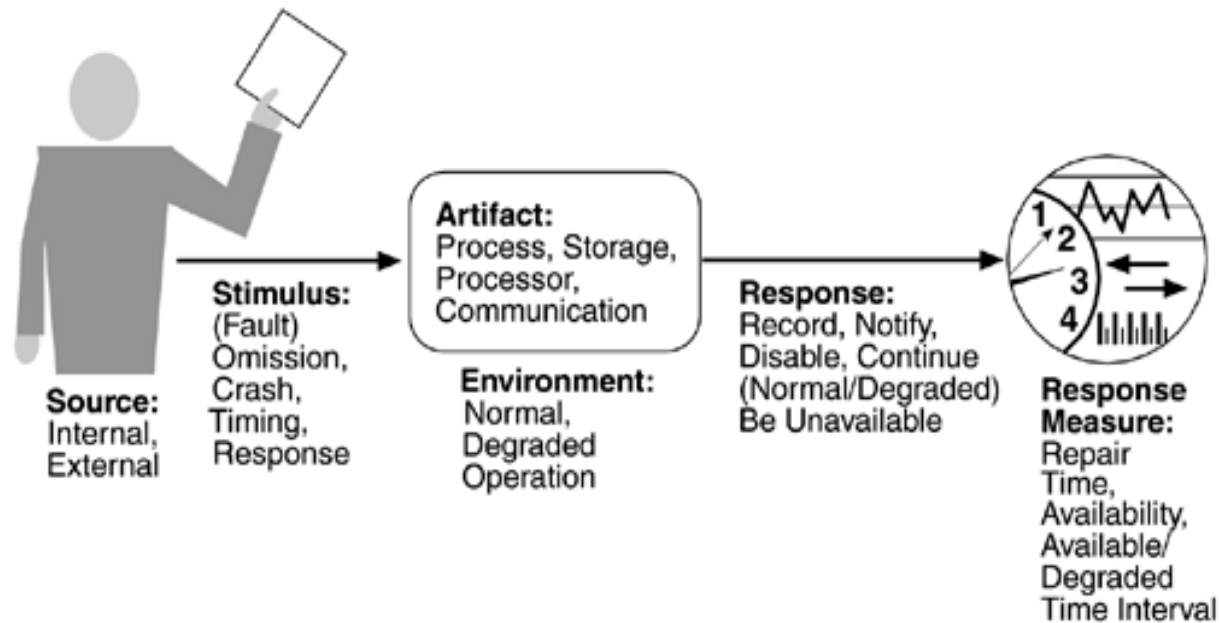
¿Cómo es esta relación?

# Estilos y tácticas

- Un arquitecto selecciona un estilo o una colección de estilos
- Cada estilo puede implementar varias tácticas
- Cada una de las tácticas tiene un determinado impacto en los atributos de calidad
- Al implementar un estilo puede haber fuertes elecciones sobre las tácticas
- Por lo tanto:
  - Es importante conocer las tácticas y su implementación
  - Al implementar estilos deben tenerse en cuenta las tácticas que implementa

# Disponibilidad - Escenario General

Disponibilidad: La habilidad del sistema de estar completa o parcialmente operacional cuando se lo requiera (y para gestionar efectivamente fallas que puedan afectar su disponibilidad)

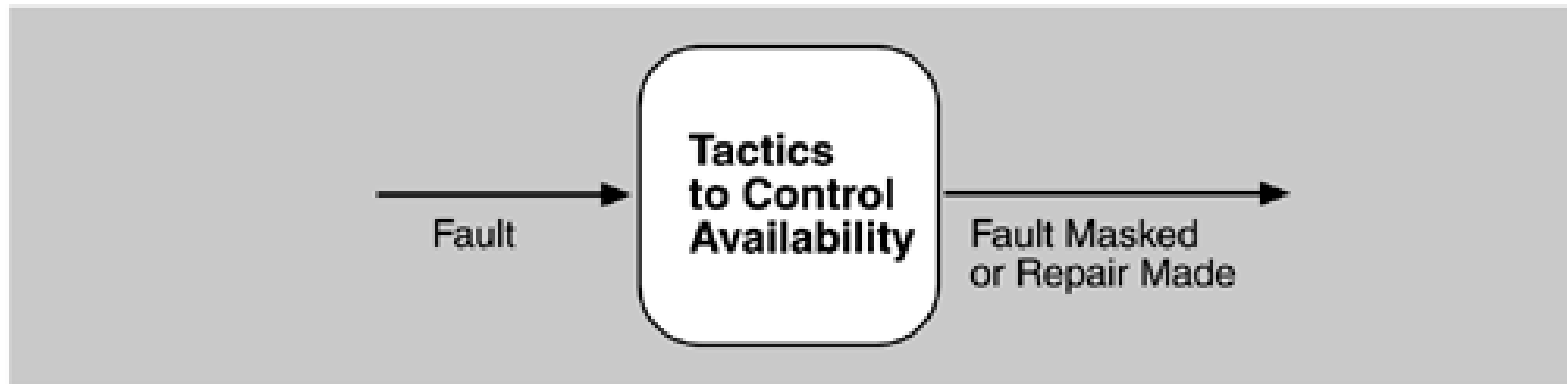




# Ejemplos de escenarios de Disponibilidad

- ▶ La actividad de la oficina central se desarrolla de lunes a viernes entre las 9.00 y las 18 hs, mientras que las tiendas están abiertas al público entre las 8.00 y las 22.00 hs y las bóvedas operan entre las 6.00 am y las 2.00 am del siguiente día. ... se espera que la disponibilidad esté relacionada con el horario y necesidad funcional.
  - ▶ La disponibilidad está relacionada con el horario y necesidad funcional de las tiendas para su correcta operación.
- ▶ El mantenimiento planeado del sistema se puede realizar fuera de los horarios de operación de las bóvedas de las tiendas. El plan de contingencia permite estar en condiciones extraordinarias a lo sumo 2 días laborales sin servicio de ningún tipo.
  - ▶ Ejemplo “downtime”: mantenimiento planeado fuera de los horarios de operación ...
  - ▶ Ejemplo “downtime”: (no planeado) ... plan de contingencia permite estar en condiciones extraordinarias a lo sumo 2 días laborales sin servicio de ningún tipo

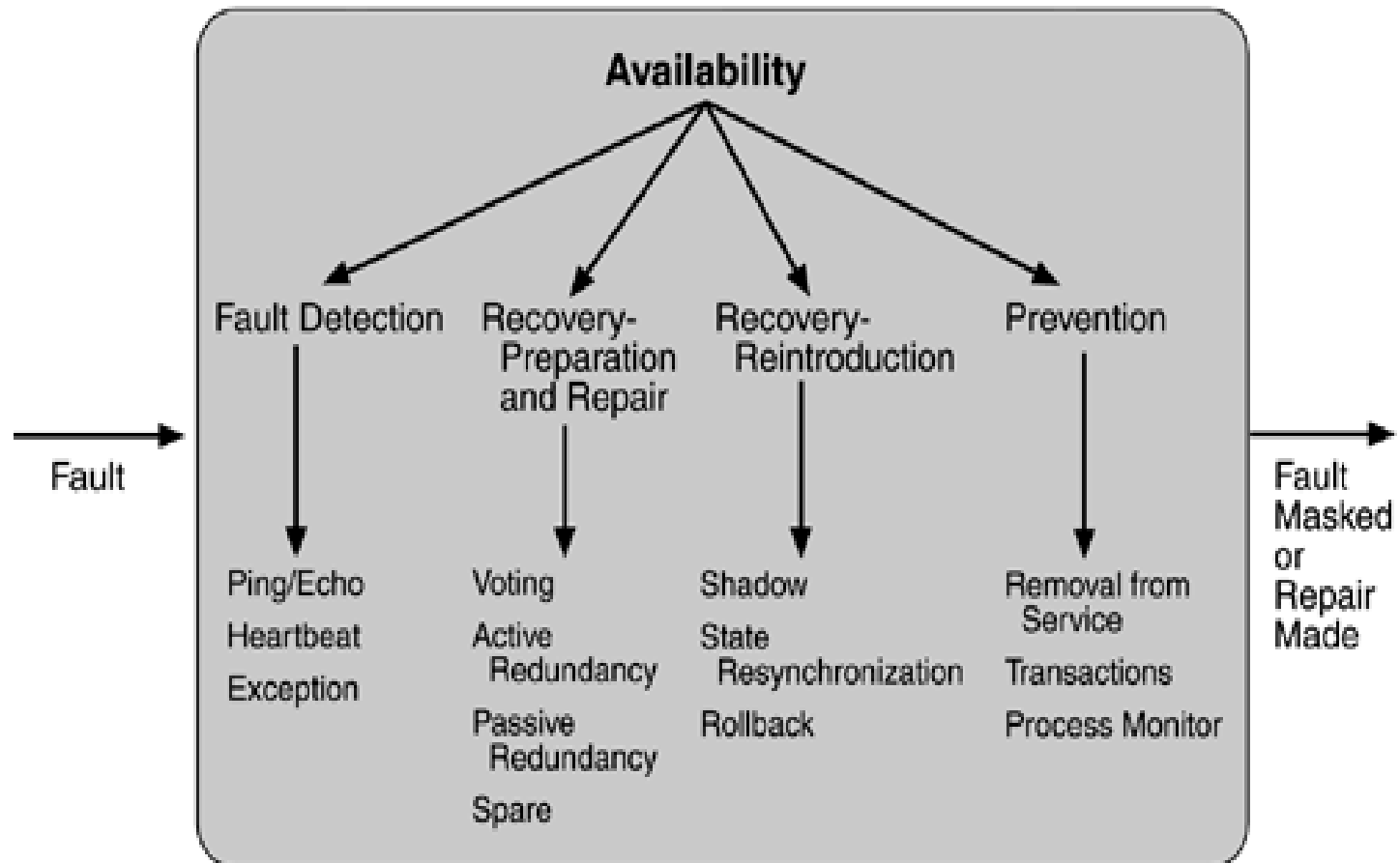
# Tácticas para Disponibilidad



# Tácticas para Disponibilidad

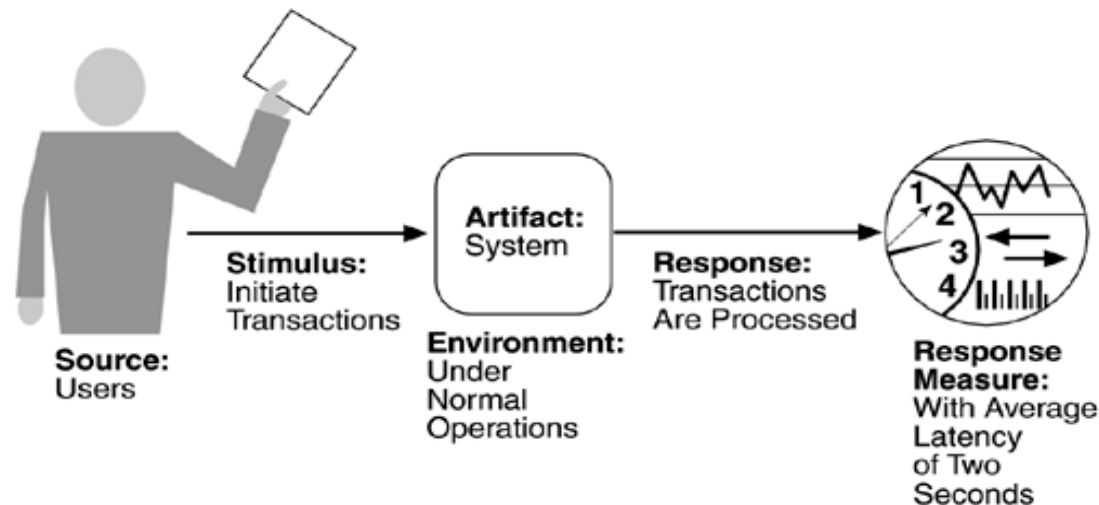
- ▶ Detección de la falla
  - ▶ Ping / Echo
  - ▶ Heartbeat
  - ▶ Excepciones
- ▶ Recuperación de la falla
  - ▶ Preparación y reparación
    - ▶ Redundancia activa
    - ▶ Redundancia pasiva
    - ▶ Votación
    - ▶ Repuesto
  - ▶ Reintroducción
    - ▶ Operación "Shadow"
    - ▶ Resincronización de estado
    - ▶ Checkpoint / Rollback
- ▶ Prevención de fallas
  - ▶ Retirar del servicio
  - ▶ Transacciones

# Tácticas para Disponibilidad



# Performance – Escenario General

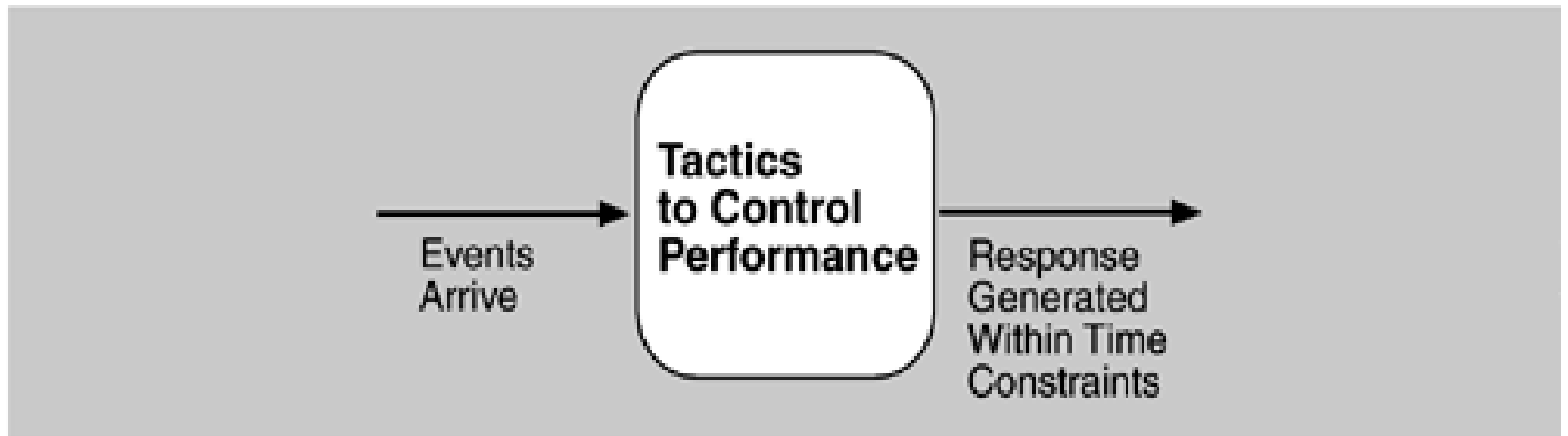
- ▶ Habilidad de un sistema para responder a un evento que ocurre (interrupciones, mensajes, pedidos de usuarios o paso del tiempo) dentro de un perfil definido.
  - ▶ Fuente: interna o externa
  - ▶ Estímulo: llegada de los eventos
  - ▶ Artefacto: servicios del sistema
  - ▶ Entorno: operación normal o “sobrecargada”
  - ▶ Respuesta: procesamiento de los estímulos
  - ▶ Medida: latencia, deadline, throughput, jitter, eventos ignorados, pérdida de datos



# Ejemplos de escenarios de Performance

- ▶ ...la oficina de ventas requiere conocer el dinero disponible en banco y realizar una predicción del dinero que será depositado ... la central requiere emitir un reporte que indica el dinero disponible estimado ... El tiempo de respuesta esperado para la emisión de este reporte no debe superar los dos minutos ...
  - ▶ Ejemplo "response time": ... no debe superar los dos minutos.
- ▶ La entrega se realiza a los 25 cajeros en promedio dentro de una ventana de tiempo que no debe superar 30 minutos... Esta actividad presenta tres ventanas entre las 7.30 y 8.00 hs, 10.30 y 11 hs y 14.30 y 15.00 hs.
  - ▶ Ejemplo "peak load": 1200 tiendas, que dentro de las tres ventanas generan cada una 25 transacciones dentro de los 30 minutos.

# Tácticas para Performance

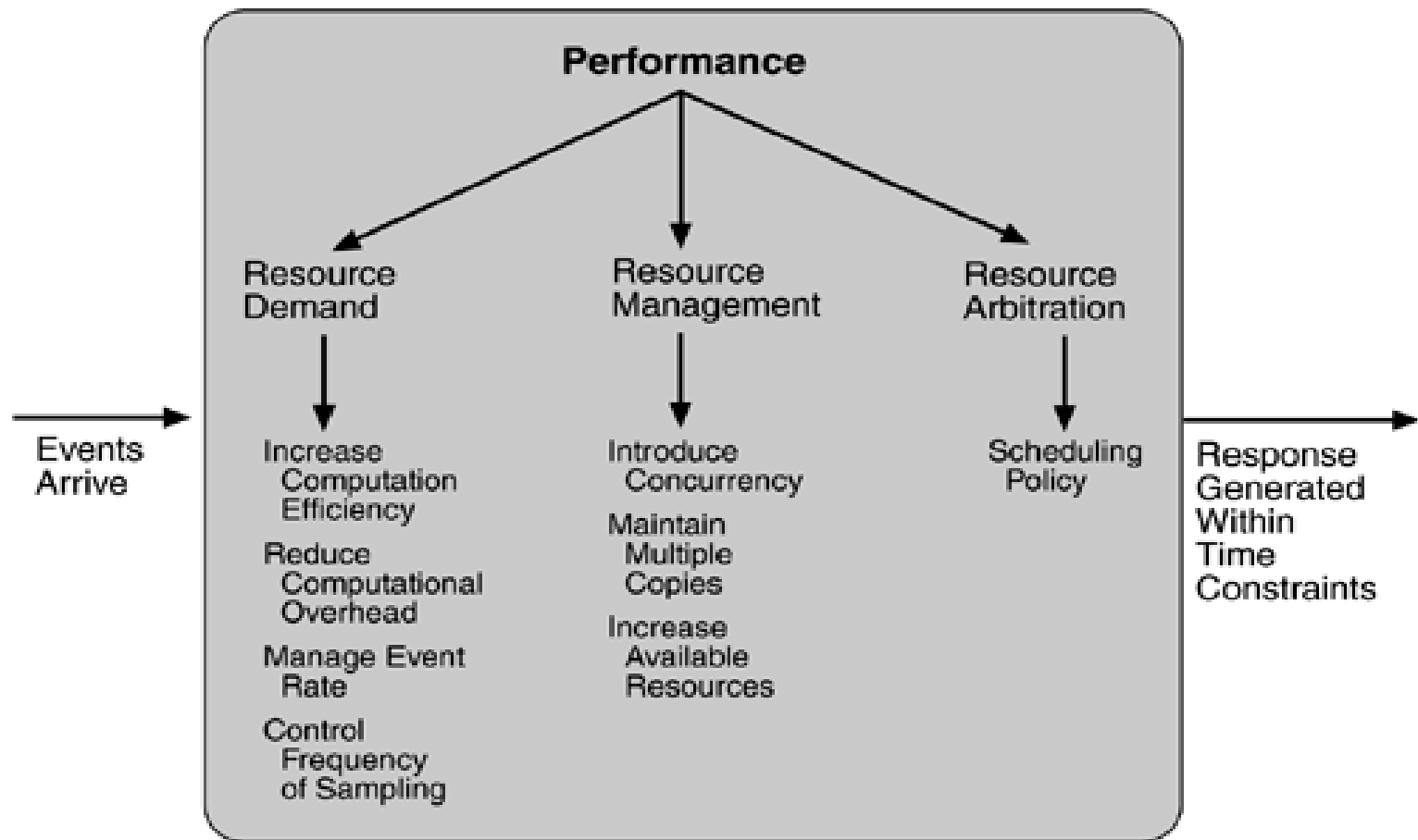


# Tácticas para Performance

- ▶ Consumo de recursos
  - ▶ Eficiencia computacional
  - ▶ Overhead computacional
  - ▶ Manejar ratio de eventos
  - ▶ Controlar frecuencia de análisis de eventos
- ▶ Administración de recursos
  - ▶ Introducir concurrencia
  - ▶ Aumentar recursos disponibles
  - ▶ Mantener múltiples copias (caching)
- ▶ Resource arbitration
  - ▶ Política de Scheduling

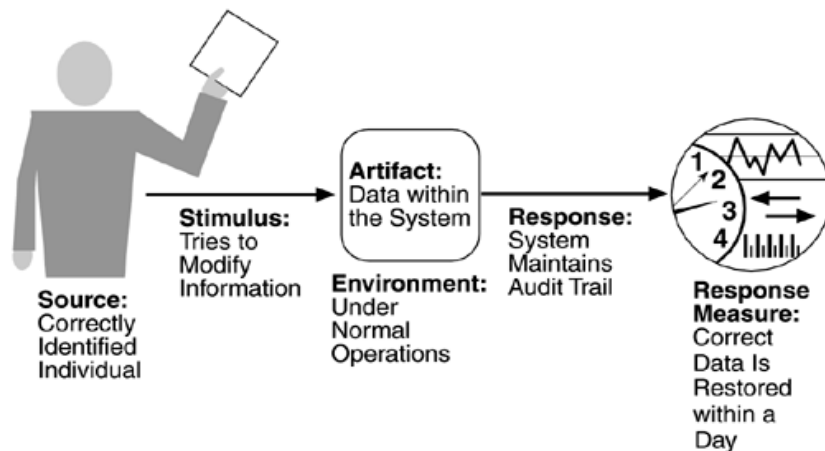


# Tácticas para Performance

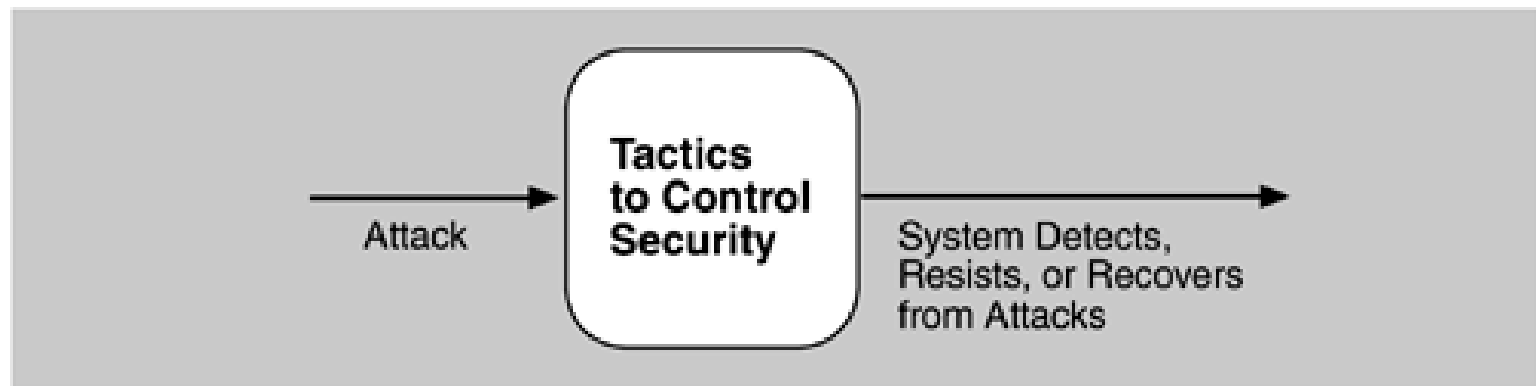


# Seguridad – Escenario General

- ▶ Seguridad: habilidad de un sistema para controlar, monitorear y auditar en forma confiable quién puede realizar qué acciones sobre el sistema y sus recursos, y la habilidad para detectar y recuperarse de fallas en los sistemas de seguridad.
  - ▶ Persona u otro sistema (autorizado o no)
  - ▶ Estímulo: ataque o intento de “romper” la seguridad (acceder a datos, servicios, reducir disponibilidad, etc)
  - ▶ Artefacto: servicios o datos del sistema
  - ▶ Entorno: Online u offline, conectado o desconectado, con o sin firewall, etc.
  - ▶ Respuesta: Autorizar / no autorizar, mantener pistas de auditoría
  - ▶ Tiempo / esfuerzo / recursos para evitar ataque, probabilidad de detectarlo, porcentaje del sistema que sigue disponible, etc.



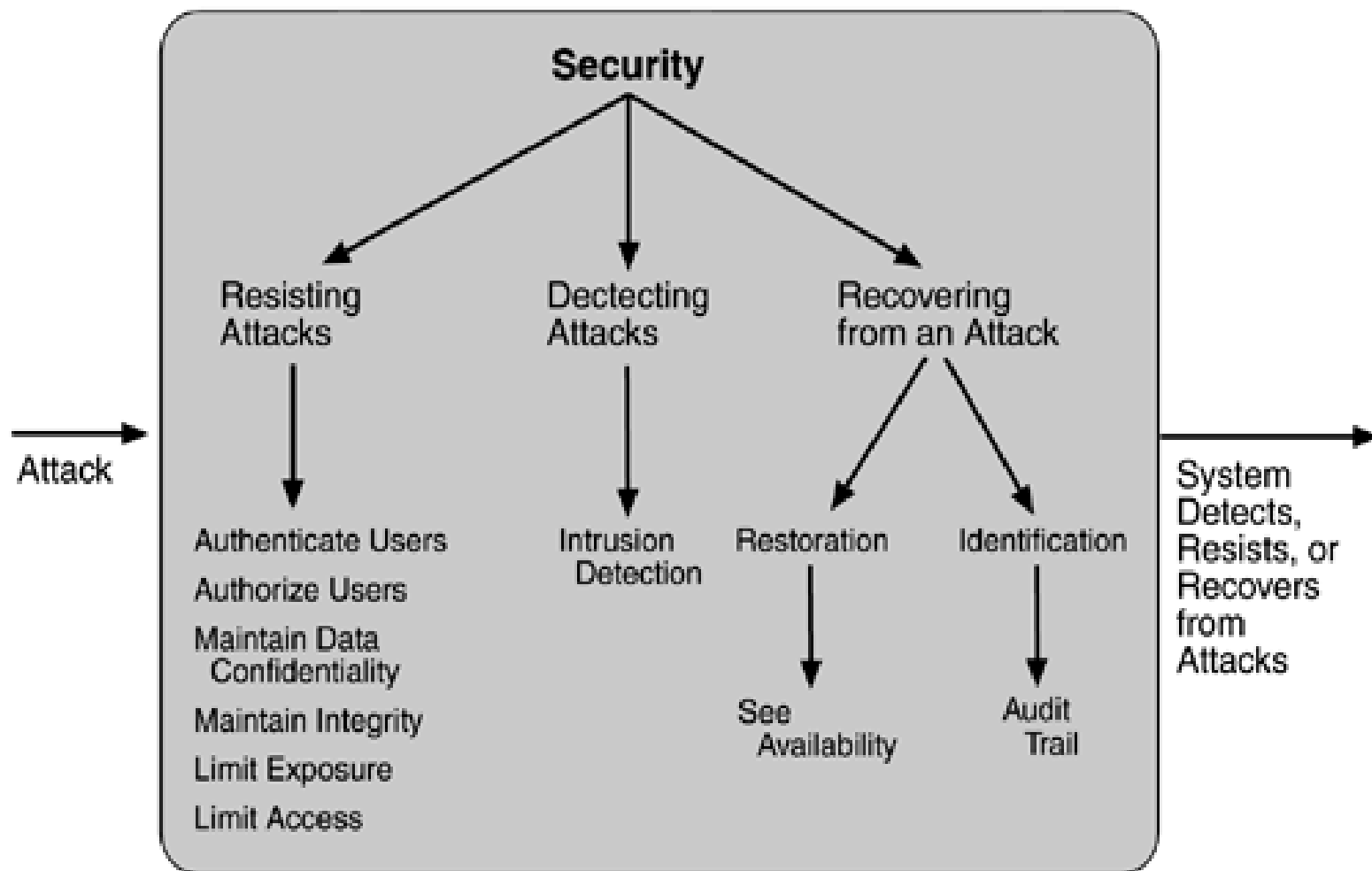
## Tácticas para seguridad



# Tácticas para Seguridad

- Resistir ataques:
  - Autenticar usuarios
  - Autorizar usuarios
  - Mantener confidencialidad de datos
  - Mantener integridad
  - Limitar exposición
  - Limitar acceso
- Detectar ataques
  - Detección de intrusos
- Recuperación de un ataque
  - Restore (idem disponibilidad)
  - Identificación (pistas de auditoría)

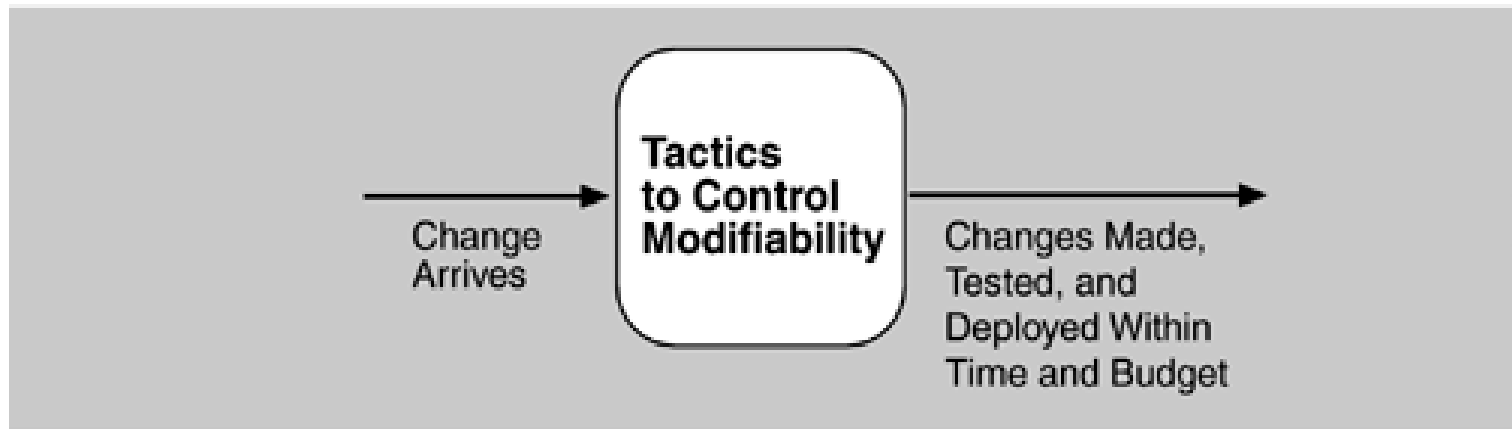
# Tácticas para Seguridad



# Facilidad de Cambios – Escenario General

- Facilidad de cambios: la habilidad del sistema para ser flexible frente a cambios inevitables durante su desarrollo y luego del despliegue. (en balance costo de construcción dentro de estos términos versus el costo de cambio)
  - Fuente: Usuario / desarrollador / administrador
  - Estímulo: quiere agregar / eliminar / modificar / variar funcionalidad / atributo de calidad / capacidad
  - Artefacto: Interfaz de usuario / entorno / plataforma / sistema / módulo / componente
  - Entorno: tiempo de diseño / compilación / ejecución
  - Respuesta: analizar impacto / hacer modificación / testear modificación / hacer deploy de la modificación
  - Medida: Costo / elementos afectados / esfuerzo / afectación de otros elementos

# Tácticas para Facilidad de Cambios

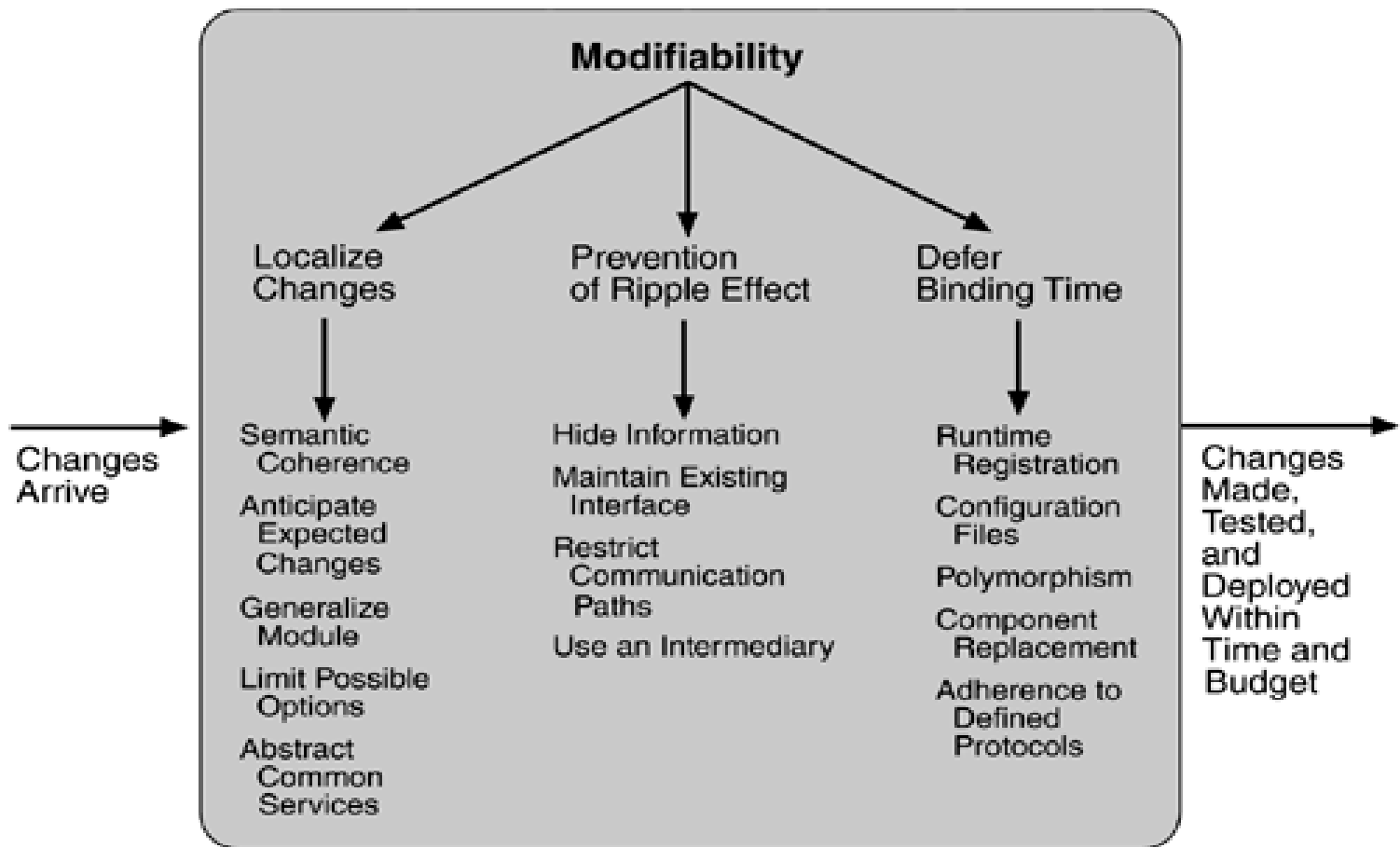


# Tácticas para Facilidad de Cambios

- Localizar modificaciones (módulos afectados)
  - Generalización
  - Coherencia Semántica
  - Anticipar cambios
  - Limitar opciones
- Prevenir efecto de propagación de cambios
  - Ocultar información
  - Conservar interfaces (adapters o interfaces)
  - Restringir caminos posibles de comunicación
  - Usar intermediarios
- Posponer momento de “binding”
  - Configuration files, polimorfismo, indirección



# Tácticas para Facilidad de Cambios



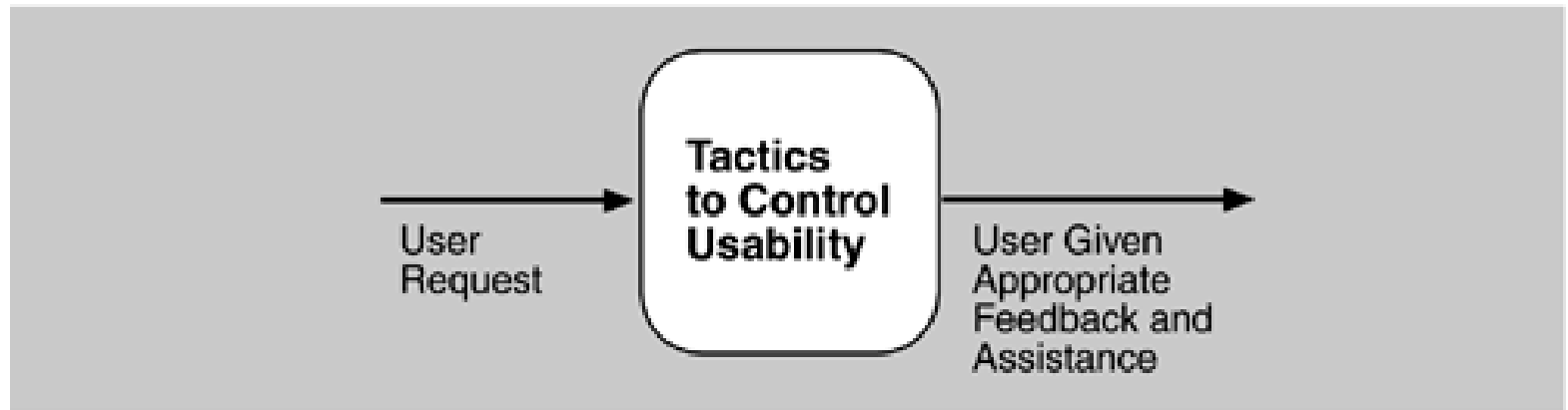
# Ejemplos de sistemas reales

- ▶ Uso de resources y entidades multi-idioma
- ▶ Uso de MVC para separar la capa de presentación
- ▶ Uso de validaciones definidas con archivos externos y expresiones regulares. Sacar y poner validaciones sigue siendo un cambio muy frecuente.
- ▶ Uso de herencia y tablas diferenciadas para especializar módulos (separación “core” – “no core”)
- ▶ Administración de tipos genéricos

# Usabilidad – Escenario General

- Usabilidad: La facilidad con la cual las personas interactúan con la aplicación en forma efectiva, y sobre como el sistema provee soporte al usuario en este sentido.
  - Fuente: Usuario final
  - Estímulo: quiere usar el sistema / usarlo eficientemente / minimizar impacto de errores / adaptar el sistema
  - Artefacto: sistema
  - Entorno: Tiempo de ejecución o configuración
  - Respuesta: características específicas de usabilidad provistas por el sistema
  - Medida: tiempo para una tarea, cantidad de errores, tiempo de entrenamiento, satisfacción de usuario, operaciones exitosas / total de operaciones.

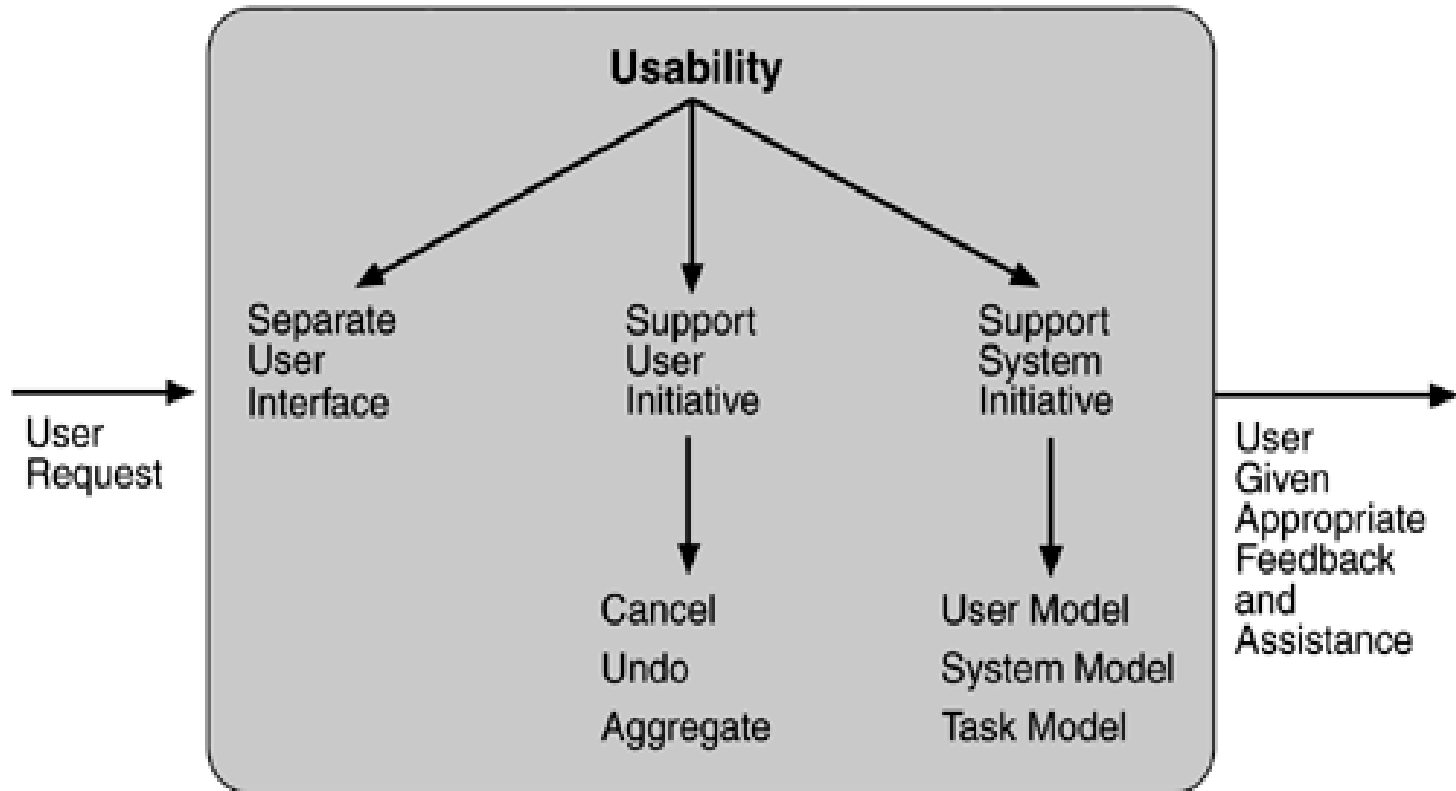
## Usabilidad – Tácticas



# Usabilidad - Tácticas

- ▶ Existen muchas tácticas para mejorar las interfaces de usuario. Algunas afectan la arquitectura, otras no.
- ▶ Tiempo de diseño
  - ▶ Separar interfaz del resto del sistema
- ▶ Seguir “reglas de oro de la usabilidad”
  - ▶ Diálogos simples y naturales
  - ▶ Hablar el lenguaje del usuario
  - ▶ Minimizar la carga de memoria
  - ▶ Ser consistente
  - ▶ Proveer feedback
  - ▶ Proveer salidas marcadas claramente
  - ▶ Proveer atajos
  - ▶ Dar buenos mensajes de error
  - ▶ Prevenir y manejar errores
  - ▶ Ayuda y documentación

## Usabilidad - Tácticas



## Tácticas de Facilidad de Testeo

