

Ingeniería de Software II

Segundo Cuatrimestre de 2011

Clase 13: Especificación de Atributos de Calidad y QAW

Buenos Aires, 6 de Octubre de 2011

Proyecto "SUBITE"

- ▶ Antes de empezar, consideremos la siguiente información de un sistema a desarrollarse. La misma nos servirá como marco de referencia para trabajar de forma más natural sobre los contenidos.
- ▶ El gobierno nacional ha decidido emprender un proyecto de **renovación del transporte público** con el fin de:
 - ▶ Crear empleos.
 - ▶ Reducir el nivel de circulación de dinero en efectivo.
 - ▶ Tener un mayor control de las empresas prestadoras.
 - ▶ Mejorar el servicio.
- ▶ El proyecto, llamado "**SUBITE**", consiste principalmente de la unificación de todos los sistemas de pago de transporte público (subterráneos, colectivos, trenes y *posiblemente* taxis y remises) en las grandes urbes del país, utilizando un **boleto electrónico**.
- ▶ Sobre las tarjetas:
 - ▶ **Tarjetas inteligentes de proximidad** y recargables.
 - ▶ Modalidad prepago y pospago.
 - ▶ **Tarjetas de bandas magnéticas** anónimas y desechables.
 - ▶ Sólo modalidad prepago (monto fijo precargado, venta en negocios/kioscos/estaciones).

SUBITE (Descripción General)

‣ **Terminales de Cobro (TC):**

- Ubicadas en cada colectivo de cada línea y en cada estación de subte y tren.
- Permitir la compra electrónica de boletos vía interfaz *touchscreen*.

‣ **Terminales de Recarga (TR):**

- Ubicadas en puntos estratégicos de las ciudades.
 - Recargar dinero de tarjetas inteligentes.
- Ambas deben de reconocer tarjetas válidas de inválidas.
- Ambas deben de brindar información de recorridos, tarifas y tiempos estimados de viajes (interacción con sistema GIS).
- Terminales conectadas directamente con un **Sistema Central** para obtener saldos de usuarios.
- Conectividad especial de **TC** móviles (colectivos):
 - Mediante 3G (+ Costoso. Para horas pico. Cobertura reducida).
 - Mediante SMS (- Costoso. Para horas no pico. Mayor cobertura).
 - Sistema de **Emergencia** Offline (*Sync* cuando haya conexión disponible / en las cabeceras de los medios de transporte).

SUBITE (Descripción General)

- ▶ Tarjetas **inteligentes prepagas** podrán ser recargadas mediante diversos medios:
 - ▶ Las terminales de recarga (TR) que deben aceptar tarjetas de debito y crédito, además de efectivo.
 - ▶ "Pago Fácil", "Red Banelco y Link".
 - ▶ Mediante SMS y la factura del teléfono celular.
- ▶ Tarjetas **inteligentes pospagas** se le enviará una factura mensual al titular por los gastos efectuados.
- ▶ El **Banco Central** recibirá la información de las terminales y será el único encargado de efectuar el *clearing*, el pago a las empresas y el manejo de las cuentas de los usuarios pospago.
- ▶ Empleados de la **AFIP** y el **Ministerio de Planificación** podrán consultar informes **recientes y actualizados** sobre:
 - ▶ Ganancias de las empresas.
 - ▶ Volumen de pasajes vendidos por cada una.
- ▶ El proyecto será implementado en primer lugar en **Capital Federal**, para luego ampliarlo a Gran Buenos Aires, y el resto del país. Posibilidad de venderlo y exportarlo a otros países.

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, los atributos de calidad 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
requeridos

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

| Características | Subcaracterísticas / «concerns» | | | | |
|-----------------------------|---------------------------------|-------------------------|-------------------|---------------------|-------------|
| Funcionalidad | Adecuación | Corrección | Interoperabilidad | Seguridad | Conformidad |
| Fiabilidad (reliability) | Madurez | Tolerancia a Fallas | Recuperabilidad | | |
| Usabilidad | Aprendibilidad | Comprensibilidad | Operabilidad | Atractividad | |
| Eficiencia | Comportamiento Temporal | Utilización de Recursos | | | |
| Mantenibilidad | Analizabilidad | Cambiabilidad | Estabilidad | Facilidad de Prueba | |
| Portabilidad | Adaptabilidad | Instalabilidad | Coexistencia | Reemplazabilidad | |

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 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.”

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. Todo esto cuesta dinero.

Escuchando a los que saben...



David Parnas: "The part of the game of designing good software is designing for change"

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 tendría buena acogida en los usuarios, ni en 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

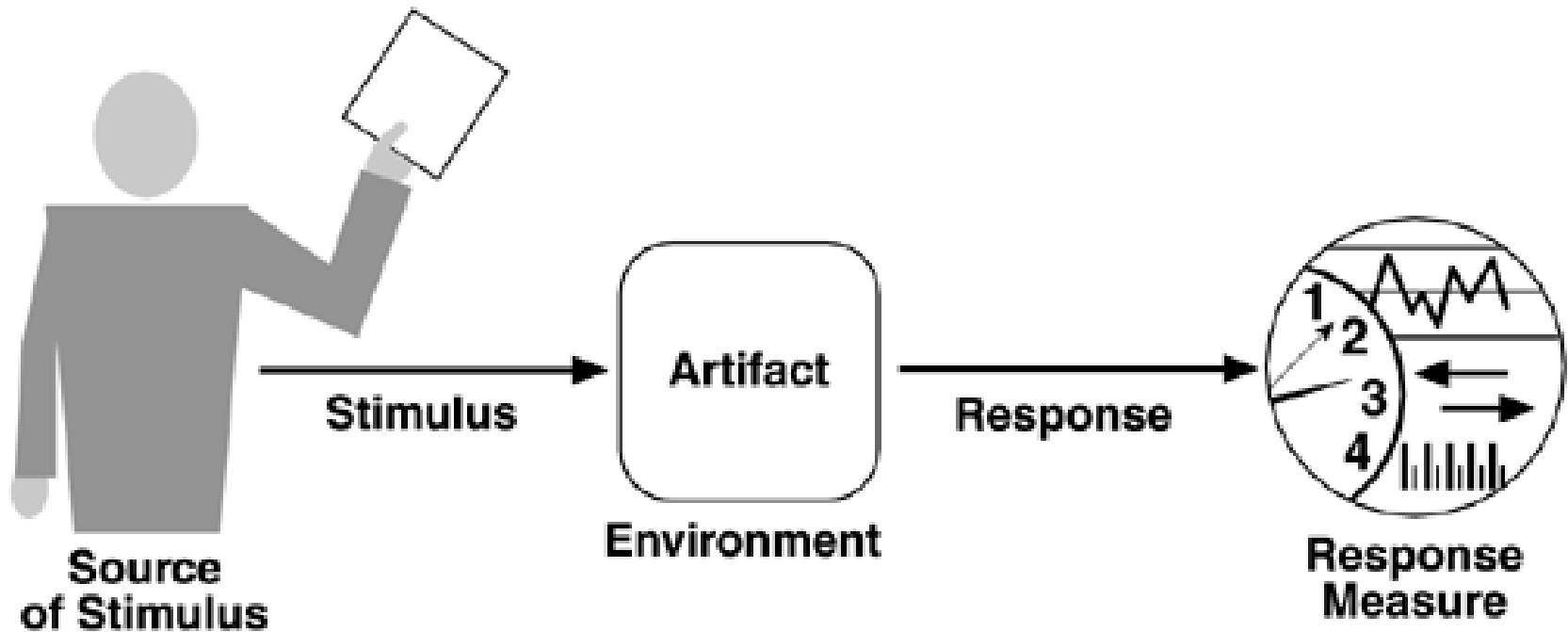
Usabilidad (Ejemplos SUBITE)

- ▶ “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 tomada 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: Las transacciones de cobro de pasaje en una terminal de cobro están tardando mas de 1 segundo.
 - ▶ 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 segundo por transacción).

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: Se quiere agregar al sistema la división por zonas y basar el costo de los pasajes en las mismas.
 - Entorno: En tiempo de ejecución.
 - 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 (35 % del tiempo del proyecto original). 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 tendría buena acogida en los usuarios, ni en 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 materializa la operación de la venta del pasaje.
 - Medición de la respuesta: Se demora a lo sumo 1 segundo.

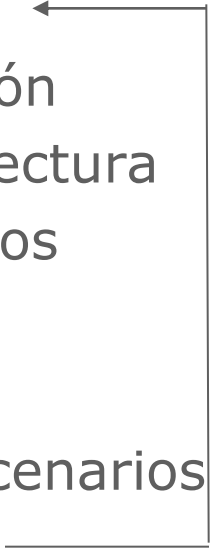
Ejemplos de escenarios de atributos de calidad (cont.)

- 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: Se captura un mensaje encriptado e intenta vulnerar la seguridad del mismo 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: En el 99.9999% de los casos.

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: El pasajero.
 - Estímulo: El pasajero interactúa con una terminal.
 - Entorno: Operación normal.
 - Artefacto: Terminal de Cobro/Recarga, Interfaz gráfica.
 - Respuesta: El pasajero realiza la operación satisfactoriamente.
 - Medición de la respuesta: El 90% 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 30 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"

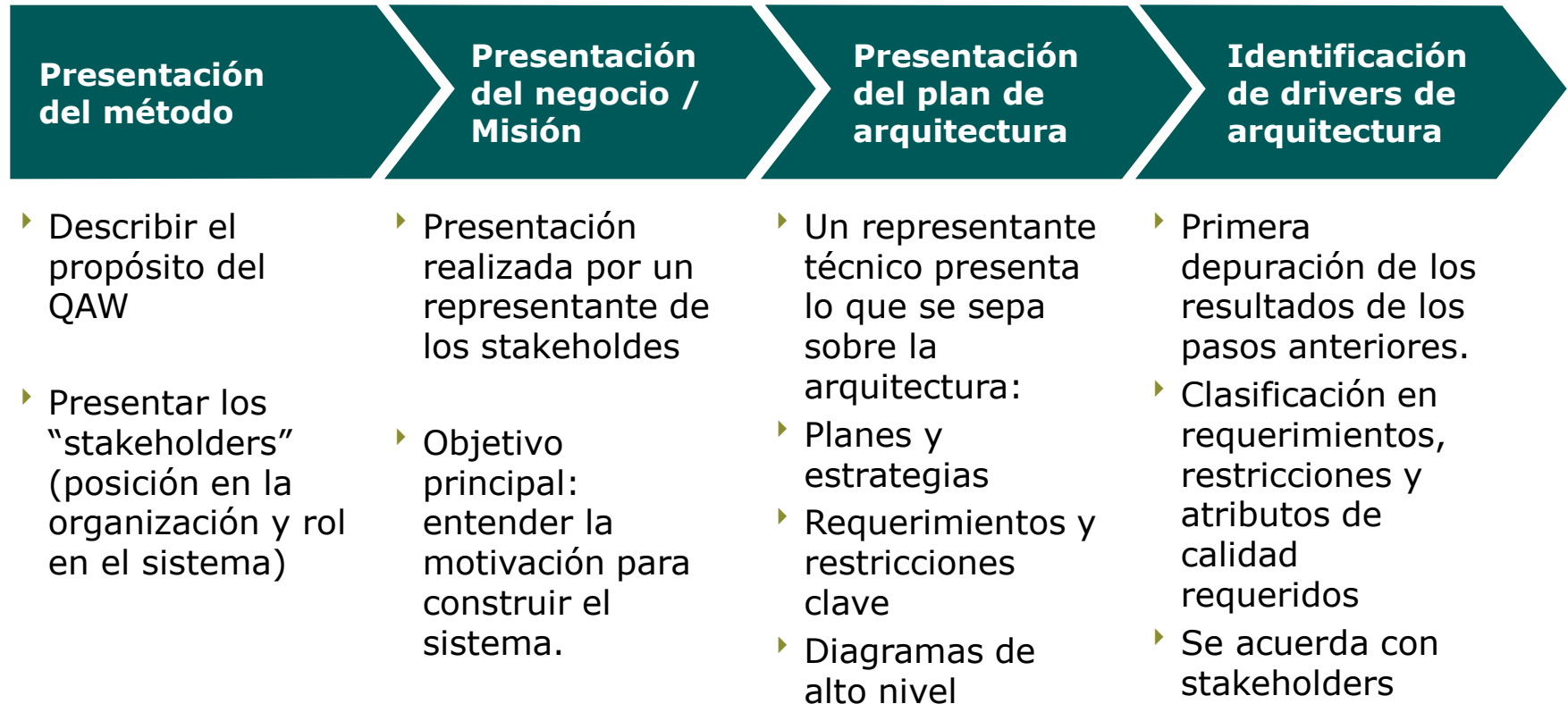
Motivación – ¿Qué problema queremos atacar?

- ▶ ¿Cuál es el significado preciso de los atributos de calidad, como modificabilidad seguridad, performance, y confiabilidad, en el contexto del sistema que se está construyendo?
- ▶ ¿Cómo se pueden descubrir, caracterizar, y dar prioridades a los atributos clave antes de que se construya el sistema?
- ▶ ¿Cómo se puede hacer que una comunidad dispersa geográficamente de “stakeholders” se involucren de una manera disciplinada y repetible en el descubrimiento y la caracterización de atributos de calidad?
- ▶ ¿Cómo se puede usar toda esta información?

Roles y duración

- ▶ Evento de 1 día ofrecido como servicio por el SEI
- ▶ Roles:
 - ▶ Líder de QAW: facilitador de las discusiones
 - ▶ Asistente: registra la información de las sesiones
- ▶ Existe un template del SEI para minutas de las sesiones

Detalle del proceso (1)



Detalle del proceso (2)



SUBITE (QAW)

- Stakeholders del proyecto:
 - Ministro de Planificación e Infraestructura. Es el portavoz de la presidencia:
 - Quiere que el sistema sea rápido, eficiente, aceptado por la gente y que pueda ser puesto en marcha lo antes posible.
 - Le interesa conocer exactamente cuando vende cada empresa para otorgar los subsidios correspondientes.
 - Director de Sistemas (nombrado por Planificación).
 - Muy interesado por la performance. Preocupado por el costo de la comunicación 3G para mantenerla y la latencia que esta tiene cuando la red celular está congestionada.
 - Quiere mantener el gasto con las empresas de telecomunicaciones al mínimo.
 - Gerente de Sistemas Banco Nación.
 - Quiere asegurar el no repudio de las operaciones al hacer el *clearing*. Le preocupa la integridad de la información.
 - Le gustaría que el sistema este la gran mayoría del tiempo *online*, o mejor dicho, no quiere que este offline.

SUBITE (QAW)

- Stakeholders del proyecto (cont.):
 - Representante de Cámara de Empresas Transportistas de Colectivos
 - Les preocupa el sistema online (no quieren que los auditen muy de cerca).
 - No quieren tener problemas con los choferes, ni quejas de la gente por la incorporación del nuevo sistema.
 - Sindicato de Empleados de Transporte.
 - Quieren aprovechar el sistema para que se mantenga el control de los recorridos, no se explote a los choferes y que se cumpla el horario laboral.
 - Representante de Defensa del Consumidor y Sindicatos.
 - Les interesa que sea muy simple de usar y accesible a todos.
 - No quieren que haya estafas, ni costos trasladados a los consumidores del servicio.
 - Les resulta interesante el sistema zonal.

SUBITE (QAW)

- ▶ Extracto de primera lista informal de requerimientos / restricciones del dominio encontradas:

[...]

- ▶ Que el proyecto esté terminado antes del 2011 (POC antes del 22 de nov), e incluya al menos colectivos y subtes.
- ▶ Que el sistema sea rápido (a lo sumo 1 segundo para aceptar una tarjeta).
- ▶ Que el pasajero común lo pueda usar fácilmente.
- ▶ Mínima interacción del colectivo, para que no lo distraiga de la conducción.
- ▶ Que las tarjetas no se puedan falsificar.
- ▶ Que el sistema esté online el mayor tiempo posible.
- ▶ Todas las transacciones deben quedar registradas correctamente.
- ▶ Que se use la red online en forma eficiente.
- ▶ Poder financiar el proyecto con el aporte de otras empresas, que el sistema se pueda expandir para adaptarse a algunas nuevas funcionalidades/integraciones.
- ▶ Registrar cuánto se viaja en cada empresa de transporte.
- ▶ Que la información de la facturación le llegue SOLAMENTE al Banco Nación (evitar los intrusos) y los datos personales de los usuarios estén guardados en forma segura. [...]

SUBITE (QAW)

► Resultado de votación

- Después de la votación (o la fuerza) quedarían los siguientes escenarios priorizados (junto a su descripción/detalle utilizando la técnica aprendida aquí no incluida):
 - 1. Disponibilidad (Todos)
 - 2. Performance (Gobierno, Banco Nación y Transportistas)
 - 3. Seguridad (Sobre todo auditabilidad) (Gobierno, Banco Nación, Defensa del consumidor)
 - 4. Escalabilidad (Gobierno, Banco Nación, Director de sistemas)
 - 5. Facilidad de Cambios (Defensa de Consumidor [sistema Zonal])
 - 6. Usabilidad (Defensa del Consumidor)