



Atributos de calidad

Requerimientos no funcionales

01

DEFINICION

" IS A MEASURABLE OR TESTABLE PROPERTY
OF A SYSTEM THAT IS USED TO INDICATE
HOW WELL THE SYSTEM SATISFIES THE
NEEDS OF
ITS STAKEHOLDERS. YOU CAN THINK OF A
QUALITY ATTRIBUTE AS MEASURING THE
“GOODNESS” ”

Características

1. **Origen del evento:** Es alguna entidad como un humano, un sistema o algún otro actor que genera el estímulo.
2. **Estímulo:** Es la condición que requiere una respuesta cuando llega al sistema
3. **Entorno/contexto:** El estímulo se da en ciertas condiciones como "sobrecarga del sistema" o "condiciones normales" (si hay más de una, especificarla)
4. **Artefacto:** Aquél que es estimulado, puede ser una serie de sistemas, el sistema entero, alguna pieza del mismo.
5. **Respuesta:** La actividad que tiene lugar cuando el estímulo llega.
6. **Medida de la respuesta:** Cuando se responde, de alguna manera tenemos que medir cómo se respondió, y tiene que ser testable.

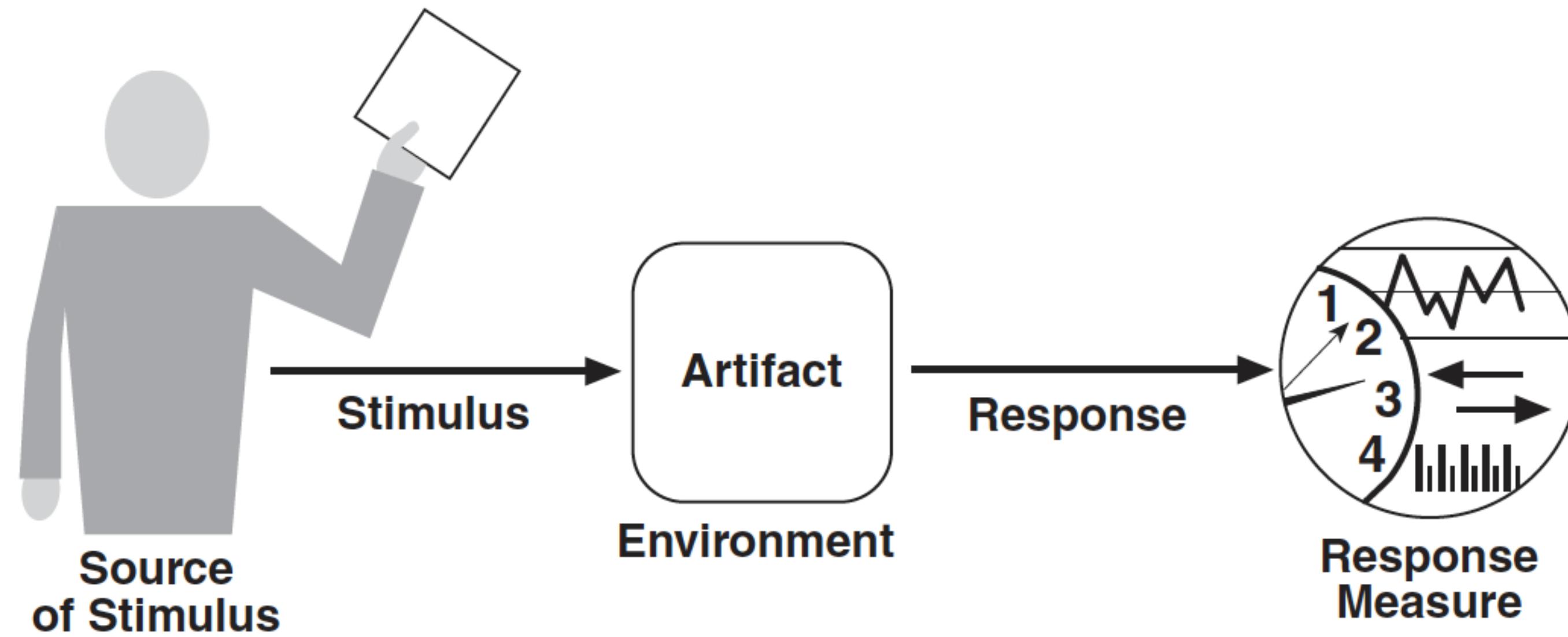


FIGURE 4.1 The parts of a quality attribute scenario

Disponibilidad / Availability

- ¿Qué haría fallar a tu sistema?
- ¿Qué tan probable es que ocurra?
- ¿Cuánto tiempo vamos a darnos para repararlo?



Disponibilidad / Availability

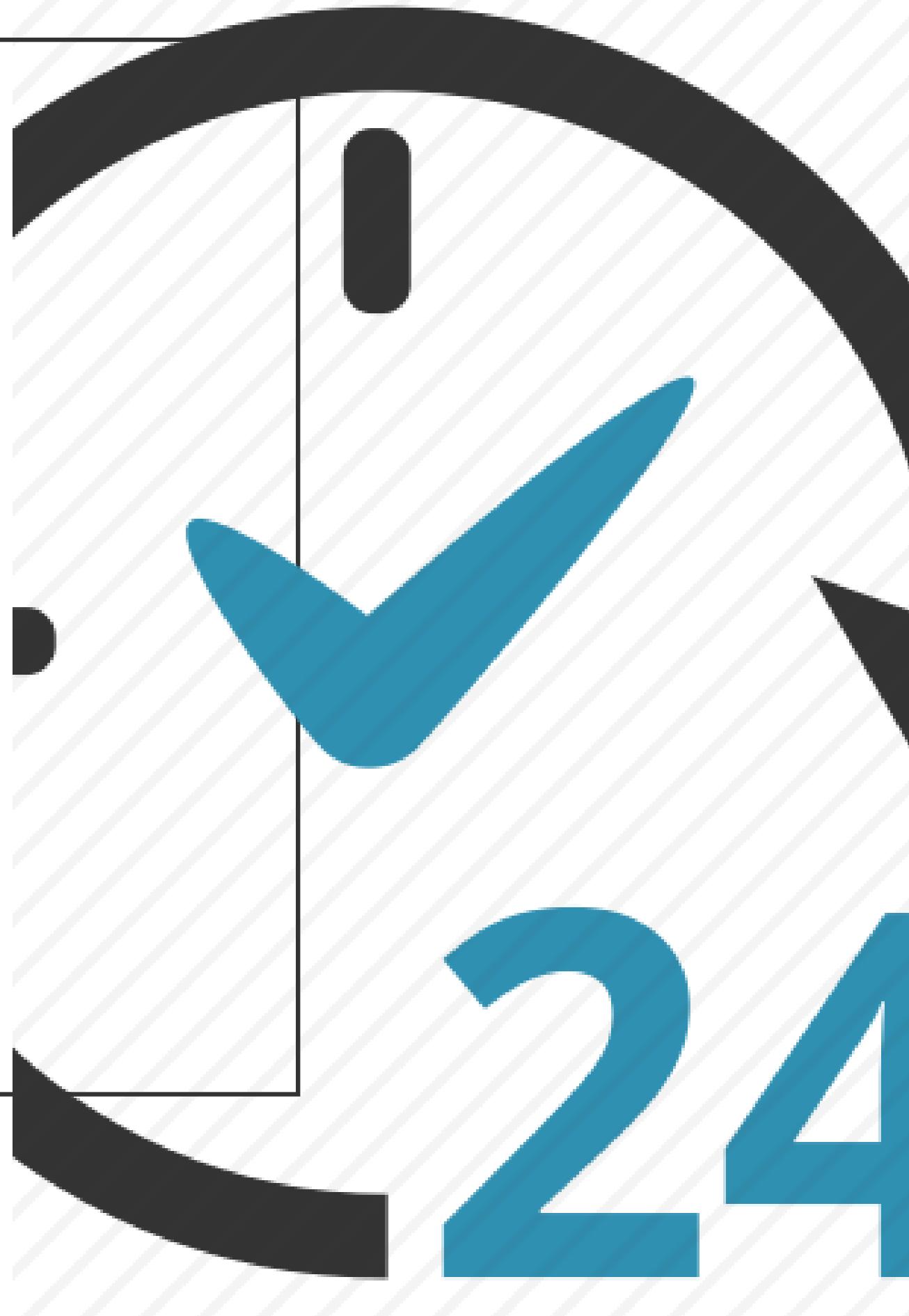
05

- **Fuente:** el origen de una falla puede ser Interno o externo. Ej: personas, hardware, software, infraestructura..
- **Estímulo:** la falla, que puede ser alguna de estas:
 - **Omisión** - no responde a un input
 - **Crash** - muchas omisiones
 - **Timing** - responde lento o muy rápido
 - **Response** - responde un valor incorrecto

Atributos de calidad



- **Artefacto:** Procesadores, canales de comunicación, almacenamiento, procesos
- **Entorno:** Operacion normal, arranque, apagado, modo reparacion,...
- **Respuesta:** Detectar y aislar, luego recuperarse de la falla
- **Evaluación de respuesta:** Tiempo en el que el sistema debe estar disponible, porcentaje de disponibilidad, tiempo para detectar la falla, tiempo para arreglarla



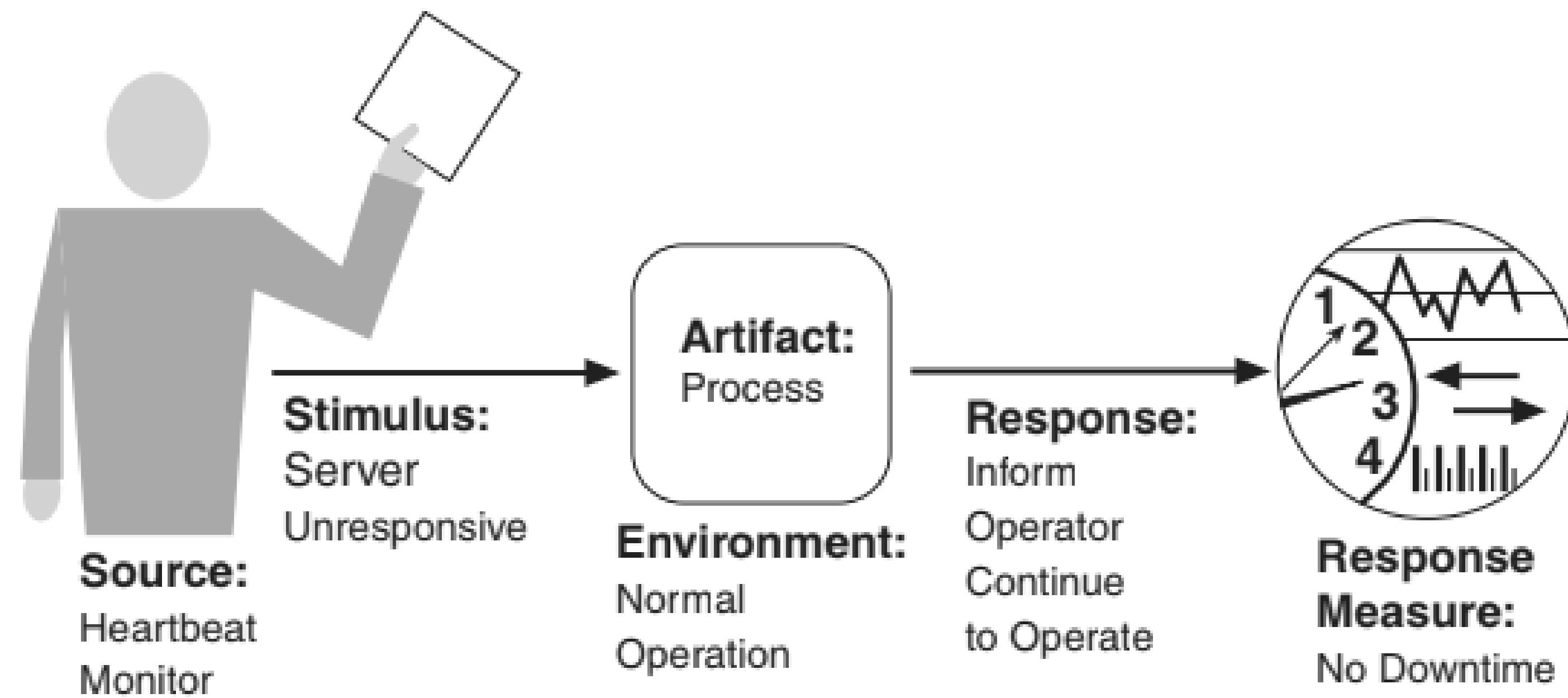
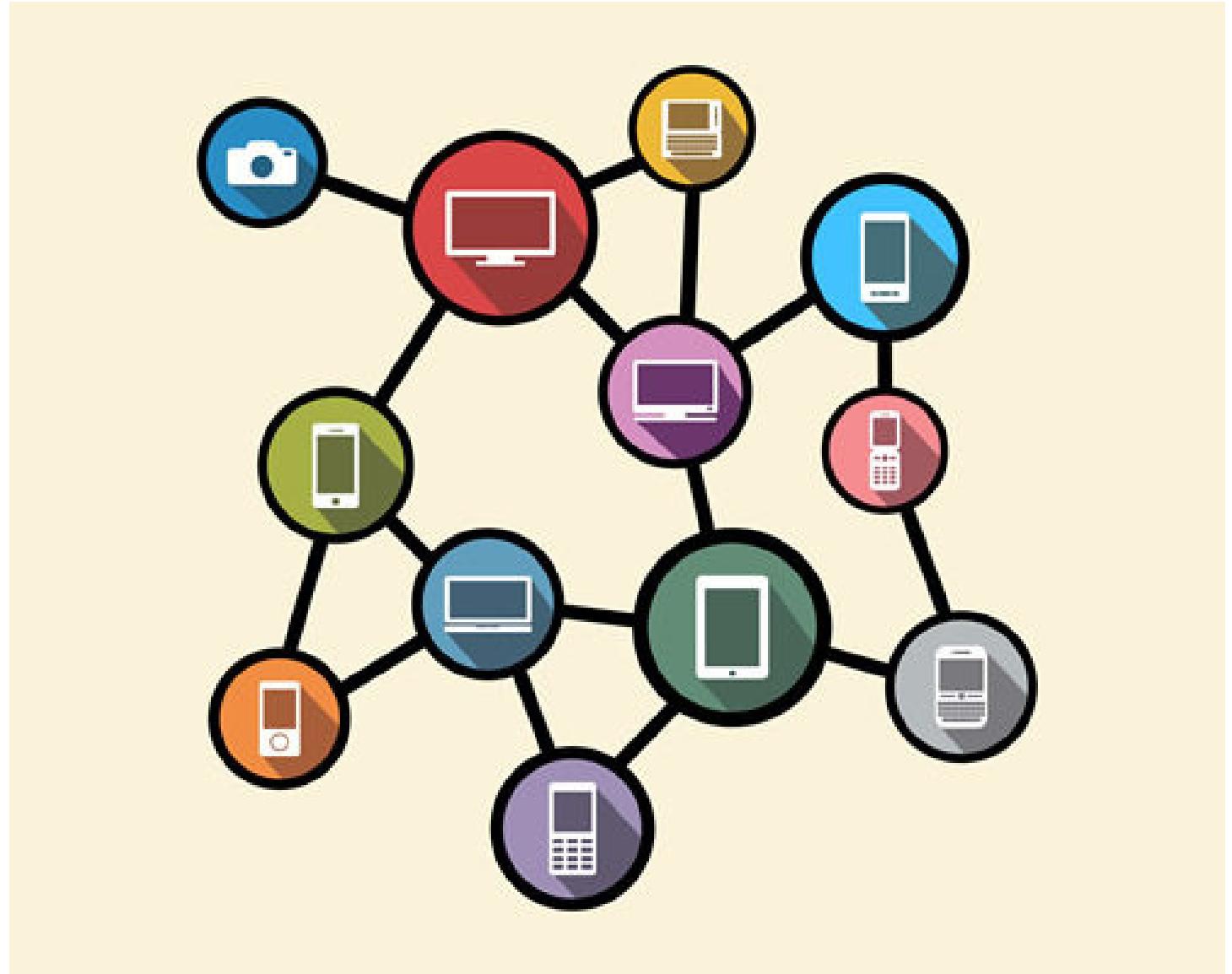


FIGURE 5.3 Sample concrete availability scenario



Interoperabilidad

La **habilidad** de **dos o más sistemas** de intercambiar información a través de **interfaces** en un determinado **contexto**



Interoperabilidad

- **Fuente:** Un sistema solicita interoperar con otro sistema
- **Estimulo:** Pedido de intercambio de información con otro sistema.
- **Artefacto:** El sistema que necesita interoperar
- **Entorno:** Se detecta en runtime o antes el sistema con el que se va a interoperar
- **Respuesta:**
 - Se rechaza y notifica / Se acepta e intercambia información
 - log del pedido
- **Evaluación de respuesta:**
Porcentajes de respuestas ok y porcentaje de pedidos rechazados

Modifiability

Habilidad de nuestro sistema de absorver cambios
¿Qué cambios vamos a permitir?
¿Qué costo tienen?



Modifiability

- **Fuente:** Usuario final, developer o system admin
- **Estimulo:** Se pide un cambio, cambiar un atributo de calidad, ..
- **Artefacto:** Código, datos, recursos, ..
- **Entorno:** Runtime, compile time, ..
- **Respuesta:**
 - Hacer el cambio
 - Testear el cambio
 - Deploy del cambio
- **Evaluación de respuesta:** Costo del cambio en términos de:
 - Cantidad de artefactos modificados
 - Esfuerzo
 - Tiempo calendario
 - Costo monetario
 - Cantidad de nuevos defectos introducidos





Performance

Habilidad de cumplir con las expectativas de tiempo de respuesta



Performance

- **Fuente:** Interna o externa al sistema
- **Estimulo:** Llegada de eventos periodicos, esporádicos o estocásticos
- **Artefacto:** El sistema o alguno/os componentes del mismo
- **Entorno:** Modo operativo: normal, emergencia, sobrecargado, pico de carga
- **Respuesta:** Procesar eventos, puede cambiar nivel de servicio (normal a sobrecargado por ej.)
- **Evaluación de respuesta:** Latencia, deadline, throughput, jitter, miss rate



Seguridad

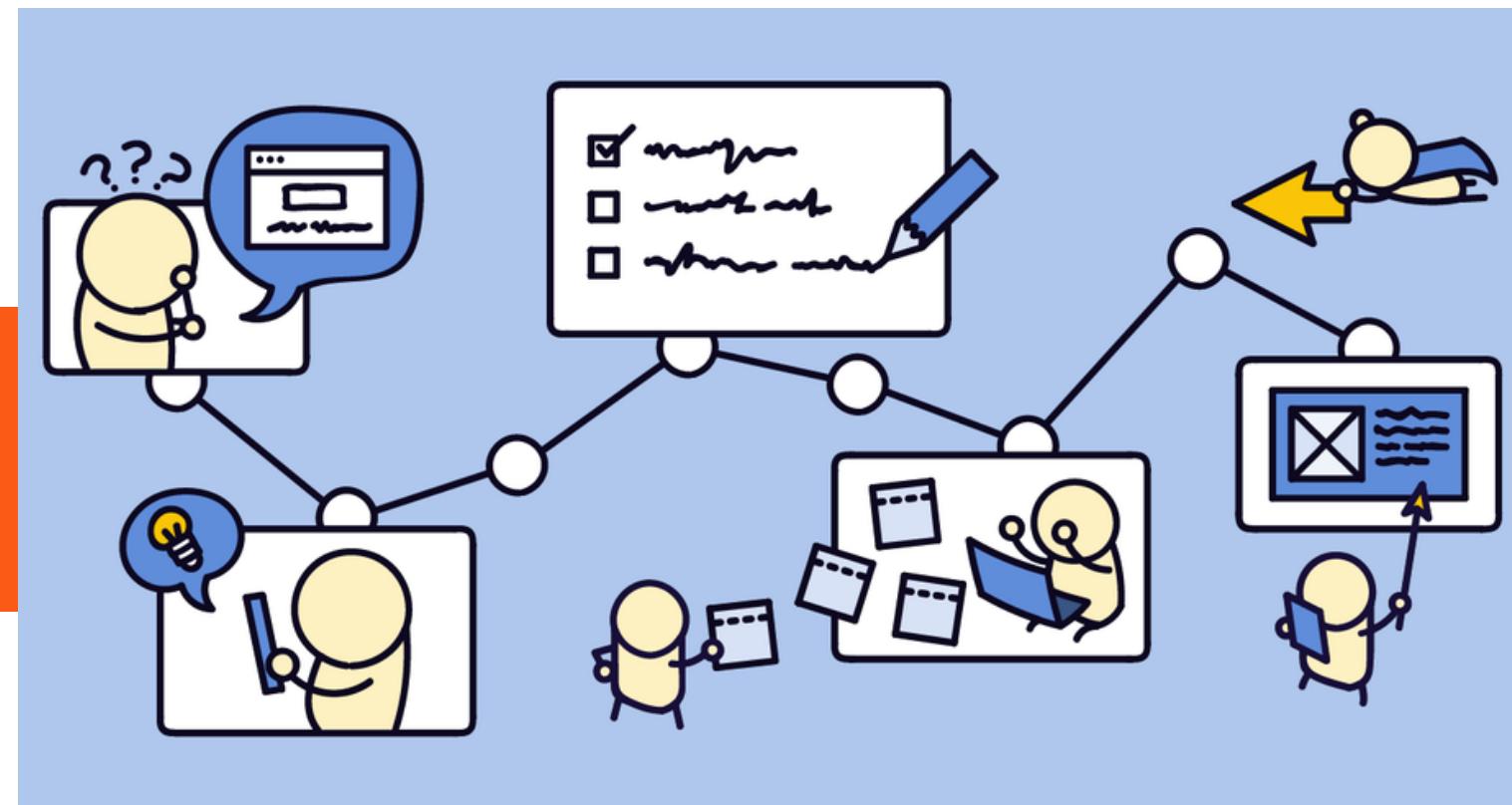
Capacidad de proteger nuestros datos de accesos no autorizados y **garantía de acceso a aquellos que sí lo estén.**

**MUY RELACIONADO CON
AVAILABILITY**



Seguridad

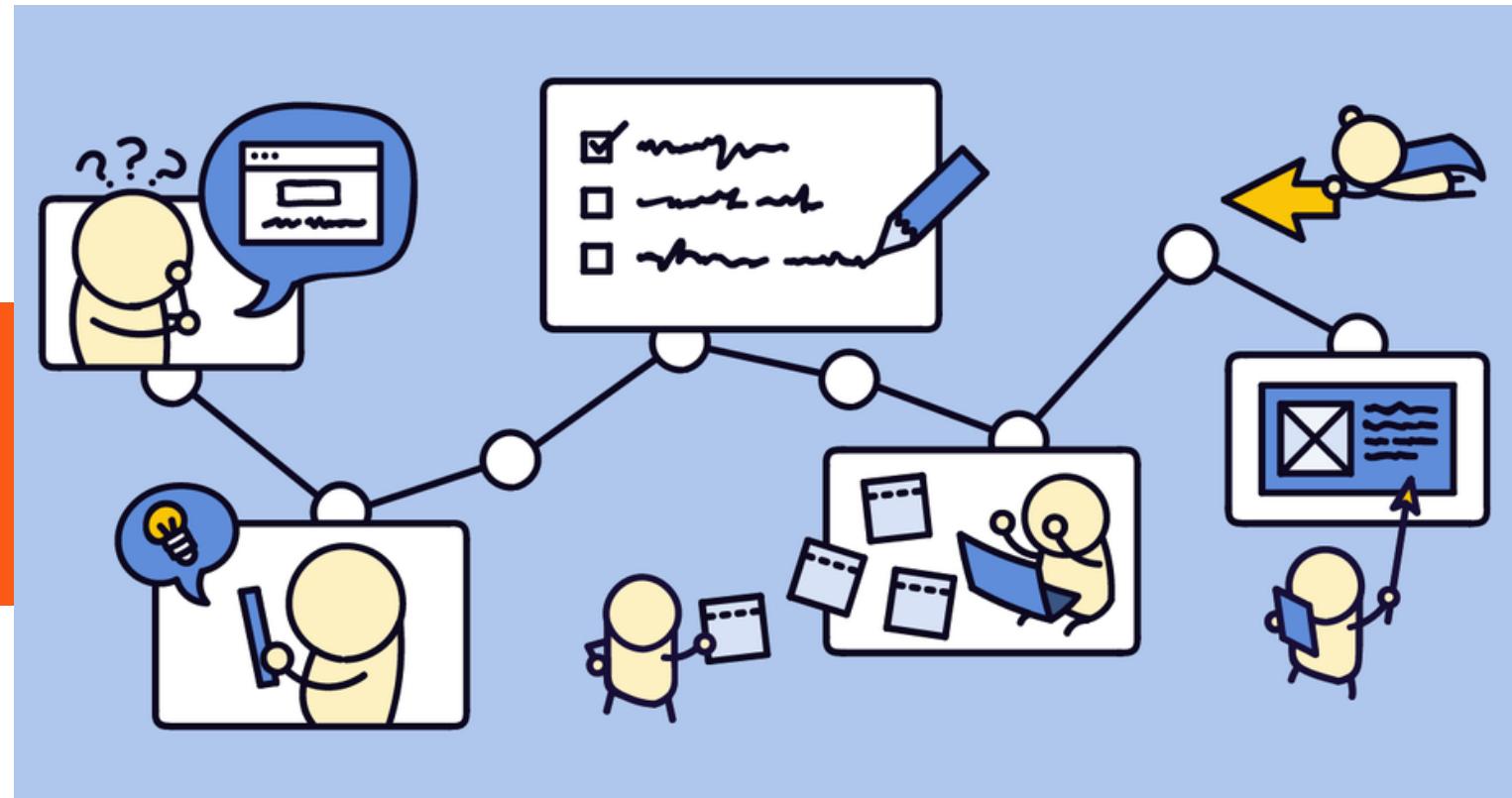
- **Fuente:** Usuario u otra aplicación
- **Estímulo:** Ataque al sistema
- **Artefacto:** los servicios, los datos internos, datos que consume o genera el sistema, componentes en particular.
- **Entorno:** online/offline, connected/disconnected a una red, con firewall/red abierta, fully/partially/not operational.
- **Respuesta:** servicios y datos solo con acceso autorizado, partes identificadas, no repudio, sistema y datos siempre disponibles para legitimar el uso.
Registro de accessos, intentos, posibles ataques
- **Evaluación de respuesta:** impacto, tiempo de recuperación, tiempo de detección, resistencia, cantidad de datos vulnerados



Usabilidad

¿Cuánto tarda el usuario en realizar su tarea?

¿Qué soporte le da el sistema?



Usabilidad

- **Fuente:** Usuario final
- **Estimulo:** uso eficiente, minimizar la posibilidad de errores, sistema adaptable, configurable.
- **Artefacto:** Sistema o parte del sistema con la que interactúa el usuario
- **Entorno:** Run or configuration time
- **Respuesta:** Cumplir expectativas o anticiparse a las necesidades
- **Evaluación de respuesta:** Tiempo de una tarea, cantidad de errores, satisfacción del usuario, cantidad de tareas, aprendizaje del usuario, tiempo perdido con un error,..



Testeabilidad

Queremos que nuestro sistema nos ayude a detectar las fallas **lo antes posible**.



Testeabilidad

- **Fuente:** testers o usuarios finales
- **Estimulo:** Set de tests ejecutados
- **Artefacto:** Unidad de código, subsistema, el sistema.
- **Entorno:** dev, compile, deploy time, sistema en runtime.
- **Respuesta:** Sistema controlable para ejecutar los tests y observar los resultados.
- **Evaluación de respuesta:** tiempo para ejecutar tests, esfuerzo para encontrar las fallas, tamaño de la cadena más larga de dependencias en el test, tiempo en preparar el entorno,..

Otros

- Variability
- Portability
- Development
distributability
- Scalability
- Deployability
- Mobility
- Monitorability
- Safety

Otros

- Variability
- **Portabilitiy**
- Development
distributability
- **Scalability**
- Deployability
- Mobility
- **Monitorability**
- Safety

Portability

*Es un caso particular de Modificabilidad
Mide la facilidad con la que se puede cambiar la*

plataforma

*Se logra minimizando las dependencias con la misma (JVM
por ejemplo)*

*Suele ser requerido cuando se quiere cambiar de
plataforma con un **mínimo esfuerzo***

Scalability

*Existen dos tipos: **vertical [potencia]** y **horizontal [cantidad]***

*Un sistema es escalable, si al aumentar la cantidad de recursos (en potencia y/o cantidad) **mejora algún atributo de calidad de manera notable***

Monitorability

24

*Tener **visibilidad sobre el estado del sistema** para tomar acciones correctivas en caso de ser necesario.*

Suelen monitorearse: queues lengths, process avg time, components health

Finalmente, nuestra arquitectura tendrá que cumplir con...

Requerimientos funcionales



Asignación de responsabilidades a nuestros elementos de arquitectura

Atributos de calidad



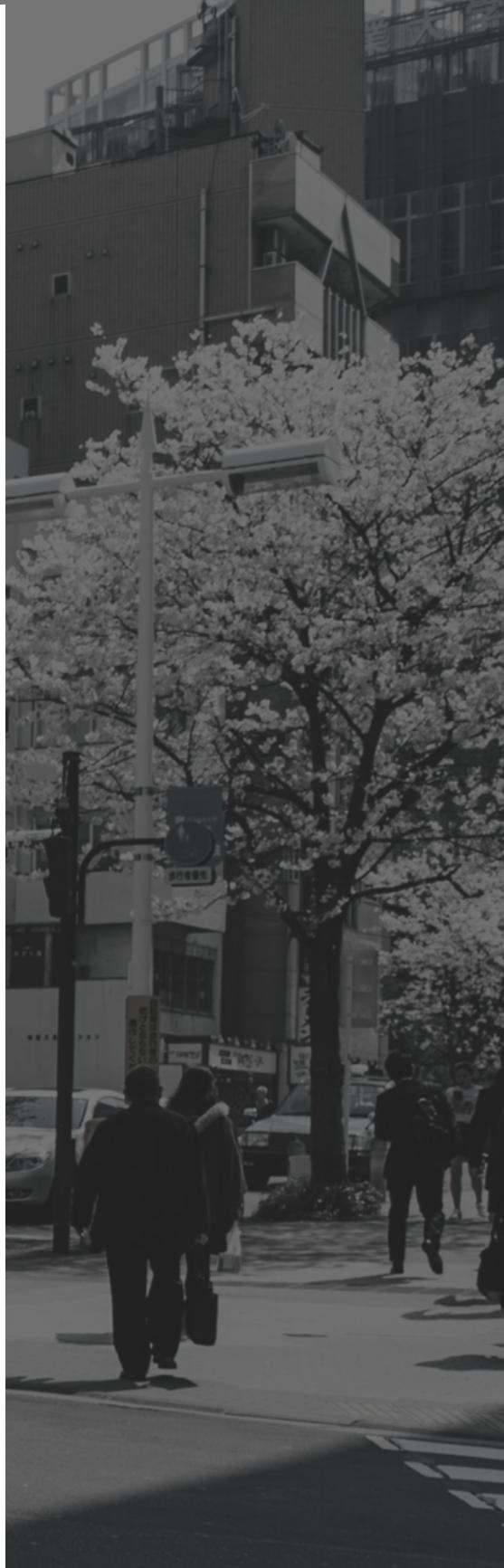
Cómo se comportan nuestros componentes y qué interacción tienen

Restricciones (lo que no dejamos modificable)



Se aceptan según las decisiones que tomemos y cómo se concilian con otras decisiones

¿Preguntas?



Bibliografía

- Software Architecture in Practice (3rd Edition) by Len Bass
(Author), Paul Clements (Author), Rick Kazman (Author)



Gracias!