

Atributos de Calidad de Software

Sistemas de Información II
Juan Carlos Ramos @2016

Atributos de Calidad

■ Agenda

- Atributos de calidad:
 - Conceptos
 - Definiciones

Atributos de Calidad

- Conformen lo que se denomina, comúnmente, “*Requerimientos No Funcionales*”.
- Son los requerimientos sobre la *forma* que debe tener el software.
- Condicionan la manera en que las *funciones* serán llevadas a cabo por el software.

Atributos de Calidad

- Categorías de '*calidades*' de un sistema
 - Calidad en ***tiempo de ejecución***:
 - pueden ser medidas cuando el sistema ejecuta.
 - ¿Qué tan bien satisface el sistema sus requerimientos de comportamiento?
 - ¿Provee los resultados requeridos?
 - ¿Los provee en el tiempo esperado?
 - ¿Son los resultados correctos o dentro de las tolerancias permitidas?
 - ¿Son las funciones del sistema las deseadas cuando se conecta a otros sistemas?

Atributos de Calidad

- Categorías de '*calidades*' de un sistema (y 2):
 - **Calidades estáticas:** no pueden observarse cuando el sistema ejecuta.
 - ¿Qué tan fácil es el sistema para *integrar, testear o modificar*?
 - **Calidad de negocio**
 - ¿Qué tan *costoso* será/fue desarrollarlo?
 - ¿En cuánto *tiempo* estará operativo?
 - **Calidad del diseño/arquitectura**

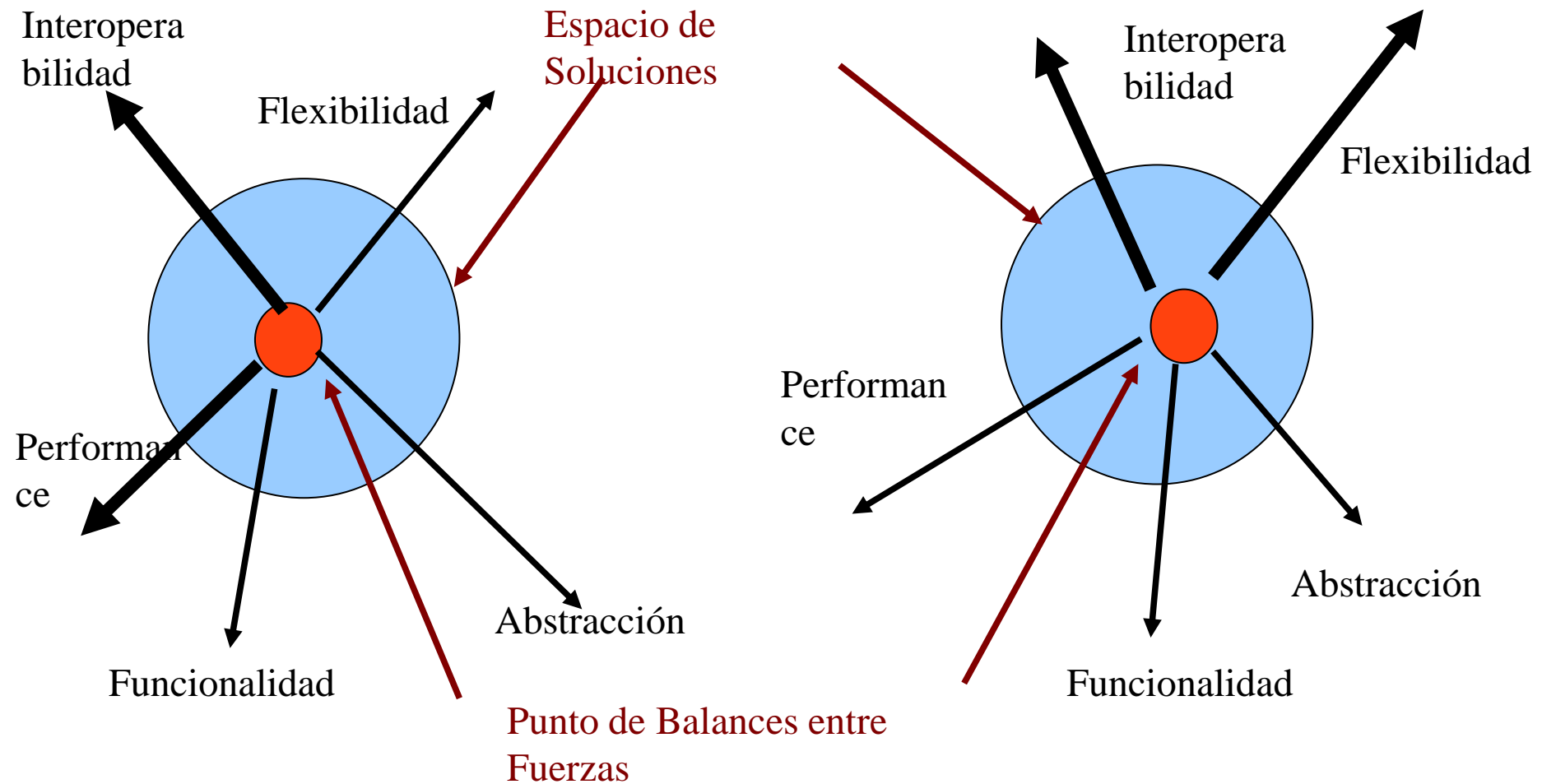
Atributos de Calidad - Interrelaciones

- Alcanzar los atributos de una categoría puede no decir nada acerca de los atributos de otra categoría.
 - Ej.: Un sistema que alcanza todos sus requerimientos en *tiempo de ejecución* puede o no ser *costoso de desarrollar*, o puede o no ser imposible de *modificar*.
 - Por otra parte, un sistema altamente modificable puede o no producir resultados correctos.
- Alcanzar los atributos dentro de una categoría no dice nada acerca de otros atributos dentro de la misma categoría.
 - Por ejemplo, un sistema que es construido en dos semanas puede tomar años para modificar.

Atributos de Calidad - Interrelaciones

- Alcanzar un atributo de calidad ***nunca*** puede hacerse en forma aislada.
- Alcanzar cualquier atributo de calidad puede tener un *efecto positivo* o *negativo* sobre otros atributos de calidad.
 - *Seguridad y Tolerancia a Falla* existen en un estado de tensión mutua.
 - Un sistema *seguro* tiene pocos puntos de falla, un kernel de seguridad → Seguro, pero No Tolerante.
 - Un sistema *tolerante a fallas* tiene más puntos de falla. Conjuntos de procesos o procesadores redundantes → Tolerante pero No Seguro.
 - *Confiabilidad y Performance*
 - Replicar unidades de procesamiento y canales de comunicación → Confiabilidad. Conflicto con performance (ej.: consume más tiempo)
 - Unir procesos críticos para alcanzar performance → Conflicto con confiabilidad (punto de falla simple)

Atributos de Calidad - Interrelaciones



Atributos de Calidad

- La calidad **debe** ser considerada en todas las fases de *diseño, implementación y distribución*.
- Diferentes cualidades se manifiestan en forma diferente durante estas fases.
- Consideraciones:
 - La arquitectura es *crítica* para alcanzar muchas de las cualidades de interés, y estas cualidades deben ser *diseñadas y evaluadas* a nivel arquitectónico.
 - Algunas cualidades no son sensitivamente arquitectónicas, e intentar alcanzarlas a través de la arquitectura será infructuoso.

Atributos de Calidad

■ Ejemplos:

□ Usabilidad

- Hacer la interface de usuario clara y fácil de usar → **No es** un problema arquitectónico (**sí** de diseño).
- Si el sistema debe proveer la capacidad de cancelar operaciones, deshacer operaciones, o reutilizar datos previamente ingresados → **Es** un problema arquitectónico

□ Modificabilidad

- Cómo se divide la funcionalidad en módulos → Arquitectónico
- La técnica de codificación de módulos → **No** Arquitectónico

□ Performance

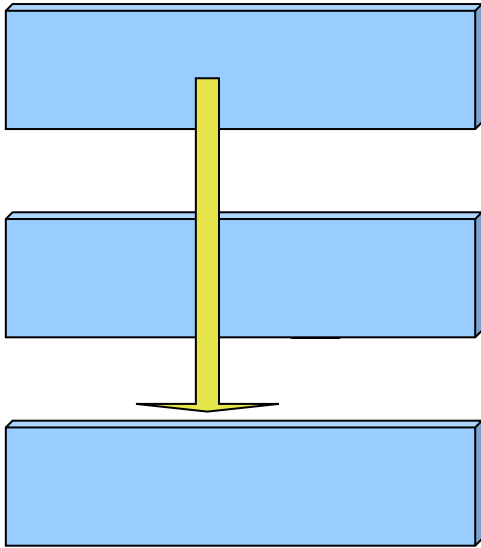
- Cuánta comunicación existe entre los componentes → Arquitectónico
- Cómo son asignados los recursos compartidos → Arquitectónico.
- La elección de un algoritmo → **No** arquitectónico

Cualidades en tiempo de ejecución: sobre la *Funcionalidad*

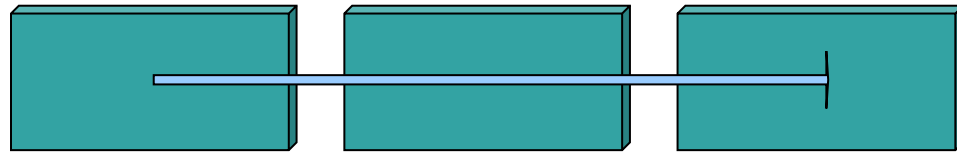
Atributos de Calidad - *Funcionalidad*

- **Funcionalidad:** habilidad del sistema de hacer el trabajo para el cual fue creado.
 - Se alcanza a través de la interacción, cooperación y sincronización de las componentes.
 - Funcionalidad es ortogonal al diseño de software (arquitectura)
 - El diseño del SW al considerar otros atributos de calidad importantes → condiciona la asignación de funcionalidad a la estructura.
 - Funcionalidad puede ser alcanzada a través de diferentes variantes de estructuras posibles.
 - Funcionalidad es *independiente* de la arquitectura.

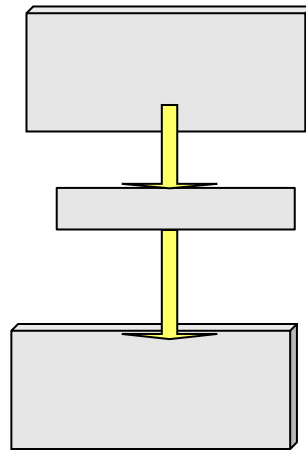
a)



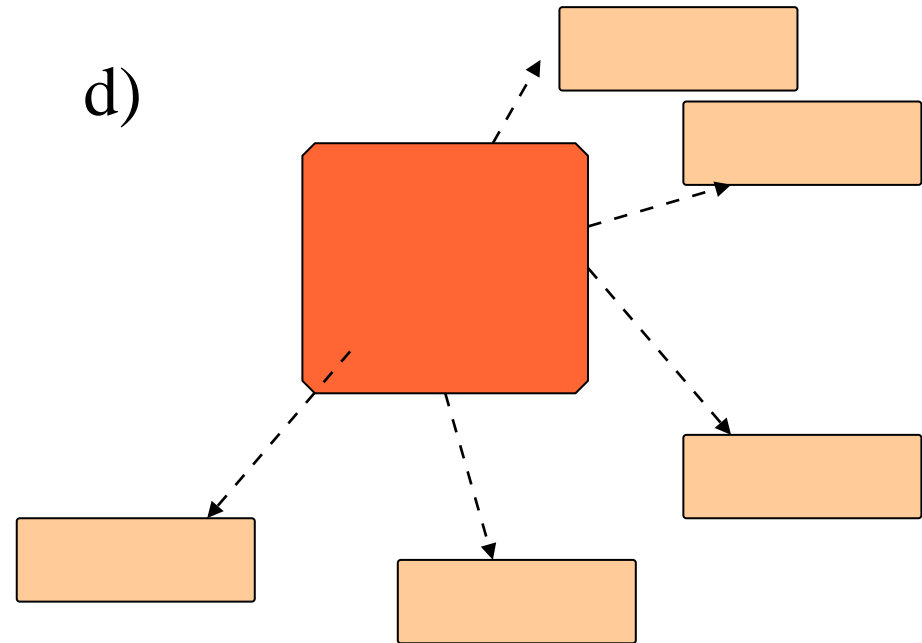
b)



c)



d)



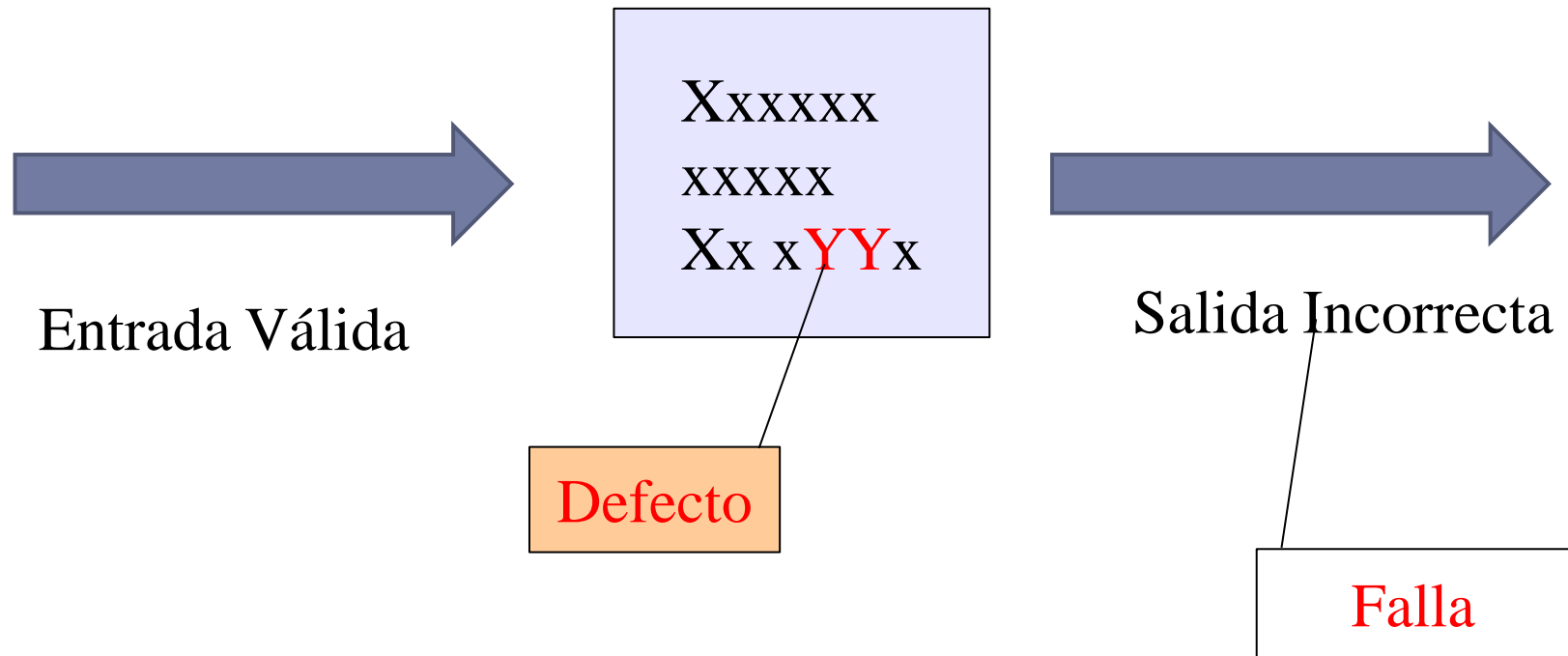
Atributos de Calidad - *Disponibilidad*

■ Disponibilidad (*Availability*):

- ❑ *La medida del tiempo en que el sistema está operativo y ejecutando correctamente.*
- ❑ Analiza las fallas del sistema y sus consecuencias.
- ❑ Una *falla* del sistema ocurre cuando este no entrega más un servicio consistente con sus especificaciones.
- ❑ Esta falla es *observable* por los usuarios del sistema (humano u otros sistemas).

Atributos de Calidad - *Disponibilidad*

- Es necesario distinguir entre *falla* y *defecto*.



Atributos de Calidad - *Disponibilidad*

- Un *defecto* llega a ser una *falla* si no es corregido u ocultado.
- Una *falla* es observable por el usuario del sistema, un *defecto* no.
- Una vez que el sistema falló, es importante el tiempo necesario para repararlo.
- ***Tiempo de reparación***: tiempo hasta que la falla no es más observable.

Atributos de Calidad - *Modificabilidad*

- **Modificabilidad:**
 - *Facilidad con la cual el software puede adecuarse a los cambios.*
- **Cualidad *estática***
- **Acerca del *costo del cambio*.**
- **Problema netamente arquitectónico: afectado por las características de las componentes de SW.**
- **Modificabilidad \equiv Mantenibilidad (Maintainability)**

Atributos de Calidad - *Modificabilidad*

- ¿Qué puede ser cambiado (artefacto)?
 - *Cualquier* aspecto del sistema puede ser cambiado.
 - Comúnmente las *funciones* que el sistema realiza.
 - La plataforma del sistema (hardware, sistema operativo, middleware, etc.) → *Portabilidad*
 - El entorno de operación (los sistemas con los que debe interactuar, protocolos de comunicación, etc.)
 - Las *cualidades* que el sistema exhibe (performance, fiabilidad, etc.)
 - Su capacidad (cantidad de usuarios soportados, cantidad de operaciones simultáneas, etc.)
 - Los cambios pueden ser Agregar, Eliminar, o Modificar.

Atributos de Calidad - *Modificabilidad*

- ¿*Cuándo* se realiza el cambio y *quién* lo hace (el entorno)?
 - Cambios durante la *implementación* (modificando el código fuente)
 - Durante la *compilación* (utilizando switches en el compilador)
 - Durante la *construcción* (mediante la elección de librerías)
 - Durante el *setup de configuración* (ej. Seteo de parámetros)
 - Durante la *ejecución* (seteo de parámetros)
 - Un cambio puede ser realizado por un desarrollador, un usuario final o un administrador de sistema.

Atributos de Calidad - *Modificabilidad*

■ Categoría de cambios:

- *Extender o cambiar capacidades*: Agregar nuevas funcionalidades, extender funcionalidades existentes, o reparar 'bugs'.
 - *Extensibilidad*: capacidad de adquirir nuevas funcionalidades.
 - *Eliminar capacidades no deseadas*: Simplificar las funcionalidades de un sistema (sistema menos capaz y menos costoso)
 - *Adaptarse a nuevos entornos operativos*: nuevos procesadores, dispositivos de entrada/salida, dispositivos lógicos, etc.. → *Portabilidad*
 - *Reestructuración*: Reasignación de servicios del sistema, modularización, optimización, o creación de componentes reusables.
-
- Una vez que el cambio ha sido especificado, debe ser diseñado, implementado, testeado y distribuido.
 - Toma tiempo y dinero, lo que puede ser medido.

Atributos de Calidad - *Modificabilidad*

- ▶ Portabilidad: la habilidad del sistema para ejecutar en diferentes entornos de computación (hardware, software, o ambos).
- ▶ Un sistema es portable cuando logra colocar todas las relaciones con un entorno operativo particular en una componente o un conjunto pequeño de componentes fácilmente intercambiables.
- ▶ Un tipo especial de modificabilidad.
- ▶ Medida: cuánto cuesta en términos de tiempo y dinero portarlo.

Atributos de Calidad - *Performance*

- Performance: se refiere a las respuestas del sistema, ya sea el *tiempo requerido* para responder a eventos específicos o la *cantidad de eventos procesados* en un intervalo de tiempo dado [Smith, 1993]
- Normalmente expresado por
 - la *cantidad de transacciones por unidad de tiempo* o
 - por el *tiempo que toma completar una transacción* con el sistema.

Atributos de Calidad - *Performance*

- **Eventos** (interrupciones, mensajes, requerimientos de usuarios, o el paso del tiempo) ocurren y el sistema debe responder a estos.
- Performance está relacionado con *cuánto tiempo le toma al sistema responder cuando un evento ocurre*.
- Las diferentes fuentes de los eventos (usuarios, otros sistemas o dentro del mismo sistema) y los patrones de arribo → Complican el problema de performance.

Atributos de Calidad - *Performance*

- No se ha considerado hasta aquí si el sistema es en red o simple usuario (standalone).
- No se consideró la configuración del sistema, ni la forma en que se consumen los recursos.
- *Comunicación toma más tiempo que los cálculos*
- → **Performance es una función de cuanta comunicación e interacción hay entre las componentes** (problema arquitectónico).

Atributos de Calidad - Usabilidad

► Usabilidad:

- Qué tan fácil es para un usuario realizar una tarea requerida (Facilidad de uso), y la clase de soporte a usuarios que el sistema provee (entrenamiento y ayuda).
- Aspectos de la usabilidad:
 - **Aprendizaje:** ¿Qué tan fácil y rápido es para el usuario aprender a utilizar el sistema?. ¿Qué puede hacer el sistema para hacer el aprendizaje más fácil?
 - **Eficiencia:** ¿el SW responde con la velocidad apropiada a las actividades del usuario? ¿Qué puede hacer el sistema para que el usuario haga en forma más eficiente su trabajo?
 - **Minimizar el impacto de errores:** ¿El sistema anticipa y previene errores comunes de usuarios? ¿Puede el sistema hacer que el impacto de un error sea mínimo?

Atributos de Calidad - Usabilidad

▶ Aspectos de la usabilidad (y 2)

- ▶ **Manejo de errores:** ¿El sistema ayuda al usuario a recuperar errores?
- ▶ **Adaptar el sistema a las necesidades del usuario:** ¿Cómo puede el usuario (o el sistema mismo) adaptarse para hacer las tareas del usuario más fácil?
- ▶ **Asistencia:** ¿El sistema se comunica bien y asiste al usuario en la resolución de problemas?.
- ▶ **Incrementar confianza y satisfacción:** ¿Qué hace el sistema para dar al usuario confianza que las acciones correctas están siendo realizadas?

► Referencias

- “Software Architecture in Practice” – Segunda Edición, Len Bass, Paul Clement, Rick Kazman, Addison-Wesley, 2003
 - Chapter 4. Understanding Quality Attributes
- **Complementaria**
- “Quality Attributes”, Mario Barbacci, Mark H. Klein, Thomas A. Longstaff, Charles B. Weinstock, Technical Report, CMU-SEI, 1995
- “Quality Attribute Design Primitives”, Len Bass, Mark Klein, Felix Bachmann, Technical Note, CMU/SEI-2000-TN-017, 2000