

# Ingeniería de Software II

## Segundo Cuatrimestre de 2011

Clase 16: Proceso de Arquitectura. Introducción a las tácticas para atributos de calidad. Escenarios generales.

Buenos Aires, 20 de Octubre de 2011

# El proceso de definir una arquitectura

- Definir una arquitectura implica una combinación de:
  - Creatividad / talento
  - Experiencia
  - Método
- Sobre la creatividad y el talento (Brooks, Turing Award Lecture)
  - “We have to grow them deliberately
    - Recruit for design brilliance, not talk skills
    - Make the dual ladder real and honorable
    - Career planning and mentoring, as for managers
    - Planned experiences, studies, and rotations”
  - We have to manage them imaginatively
  - We have to protect them fiercely
    - From managers
    - From managing”

# Cómo ser un mejor arquitecto

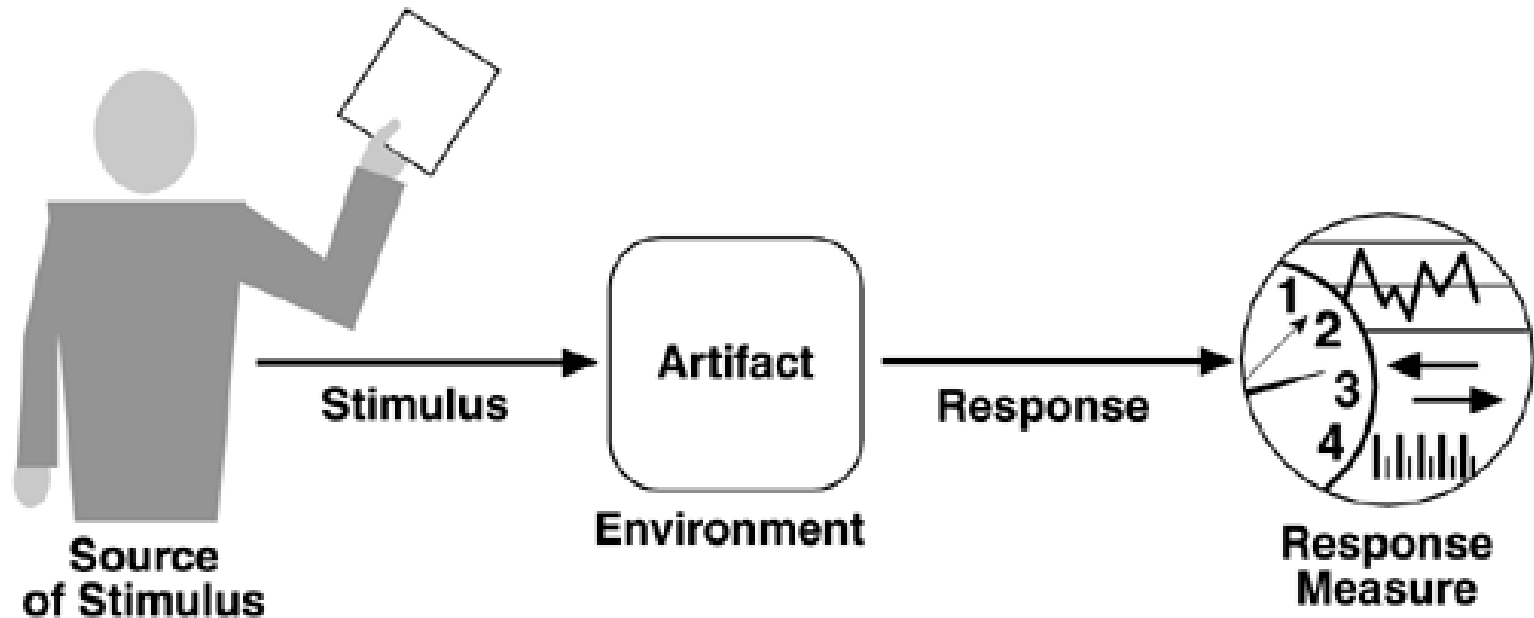
- ▶ Brooks (cont.)
  - ▶ Diseñar muchas cosas, mantener un “notebook”
  - ▶ Reflejar las experiencias de diseño
  - ▶ Estudiar otros diseños documentados
    - ▶ Revisar herramientas, software
- ▶ Usar un “chief designer”
- ▶ No usar colaboración, salvo para:
  - ▶ Determinar necesidades de los usuarios
  - ▶ Exploración conceptual, alternativas radicales
  - ▶ Revisiones de diseño

# Sobre el método

- Usar enfoques iterativos, recordar el enfoque co-evolutivo
- Buscar arquitecturas de referencia / estilos arquitectónicos del dominio
- Contrastar características de la arquitectura de referencia / estilo con los requerimientos de alto nivel
- Ejemplo de un método: ADD, Attribute Driven Design (SEI):
  - Elegir el módulo a descomponer
  - Refinar el módulo
    - Elegir drivers de arquitectura a partir de Escenarios de Atributos de Calidad y requerimientos funcionales
    - Elegir un patrón arquitectónico que satisfaga los drivers
    - Instanciar módulos y asignar funcionalidad / representar usando vistas
    - Definir interfaces de módulos hijos
    - Verificar y refinar casos de uso y escenarios
  - Iterar

# 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

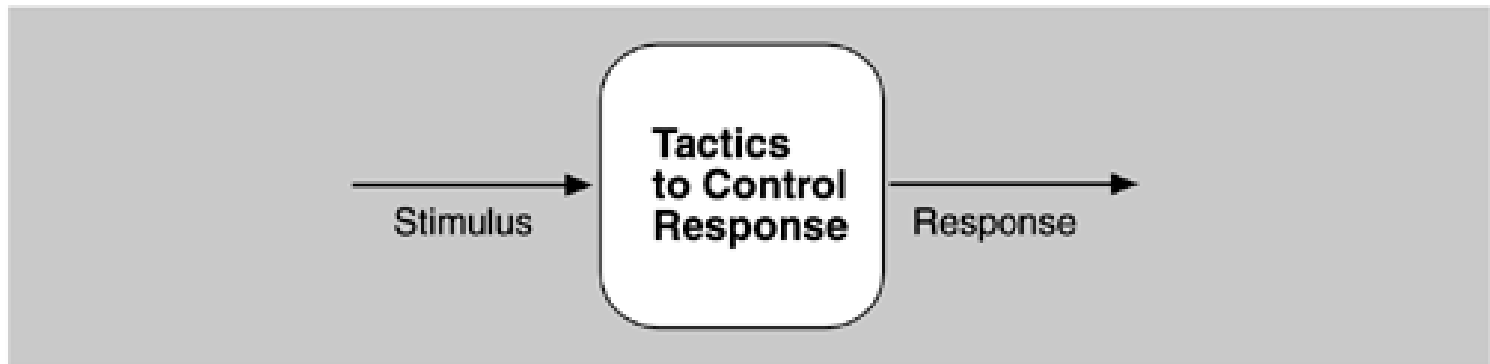
**Response measure:** cuantificación de un atributo de la respuesta

# 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**
- ▶ Son muchas y existen desde hace mucho tiempo
- ▶ Cada táctica representa una **decisión** de diseño
- ▶ Estrategia: colección de tácticas



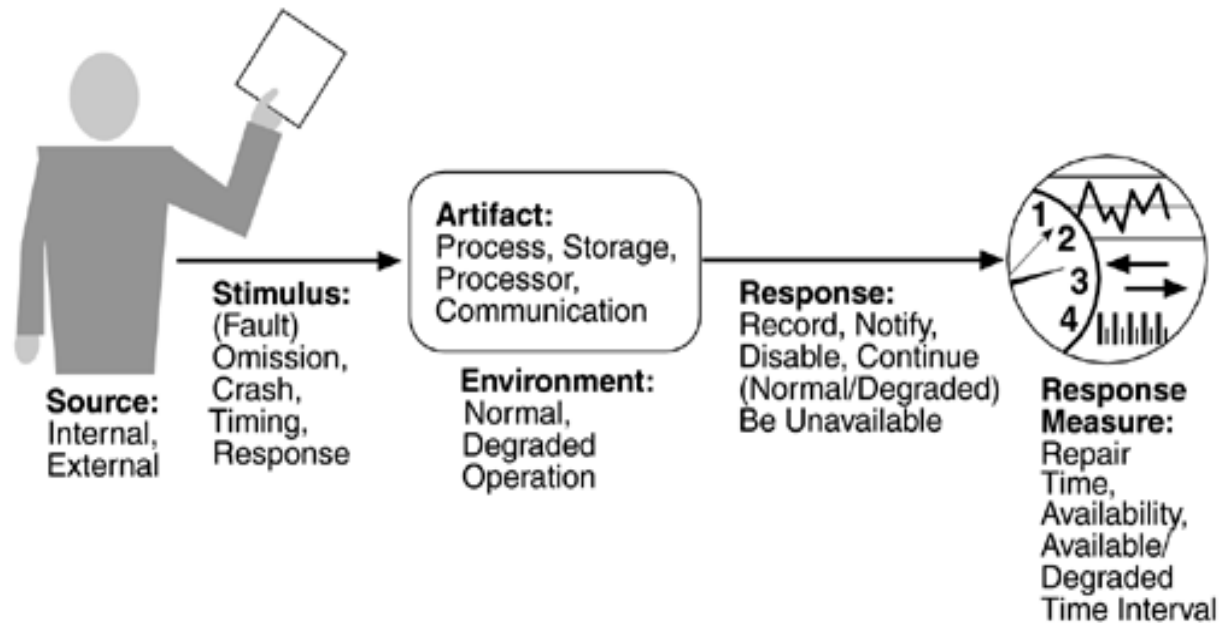
# Estilos y tácticas

- 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

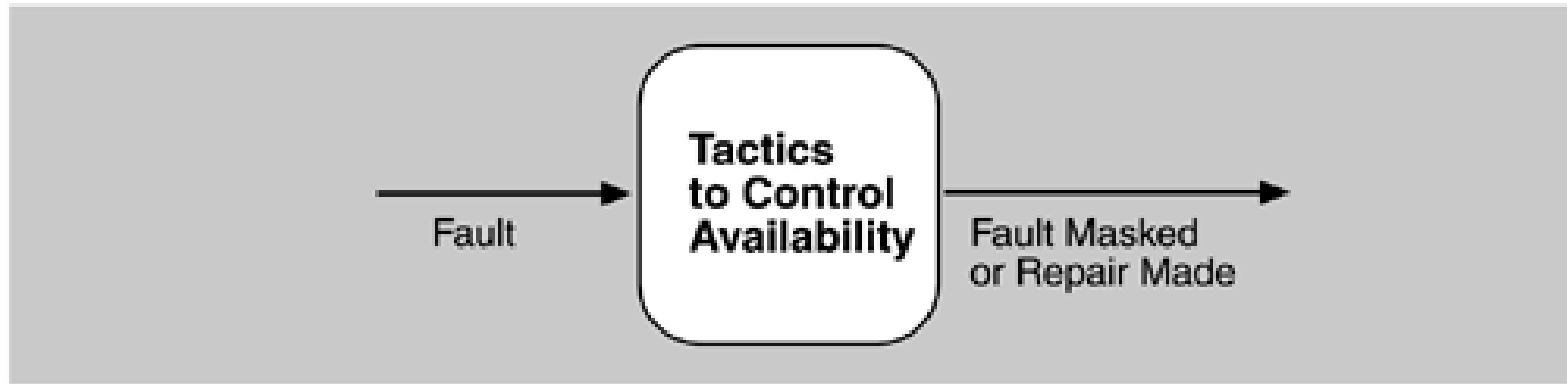
Disponibilidad: La habilidad del sistema de estar completa o parcialmente operacional cuando se lo requiera (y para gestionar efectivamente fallas que puedan afectar su disponibilidad)



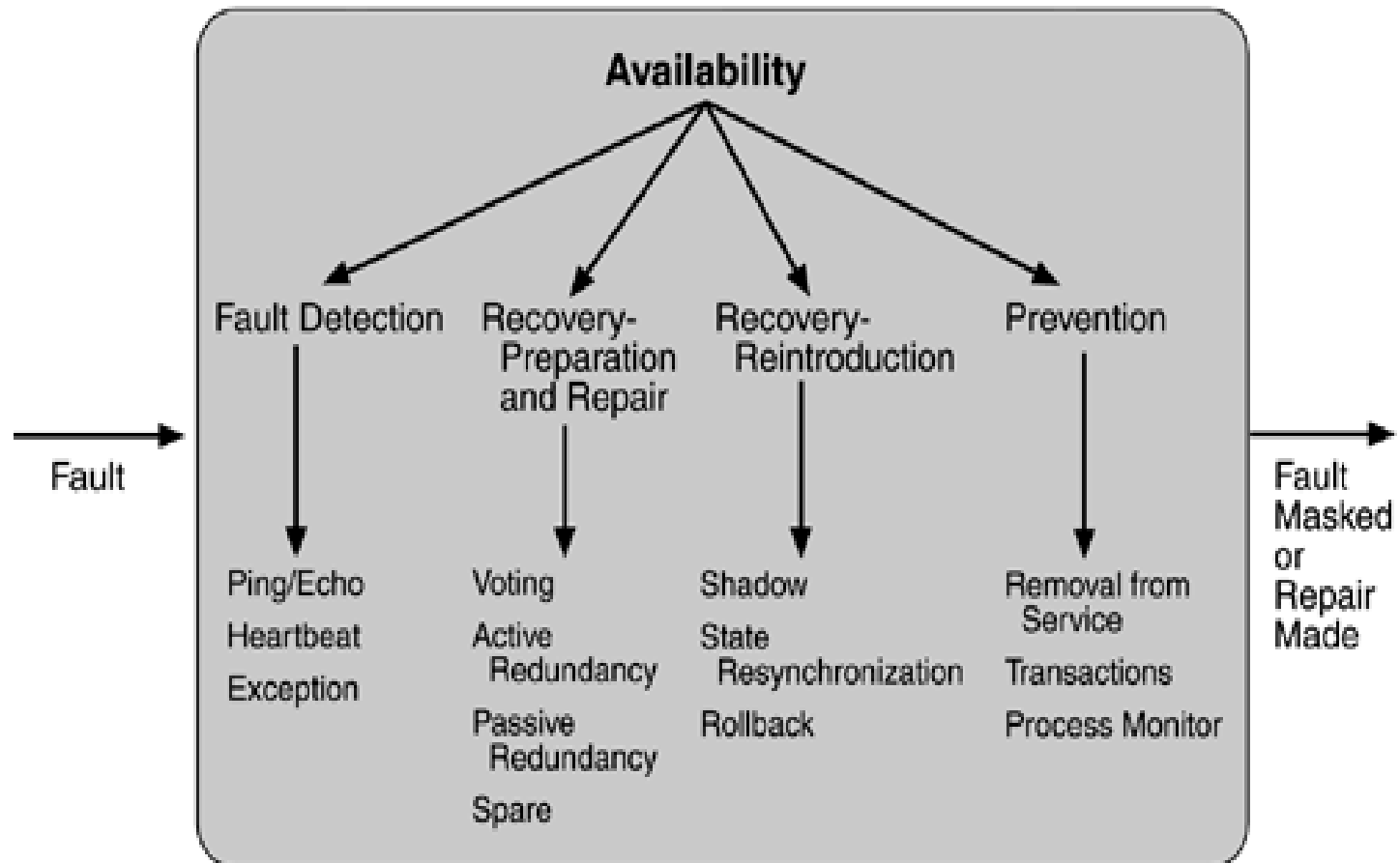
## Disponibilidad (Ejemplos SUBITE)

- ▶ “En zonas donde la cobertura 3G sea débil o inexistente, se debe mantener la conexión entre las Terminales de Cobro móviles y el Sistema Central utilizando paquetes GSM SMS.”
- ▶ “Si la comunicación entre las terminales de cobro y el Sistema Central se pierde, o los tiempos de transmisión son prohibitivos, las terminales de cobro deben seguir funcionando en modo offline, de forma transparente al usuario.”
- ▶ “Dado que el Banco Nación es el organismo encargado de centralizar todas las operaciones, y sabiendo que el sistema puede funcionar, eventualmente, de manera offline, se desea que cuando el sistema vuelva a su operatoria normal online, la información sea actualizada en el Banco Nación prontamente.”

# Tácticas para Disponibilidad

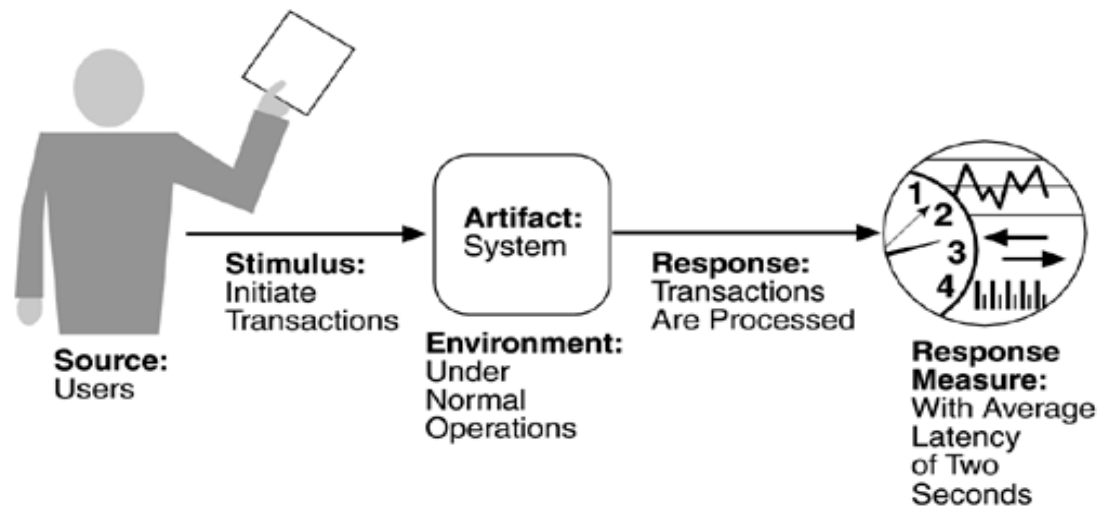


# Tácticas para Disponibilidad



# Performance – Escenario General

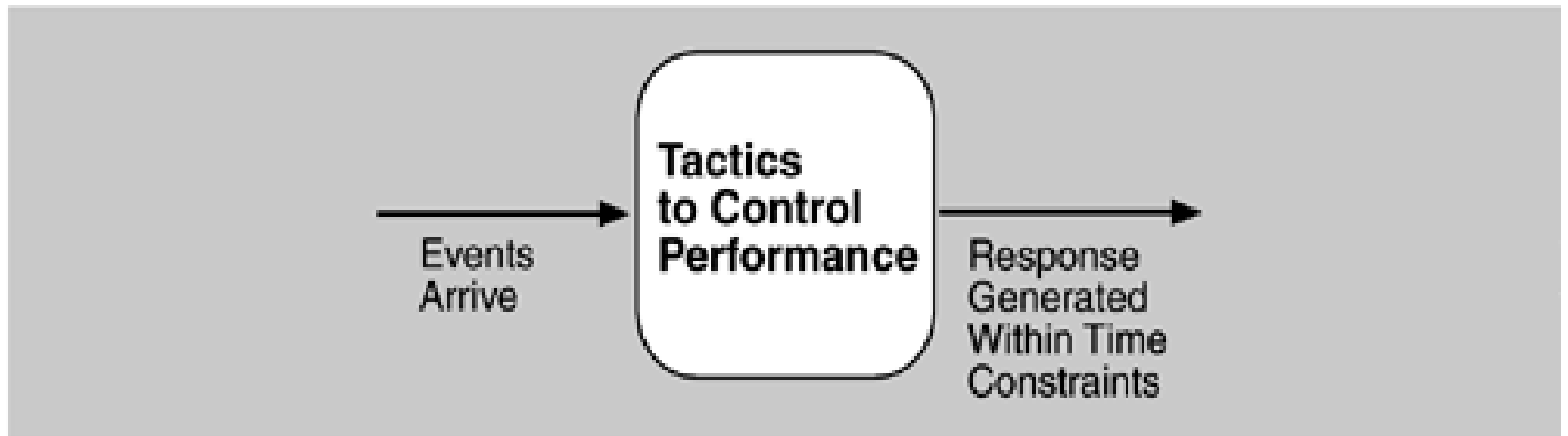
- ▶ Habilidad de un sistema para responder a un evento que ocurre (interrupciones, mensajes, pedidos de usuarios o paso del tiempo) dentro de un perfil definido.
  - ▶ Fuente: interna o externa
  - ▶ Estímulo: llegada de los eventos
  - ▶ Artefacto: servicios del sistema
  - ▶ Entorno: operación normal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



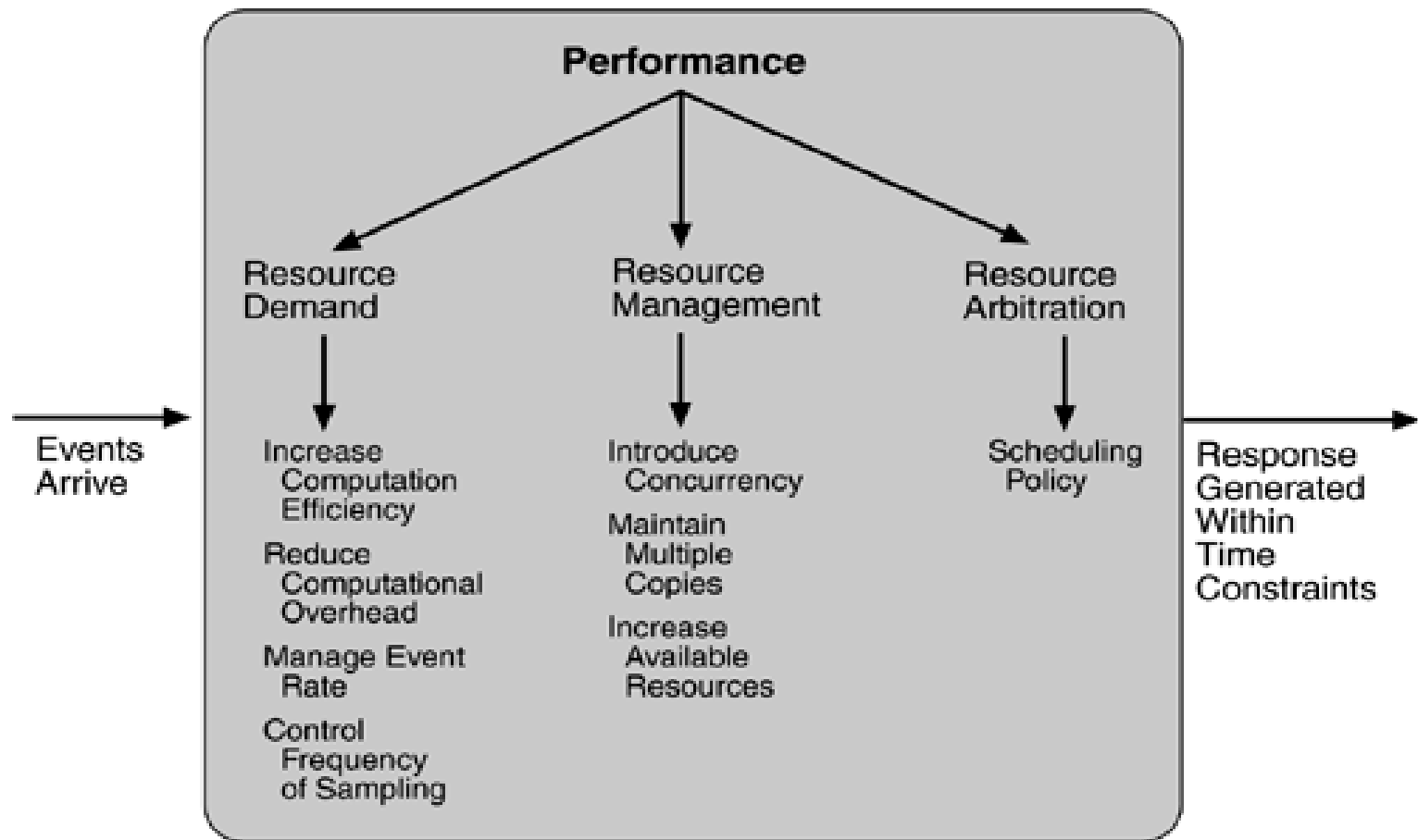
## Performance (Ejemplo SUBITE)

- ▶ “Se requiere que el sistema sea capaz de realizar las operaciones de autenticación y cobro en a lo sumo 1 segundo ya que, de otro modo, no tendría buena acogida en los usuarios, ni en los choferes de colectivos. Esto es especialmente crítico en horas pico.”

# Tácticas para Performance



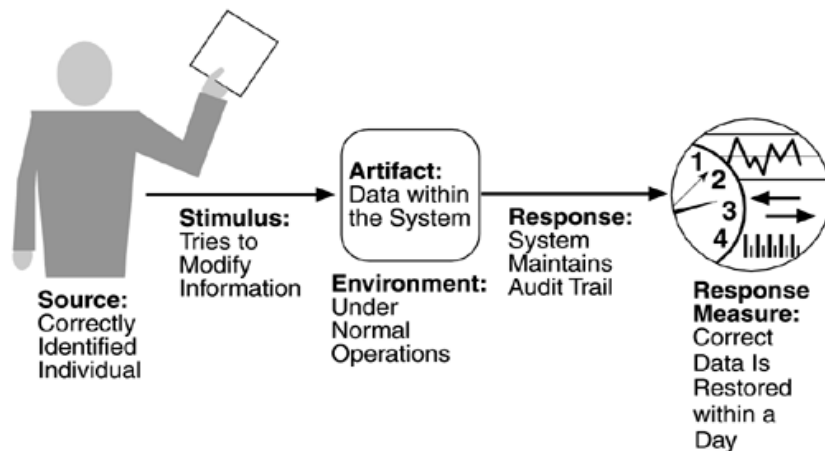
# Tácticas para Performance





# Seguridad – Escenario General

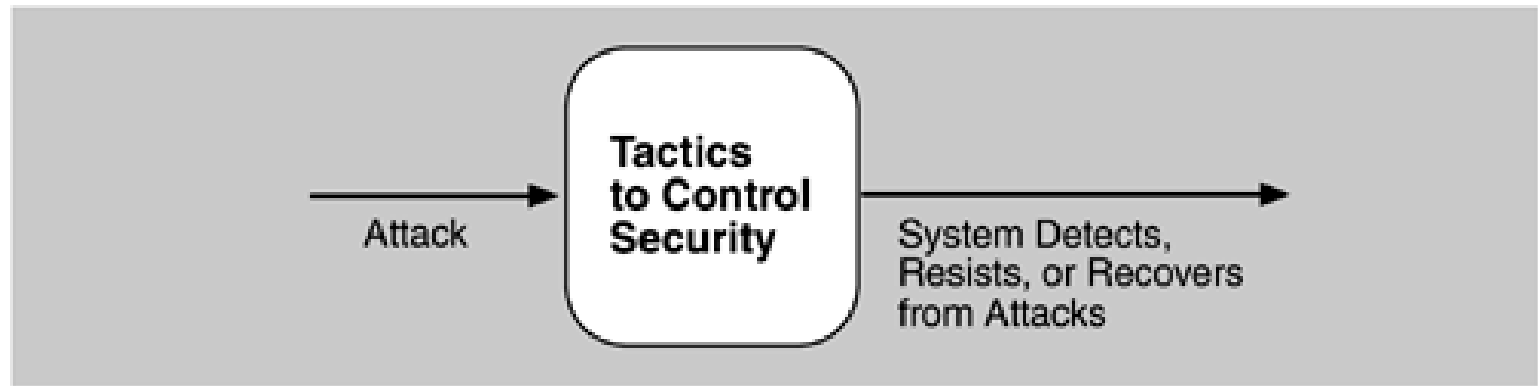
- ▶ Seguridad: habilidad de un sistema para controlar, monitorear y auditar en forma confiable quién puede realizar qué acciones sobre el sistema y sus recursos, y la habilidad para detectar y recuperarse de fallas en los sistemas de seguridad.
  - ▶ Persona u otro sistema (autorizado o no)
  - ▶ Estímulo: 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.



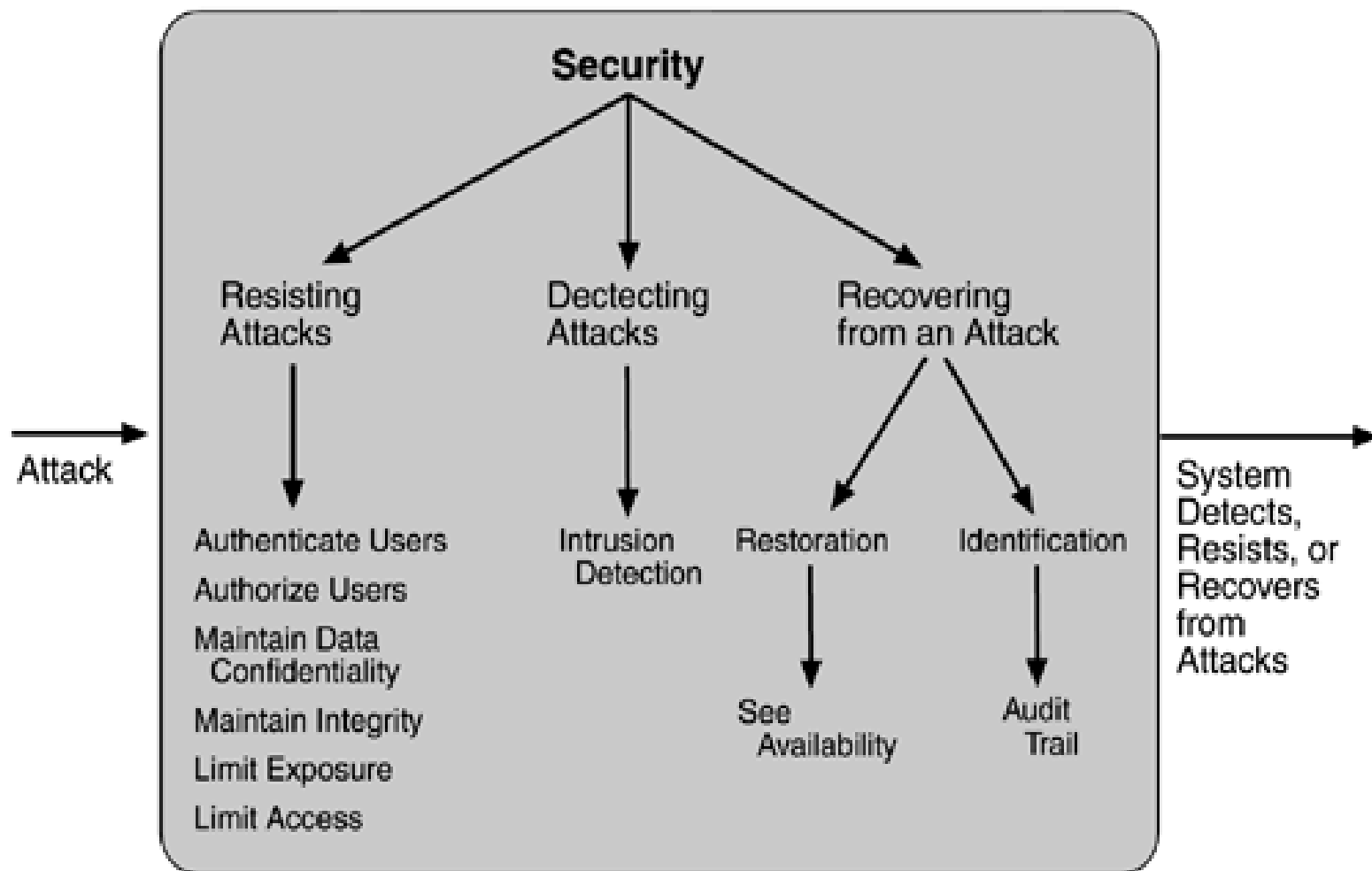
## Seguridad (Ejemplos SUBITE)

- ▶ “Es crítico que nadie fuera del Sistema Central y el Banco Nación conozca ninguno de los datos de los usuarios, sus compras realizadas, recargas efectuadas, etc...”
- ▶ “Es también muy importante que los datos (por ej. de una recarga) lleguen de forma correcta, y sin ningún tipo de modificación por factores externos como la forma de transmisión.”
- ▶ “Deben de detectarse potenciales diferencias malintencionadas entre los saldos almacenados de forma local en las tarjetas y los valores guardados para las mismas en el Sistema Central. Nadie debería viajar ‘gratis’.”
- ▶ “El Sistema Central debe mantener un registro completo de absolutamente todos los viajes que se realizaron.”

# Tácticas para seguridad



# Tácticas para Seguridad



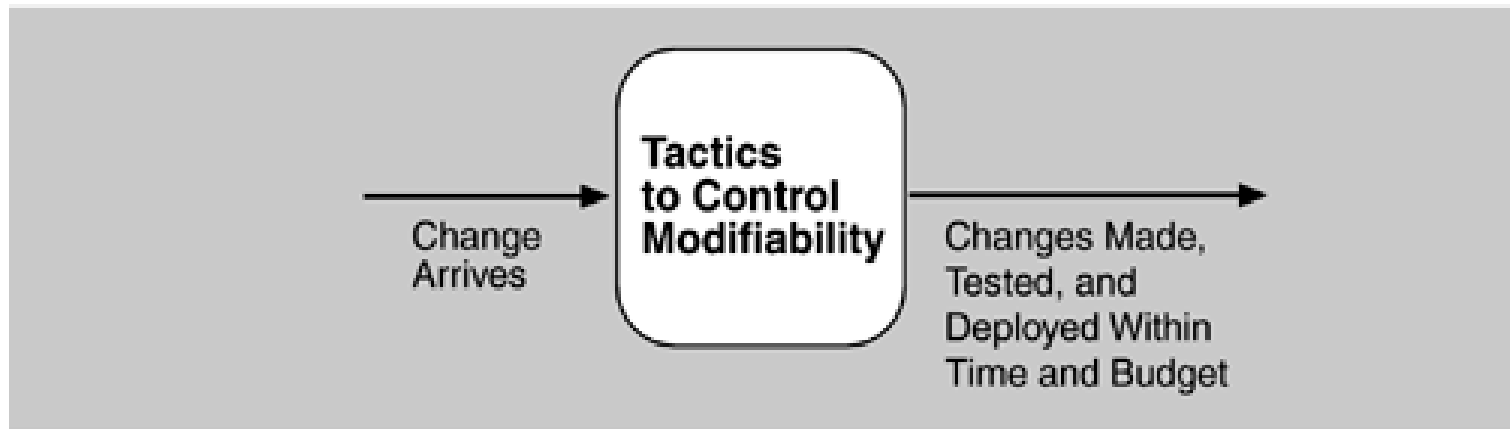
# Facilidad de Cambios – Escenario General

- 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

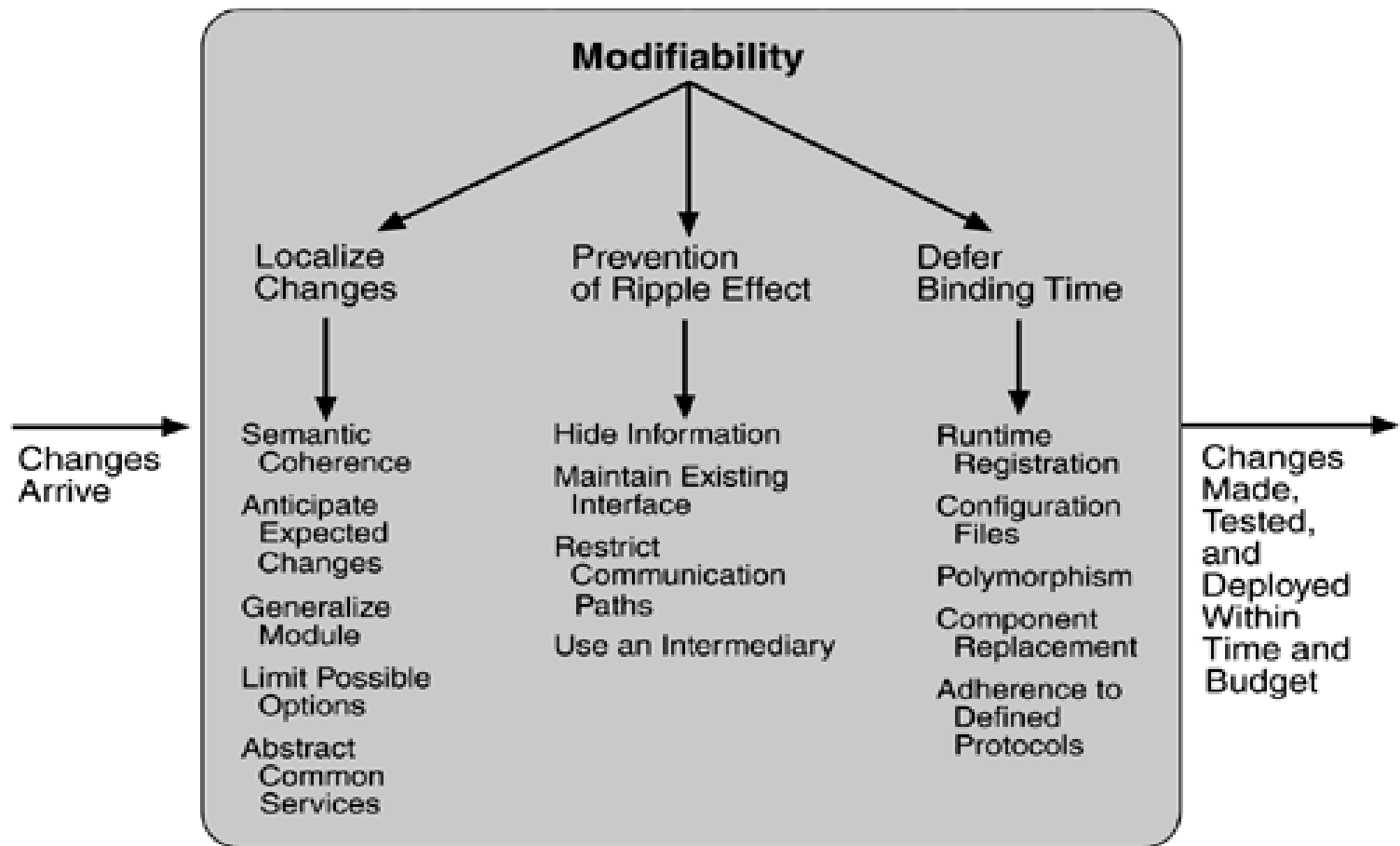
## Facilidad de Cambios (Ejemplos SUBITE)

- ▶ “Si bien existen varias maneras para realizar la recarga de una tarjeta de proximidad, es probable que en el futuro se desee aceptar nuevas formas de pago, por ejemplo, vía web. Es por ello que se requiere tener un sistema fácil de extender a nuevas formas de pago sin que esto impacte demasiado y sea rápido de implementar.”
- ▶ “Se pretende, en un futuro, implementar un sistema tarifario basado en zonas, en el cual el costo del pasaje este en relación con la cantidad de zonas que el pasajero atraviesa y no respecto de la cantidad de transportes que utilice. Por lo tanto, se espera que el sistema sea fácil de modificar para aceptar nuevas formas de realizar el cobro de los pasajes.”

## Tácticas para Facilidad de Cambios



# Tácticas para Facilidad de Cambios





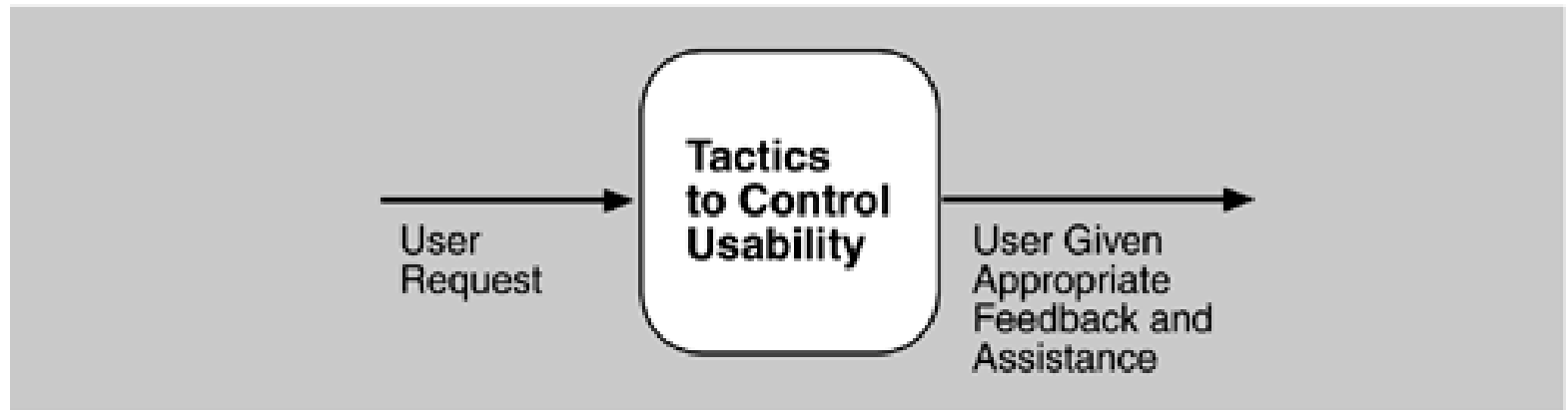
# Ejemplos de sistemas reales

- ▶ Uso de resources y entidades multi-idioma. Prever internacionalización.
- ▶ Uso de MVC para separar la capa de presentación. Especial cuidado al seguir el “framework de moda”
- ▶ Aislar con information hiding componentes comerciales “cerrados” de dudoso éxito
- ▶ 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

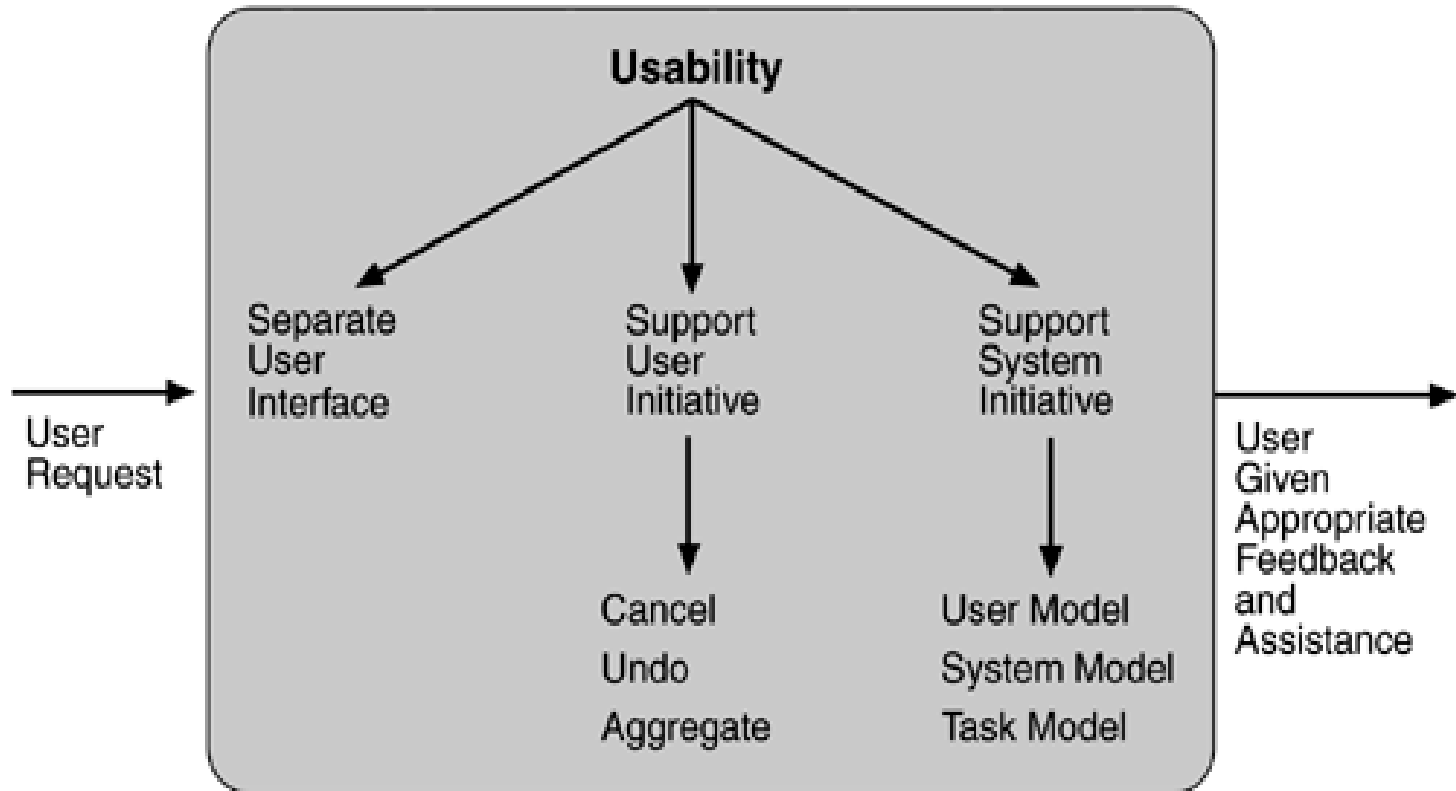
# Usabilidad – Escenario General

- 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



Fuente: Software Architecture in Practice, 2nd ed. Bass, Clements, Kazman. Addison Wesley, 2003.

# Usabilidad - Tácticas

- ▶ 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

## Tácticas de Facilidad de Testeo

