

Ingeniería de Software II

Segundo Cuatrimestre de 2016

Clase 14: Especificación de Atributos de Calidad y QAW

Buenos Aires, 17 de Octubre de 2016

Los Atributos de Calidad

- ▶ La funcionalidad «de negocio» es sólo una parte de lo que un sistema debe hacer
- ▶ Además, están los atributos de calidad (“ilities”), que hablan de características específicas que debe tener el sistema (anteriormente llamados “requerimientos no funcionales”).
 - ▶ Ejemplo: portabilidad, flexibilidad, usabilidad
- ▶ Necesitamos conocerlos para definir una arquitectura
- ▶ En muchos casos, se afectan entre si. Por ejemplo, portabilidad vs. performance o flexibilidad vs. Performance
- ▶ *"Software quality is the degree to which software possesses a desired combination of attributes."*

[IEEE Std. 1061]

Algunas realidades sobre los atributos de calidad

- ▶ Suelen estar pobremente especificados, o directamente no especificados (“un requerimiento que no es testeable no es implementable”)
- ▶ En general no se analizan sus dependencias
- ▶ La importancia de estos atributos varía con el dominio para el cual se construye el software
- ▶ Además de requerimientos funcionales y atributos de calidad, el ingeniero de software debe identificar correctamente restricciones.
- ▶ Las “tácticas” de arquitectura no son fines en si mismas, son formas de alcanzar atributos de calidad deseados
- ▶ El atributos de calidad que suele ser más importante: la flexibilidad (“facilidad de cambios”)

Ingeniería de Requerimientos

- ▶ Una forma disciplinada y sistemática de llegar desde las necesidades de los usuarios a una especificación
- ▶ Lo que hay que conocer para definir bien una arquitectura (y por lo tanto para hacer bien un sistema):

Requerimientos
Funcionales
“de negocio”

Otros Atributos
de Calidad

Restricciones

Distintas clasificaciones de atributos de calidad

‣ **IEEE Std 1061 / ISO 9126**

- Efficiency
- Functionality
- Maintainability
- Portability
- Reliability
- Usability

‣ **Mitre**

- Efficiency
- Reliability
- Usability
- Maintainability
- Expandability
- Interoperability
- Reusability
- Integrity
- Survivability
- Correctness
- Verifiability
- Flexibility
- Portability

Atributos de calidad: Apertura en ISO 9126 – IEEE 1061

Características

Sub características / “concerns”

Funcionalidad

Adecuación

Correctitud

Interoperabilidad

Seguridad

Conformidad

Confiabilidad (reliability)

Tolerancia a Fallas

Facilidad de Recuperación

Usabilidad

Fac. Aprendizaje

Fac. Comprensión

Fac. Operación

Atractividad

Eficiencia

Comportamiento Temporal

Utilización de Recursos

Mantenibilidad

Fac. Análisis

Fac. Cambios

Estabilidad

Facilidad de Prueba

Portabilidad

Adaptabilidad

Fac. Instalación

Coexistencia

Fac. Reemplazo

Atributos de calidad – Disponibilidad

- ▶ Relacionada con fallas (“failures”) en el sistema y sus consecuencias asociadas
- ▶ Un “failure” ocurre cuando un sistema no entrega más un servicio de acuerdo con su especificación
- ▶ Esas “failures” son observables por los usuarios (personas u otros sistemas).
- ▶ Error <> Defecto (defect) <> Fault <> Failure
- ▶ Tiempo de reparación = tiempo hasta que la falla no es más observable
- ▶ Disponibilidad = probabilidad de que un sistema esté disponible cuando se lo necesite
 - ▶ $D = \text{Mean Time to Failure} / (\text{Mean Time to Failure} + \text{Mean Time to Repair})$
- ▶ Los “downtimes” programados no se consideran
- ▶ Relativamente fácil de especificar, difícil de verificar

Disponibilidad (Ejemplos SUBITE)

- ▶ “En zonas donde la cobertura 3G – LTE 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.”

Atributos de calidad – Facilidad de cambios

- ▶ Relacionada con el costo de los cambios. Uno de los atributos de calidad más difíciles de expresar
- ▶ Temas importantes:
 - ▶ ¿Qué puede cambiar?
 - ▶ Funcionalidad
 - ▶ Plataforma
 - ▶ Otros atributos de calidad
 - ▶ Interfaces
- ▶ ¿Quién y dónde se hace el cambio?
 - ▶ Usuarios, desarrolladores, administradores
 - ▶ Código, configuración, parametrización
- ▶ Una vez que un cambio se especifica, debe ser diseñado, implementado, probado y liberado

Escuchemos al amigo Parnas



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.”

Performance

- ▶ Relacionada con el tiempo que le lleva al sistema responder a un evento que ocurre (interrupciones, mensajes, pedidos de usuarios o paso del tiempo).
 - ▶ Latencia: tiempo entre la llegada del estímulo y el inicio de la respuesta del sistema
 - ▶ “Jitter”: variación en la latencia
 - ▶ Deadlines: límites de tiempo para un proceso
 - ▶ Throughput: cantidad de transacciones que el sistema puede procesar en un período de tiempo
 - ▶ Eventos no procesados
- ▶ Difícil de expresar. Depende de volúmenes del sistema, equipamiento en uso y versiones de sistema operativo y otros software de base.

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 sería bien recibido los usuarios, ni los choferes de colectivos. Esto es especialmente crítico en horas pico.”

Seguridad

- ▶ Habilidad de un sistema para resistir usos no autorizados y seguir proveyendo sus servicios a usuarios legítimos. Incluye:
 - ▶ Nonrepudiation: mecanismos para asegurar que quienes hicieron algo no puedan negarlo
 - ▶ Condifencialidad: propiedad por la cual datos o servicios son protegidos de accesos no autorizados
 - ▶ Integridad: propiedad por la cual datos o servicios se brindan como fue previsto.
 - ▶ Disponibilidad (en el contexto de seguridad): que un sistema esté disponible para su uso legítimo
 - ▶ Auditabilidad: habilidad de un sistema para hacer un seguimiento de actividades realizadas

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.”

Usabilidad

- ▶ Usabilidad: Relacionada con la facilidad con la cual un usuario puede cumplir una tarea o utilizar un servicio ofrecido por el sistema y el tipo de soporte que provee el sistema.
 - ▶ Aprender la funcionalidad del sistema
 - ▶ Usar el sistema eficientemente
 - ▶ Minimizar el impacto de los errores
 - ▶ Adaptar el sistema a las necesidades de los usuarios
 - ▶ Aumentar confianza y satisfacción

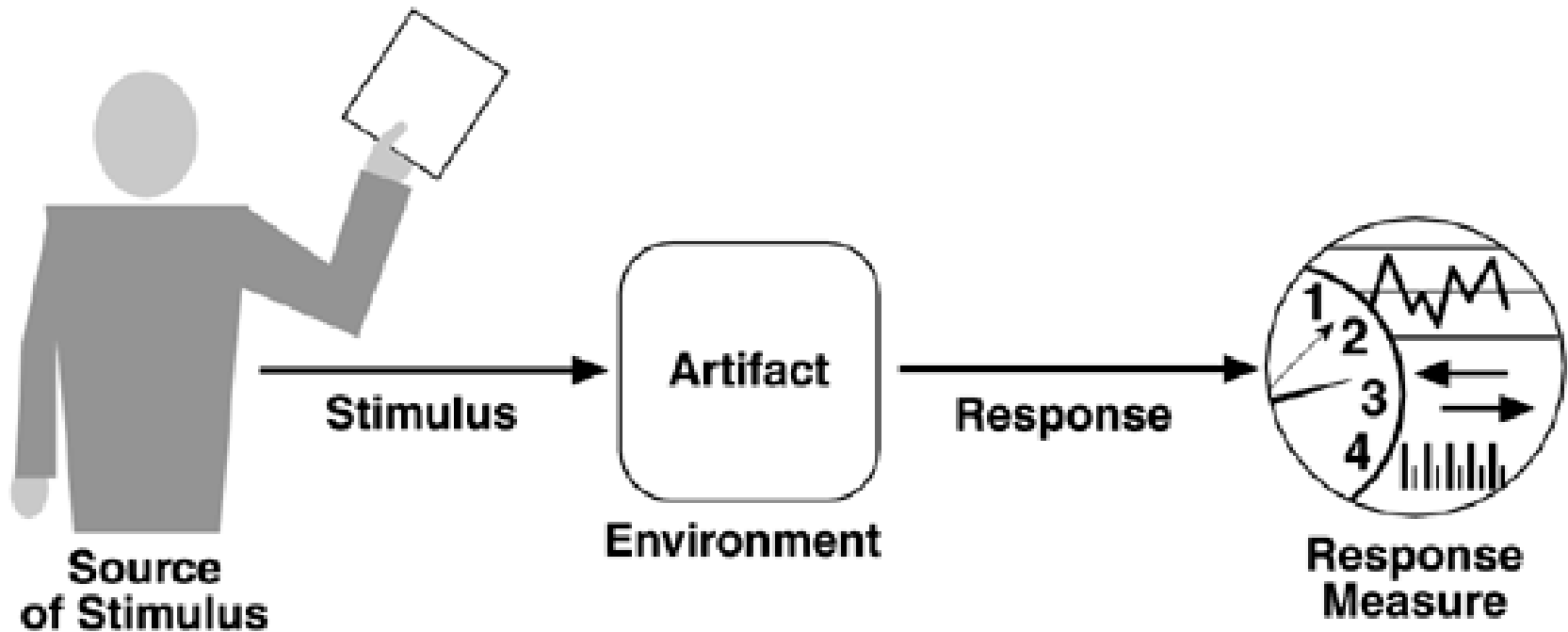
Usabilidad (Ejemplos SUBITE)

- ▶ “Dado que en los colectivos será el propio pasajero quien determine el valor del pasaje mediante una interfaz de tipo *touchscreen*, se desea que la misma sea intuitiva y permita realizar dicha operación de manera rápida y sencilla en la mayoría de los casos.”
- ▶ “Se quiere que el sistema de información de tarifas, rutas, y tiempos sea accesible para personas con discapacidades visuales.”

Otros Atributos

- ▶ Escalabilidad: una medida de qué tan bien una solución sigue cumpliendo con sus requerimientos al cambiar los volúmenes del problema que resuelve.
 - ▶ Ej. Subite: "Si bien en un principio, los medios de pago estarán acotados a unos pocos, el gobierno desea que progresivamente se vayan colocando terminales de recarga en distintos puntos de la ciudad para que se pueda recargar el saldo en las tarjetas inteligentes de proximidad. Luego, el sistema debe ser capaz de soportar una cantidad creciente de transacciones de recarga."
- ▶ Portabilidad: facilidad de un sistema para poder ser operado en distintas plataformas.
- ▶ Facilidad de testing: posibilidad de ver el «estado interno» de la aplicación.

Especificación de Atributos de Calidad (SEI)



Quality Attribute Scenario, formado 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

Ejemplos de escenarios de atributos de calidad

- ▶ Escenario de **disponibilidad**:
 - ▶ “Si la comunicación entre una Terminal de Cobro móvil y el Sistema Central se pierde, o los tiempos de transmisión son prohibitivos, la Terminales de Cobro debe de seguir funcionando en modo *offline*, de forma transparente al usuario.”
 - ▶ Fuente: Interno al sistema
 - ▶ Estímulo: Tiempo promedio transacciones mayor a 1 seg.
 - ▶ Entorno: Operación normal
 - ▶ Artefacto: Terminal de Cobro
 - ▶ Respuesta: La terminal pasa a modo *offline*, utilizando la información local almacenada en las tarjetas y *logueando* toda operación realizada para que este lista cuando la conexión se restablezca.
 - ▶ Medición: No se requieren acciones adicionales por parte del usuario, y no se ve afectada la performance (≤ 1 seg. X trans.).

Ejemplos de escenarios de atributos de calidad (cont.)

- Escenario de **facilidad de cambios**:
 - “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.”
 - Fuente: Ministerio de Planificación
 - Estímulo: Agregar al sistema la división por zonas y basar el costo de los pasajes en las mismas
 - Entorno: En tiempo de diseño
 - Artefacto: Sistema
 - Respuesta: Cambio efectuado sin efectos secundarios, el pago se efectúa con la nueva modalidad de cobro
 - Medición de la respuesta: Se invierten menos de 550 horas hombre. La usabilidad no se ve afectada negativamente.

Ejemplos de escenarios de atributos de calidad (cont.)

- Escenario de **performance**:
 - “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 sería bien recibido por los usuarios, ni los choferes de colectivos. Esto es especialmente crítico en horas pico.”
 - Fuente: Pasajero
 - Estímulo: Se aproxima o inserta una tarjeta para realizar el pago de un pasaje
 - Entorno: En operación normal y hora pico
 - Artefacto: Terminal de Cobro
 - Respuesta: Se concreta la operación de la venta del pasaje.
 - Medición de la respuesta: tiempo de respuesta total ≤ 1 seg.

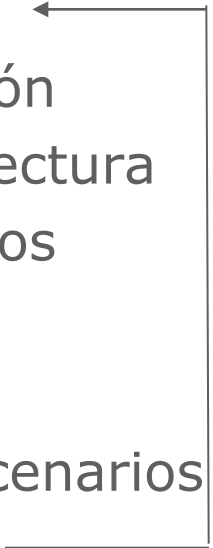
Ejemplos de escenarios de atributos de calidad (cont.)

- Escenario de **seguridad**:
 - “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...”
 - Fuente: Atacante externo
 - Estímulo: Captura un mensaje encriptado e intenta vulnerar la seguridad para obtener información confidencial de las transacciones de los usuarios
 - Entorno: Operación Normal
 - Artefacto: Módulos de encriptación de las terminales y el Sistema Central
 - Respuesta: Los datos no son accesibles al usuario en tiempos razonables (miles de años)
 - Medición de la respuesta: (tiempo para “romper” el sistema, % de casos en los que el ataque no es exitoso).

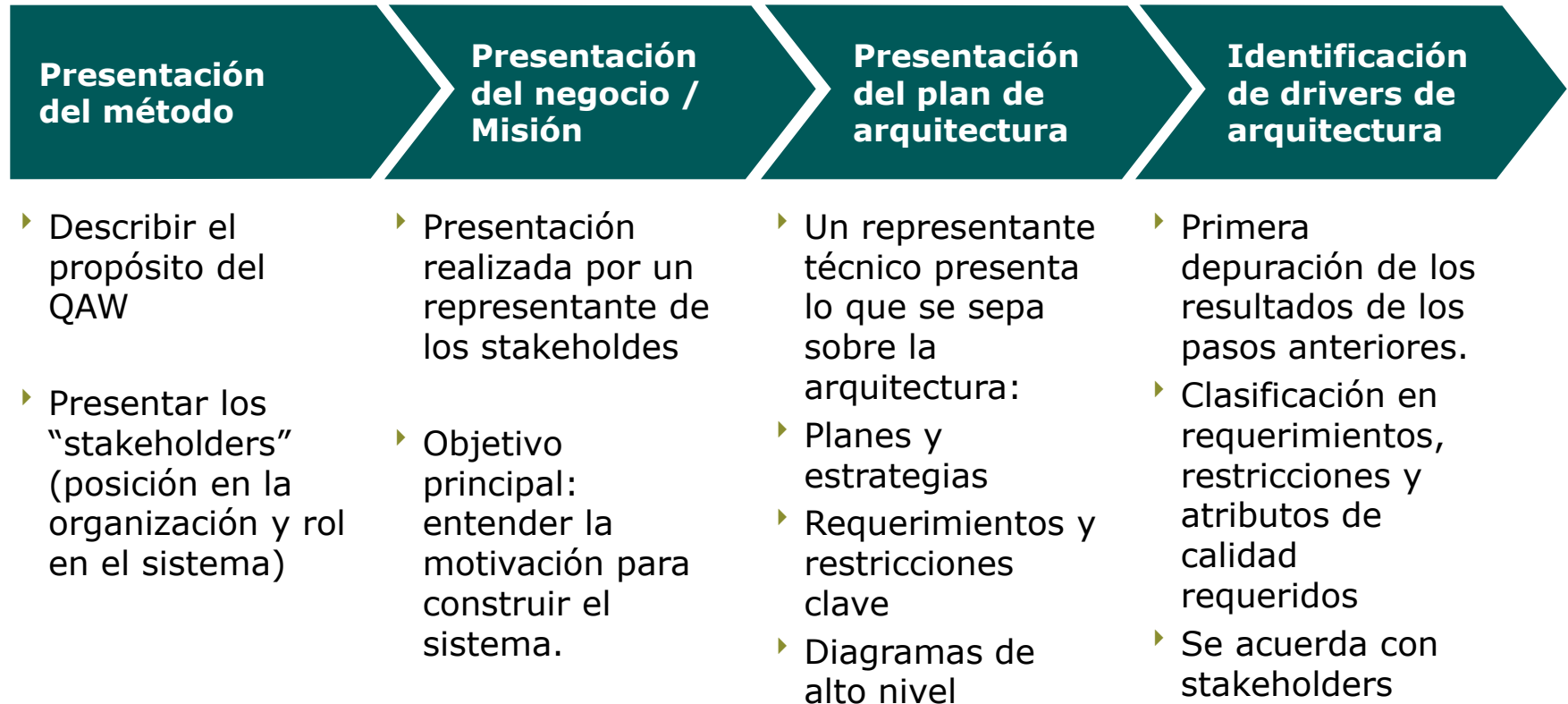
Ejemplos de escenarios de atributos de calidad (cont.)

- Escenario de **usabilidad**:
 - “Dado que en los colectivos será el propio pasajero quien determine el valor del pasaje mediante una interfaz de tipo *touchscreen*, se desea que la misma sea intuitiva y permita realizar dicha operación de manera rápida y sencilla en la mayoría de los casos.”
 - Fuente: Pasajero
 - Estímulo: Selecciona pasaje que quiere adquirir
 - Entorno: Operación normal
 - Artefacto: Terminal de Cobro/Recarga, Interfaz gráfica
 - Respuesta: El pasajero realiza la operación satisfactoriamente
 - Medición de la respuesta: En test usabilidad, 95% de los pasajeros pueden realizar la operación, sin ningún tipo de ayuda de otra persona, la primera vez que se enfrentan al sistema, y en menos de 20 segundos.

Quality Attribute Workshops

- ▶ El Quality Attribute Workshop (QAW) es un método facilitado que relaciona los stakeholders de un sistema de manera temprana en el ciclo de vida para descubrir los atributos de calidad clave en un sistema de software
 - ▶ Sus pasos son:
 - ▶ Presentación del método
 - ▶ Presentación del negocio / misión
 - ▶ Presentación del plan de arquitectura
 - ▶ Identificar drivers arquitectónicos
 - ▶ Brainstorming de escenarios
 - ▶ Consolidación de escenarios
 - ▶ Definición de prioridades de escenarios
 - ▶ Refinamiento de escenarios
- 
- Iterar mientras sea necesario agregando "stakeholders"

Detalle del proceso (1)



Detalle del proceso (2)

